

Report number:

2047352_1-TN-01

Title:

Review of methods and toolkits for uncertainty quantification of single and coupled-model applications

Authors:

Maxime Vassaux, Wouter Edeling, Peter V. Coveney
University College London

Executive summary:

The present report draws a concise review of uncertainty quantification methods classified according to their intrusiveness. Attention is paid to non-intrusive and later semi-intrusive methods which enable to define procedures which are independent of the models equations, thus enabling full separation of concern. The report also presents upto-date toolkits, libraries and pieces of software that enable the high-throughput ensemble-based computations required for verification, validation and uncertainty quantification. The report concludes with a brief review of direct attempts to quantify uncertainties with existing plasma fusion codes.



UK Atomic
Energy
Authority

1 Introduction

Uncertainty quantification, verification and validation processes are crucial in order to demonstrate the robustness of all forms of simulation. Code results can be "validated" by comparison with experiment in a number of ways, ranging from qualitative (subjective) measures to quantitative measures which apply a validation metric. Verification (confirmation that the mathematical model has been coded correctly) and validation of computer simulations have been discussed at length for fluid dynamics [1]. Applications to fusion have been made in a number of subsequent papers, including "Validation in fusion research: Towards guidelines and best practices" [2], "Verification and validation for magnetic fusion" [3] and "Validation metrics for turbulent plasma transport" [4].

Computer modelling is widely used in science and engineering to study systems of interest and to predict their behaviour. These systems are usually multi-scale or multi-physics in nature, as their accuracy and reliability depend on the correct representation of processes taking place on several length and time scales involving different physics [5–8]. The resulting code often simulates a collection of coupled models. Moreover, these systems can be stochastic, since there are always some unresolved scales whose effects are not taken into account due to lack of knowledge or limitations of computational power [6,9]. Additionally, measurements of model parameters, model validation, and initial and boundary conditions themselves can be rarely if ever achieved with perfect accuracy [10]. Therefore, the simulation model and its output results inevitably contain uncertainties, and one needs to estimate their magnitudes by applying a forward uncertainty quantification (UQ) method.

UQ is familiar in engineering and applied mathematics communities but quite immature at lower length and time scales relevant of physics and chemistry, let alone in combinations which arise in multiscale applications. Handling a large multiscale/multiphysics problem is arguably among the most complex one can address. Collectively speaking, verification, validation and UQ for such systems is an active research topic and off-the-shelf solutions remain absent.

It is standard practice in UQ to distinguish two sources of uncertainty – “epistemic” and “aleatoric”. The former addresses systematic errors (caused by parameter values, etc.), the latter random ones, which are linked to the use of random numbers generators and random seeds. Importance must be attached to intrinsic stochasticity coming from chaos. Turbulence is the primary source in fusion research, but it is also present in many particle-based methods (such as classical molecular dynamics). Our current investigations of binding affinity calculations using molecular dynamics show that aleatoric uncertainty can more than double the variability of predictions compared with studies performed without ensemble averaging. For epistemic UQ, information about the distribution of the uncertainty in the parameters must be specified, but such information is rather rarely known. In our own work, we have often had to assume uniform distributions across a fixed range (say up to 20% changes in the parameter of interest).

The purpose of the present report is to provide a short overview of approaches to uncertainty quantification including recommendations as to which are likely to be of most relevance to the Neptune project. This report draws heavily upon the experience gathered over recent years including the past three years running the VECMA project (www.vecma.eu). The goal of the project is to provide an open source toolkit (VECMAtk, www.vecma-toolkit.eu) containing a wide range of tools to facilitate the use of VVUQ techniques in multiscale, multi-physics applications [39]. Approaches are classified by degree of intrusiveness, and we focus on ones suited to enable separation of concerns, that is avoiding the development of methods on a per-application basis. The report then provides a review of existing toolkits enabling the execution of UQ workflows on high-performance computing infrastructures.

2 Classification of methods by intrusiveness

Usually a distinction is made between *intrusive* UQ methods, where one substitutes the original model with its stochastic representation, and *non-intrusive* methods, where the original model is used as a black-box [11,12]. Intrusive methods are efficient and relatively easy to apply to linear models, e.g. [13]. This, however, represents only a relatively small class of models. They can be applied to non-linear models as well, but the solution of the resulting equations may become very demanding. Non-intrusive methods can be applied to any type of non-linear model. However, if a single model run requires large execution times, these UQ methods may be ineffective, or even computationally intractable.

Non-intrusive uncertainty propagation methods consider the entire system as one black box, see Figure 1. The main advantage is that the (legacy) simulation code is left completely untouched, hence the name ‘non-intrusive’. This allows users to quickly add a UQ component to their existing simulation framework. The one thing any application user must do is write an encoder/decoder to allow a code of interest to connect to EasyVVUQ. To facilitate this, several non-intrusive methods are considered in the VECMAtk [14] and more specifically in EasyVVUQ [15] (which will be described in section 5): quasi Monte Carlo (qMC), Polynomial Chaos (PC) and the stochastic collocation (SC) methods [16]. All these methods follow a similar pattern, namely:

1. Specify the input distributions and draw samples (create a so-called “design-of-experiment”).
2. Run the ensemble.
3. Perform post-processing analysis.

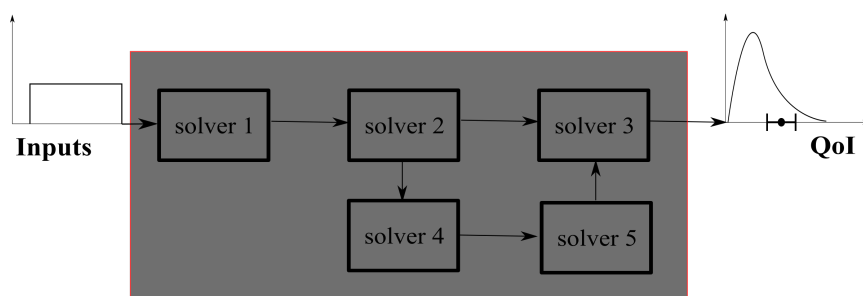


Figure 1: Schematic of non-intrusive uncertainty propagation through a multiscale system of coupled single-scale solvers, mapping input distributions to a distribution of any output Quantity of Interest (QoI). The propagation technique is agnostic with respect to the structure of the multiscale system and treats it as a black box.

The stochastic Galerkin method [17, 18] is often labelled as intrusive, due to the fact that dedicated solvers have to be developed in order to tackle the stochastic problem at hand. The equations of the problem are rewritten directly with stochastic variables. The additional programming effort is usually regarded as a major disadvantage, especially in the case of complex computational models whose software and underlying solvers are difficult to be accessed, modified or otherwise manipulated. Therefore, and despite the fact that stochastic Galerkin methods have appealing properties for error analysis and estimation, collocation methods are generally preferred, as they allow for non-intrusive, black-box use of the original computational models. It must be noted that the separation into intrusive and non-intrusive methods is an ongoing topic of discussion, see e.g. [19].

An intermediate class of methods exist for codes which couple multiple models. Such methods are called *semi-intrusive* UQ algorithms [20]. These algorithms are intrusive only on the level of the multiscale model, that is, in the way the single scale components are coupled together. The single scale components themselves are, however, treated as black-boxes, see Figure 2. Semi-intrusive algorithms will be discussed in more detail in section 6.

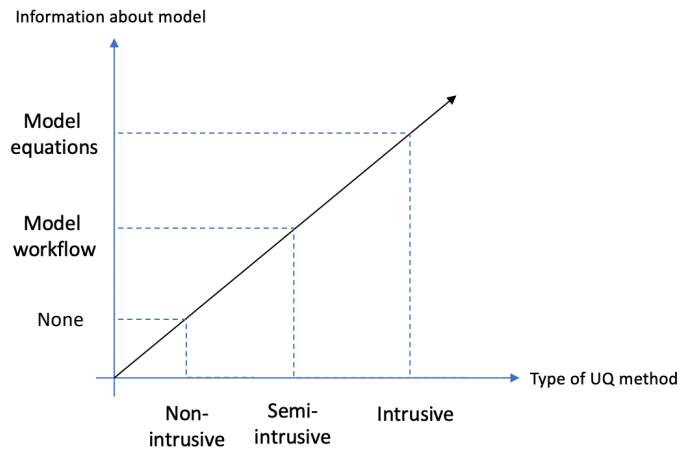


Figure 2: Intrusiveness of UQ methods. The different levels of intrusiveness are associated with the components of an application which need to be modified to quantify uncertainty.

3 Enhanced sampling methods

Most commonly, UQ studies rely on sampling methods. Monte Carlo (MC) sampling converges irrespective of the number of random variables (RVs) or the regularity of the given problem, albeit with a slow convergence rate in the mean-square-error sense. Improved cost-error ratios can be achieved with multilevel MC methods [21]. Spectral UQ approaches converge much faster, exponentially in the most favourable cases, for a small to moderate number of random inputs and smooth input-to-output map [22]. Typical methods of this type are stochastic collocation [16,23,24] or point collocation [25,26] methods. Comparisons between stochastic and point collocation methods, see e.g. [27], indicate that the former tends to provide superior accuracies and convergence rates for smooth quantities of interest (QoI). However, since these approaches differ significantly, a fair comparison between the two is still an open research topic, as also indicated in [26].

A common bottleneck of all aforementioned methods is the so-called “curse of dimensionality” [28], i.e. convergence rates deteriorate, and computational costs increase with the number of considered input parameters, by definition, exponentially. As a possible remedy, state-of-the-art methods employ sparse, adaptively constructed polynomial approximations, see e.g. [29,30] for adaptive stochastic collocation methods and [25] for adaptive point collocation methods. While generally not free of the curse of dimensionality, adaptive methods exploit possible anisotropies among the input parameters regarding their impact upon the QoI. Assuming that such anisotropies exist, adaptivity may enable studies with a comparably large number of input parameters. More recently, tensor decompositions (see [31] and the references therein) have been used to exploit possible low-rank structures of parametric problems in order to tackle the curse of dimensionality. In several cases, again relying on high regularity, superior asymptotic convergence rates have been obtained compared to sparse grid methods [32]. However, comparisons between these methods remain an active field of research. In EasyVVUQ only adaptive stochastic collocation methods were considered and applied to the large-scale UQ of the CovidSim code [69]. In the search for an acceptable compromise between computational work and approximation accuracy, such approaches are receiving increasing attention in uncertainty quantification. Dimension-adaptive methods are based on nested univariate collocation points, e.g. Clenshaw-Curtis and Genz-Keister nodes are typical choices for uniform and normal input distributions, respectively.

The aforementioned adaptive algorithms don’t break the curse of dimensionality, they postpone it. Although the sampling plan is iteratively refined in directions that are found to be more important than others, they ultimately still create a sampling plan in a high dimensional space. A class of methods that attempts to circumvent this are the so-called High-Dimensional Model Representation (HMDR) models [59]. Without going into detail, the basic idea is to write the model response as an expansion of component

functions of increasing dimension (akin to the ANOVA expansion). The assumption is then made that in most physical models, (very) high-order interaction effects between parameters are not important. This is not proven, but often observed in practice. One can then truncate the expansion at for instance second-order interactions. Each remaining component function must now be approximated by, for instance, QMC or stochastic collocation, which can be readily performed since each function is at most two-dimensional. Thus, instead of trying to sample a single high-dimensional space as efficiently as possible, the problem is broken up into a series of low-dimensional subproblems. This could potentially be implemented in EasyVUQ, since the machinery to approximate each component function is already in place. Note that instead of manually choosing the order at which to truncate the expansion, the order can also be found adaptively [60]. Although the number of component functions can be large, this algorithm does have a high degree of parallelism, as all component functions of a given order can be approximated in parallel.

Thus far we have discussed adaptivity in the stochastic dimensions. Another type of adaptivity relates to locally refining the stochastic space (of a given dimension), in the case when the response in this space is not entirely regular. The stochastic collocation and polynomial chaos methods write the code output as an expansion over *global* polynomials. However, if say a discontinuity exists in the stochastic domain, an expansion over global polynomials can lead to the well-known Runge phenomenon. Various methods exist that instead use a (polynomial) basis with local support, e.g., Adaptive Sparse Grid methods [61] or the Simplex Stochastic Collocation method [62]. Adaptivity in this case means placing more samples in regions of the stochastic space where the solution is not regular. It is also possible to combine dimension adaptivity with local adaptivity, see e.g. [63].

Active subspace methods [64] are a more recent class of UQ methods that deal with high dimensional input spaces. These are not adaptive in nature, but instead use gradient information to find a matrix that projects the high-dimensional input vector to a low-dimensional ‘active subspace’, in which most of the variation takes place. Although certainly promising, the classical active subspace method requires the gradient of the output with respect to the inputs to be available, which will not always be the case. At the Turing Institute, work has been performed which combines active subspace ideas with Gaussian Processes, without the need for computing the gradients [65]. Finally, machine-learning methods for finding active subspaces have recently also been developed, for instance the “Deep active subspaces” [66] or “Deep UQ” frameworks [67]. These also work without the need for gradients. An early implementation is in development within the EasySurrogate module within the VECMA Toolkit.

4 Surrogate modelling methods

The construction and use of surrogate models (also referred to as metamodels or emulators) is a central computational strategy in UQ [11]. A surrogate model is trained or fitted to the output of a limited number of evaluations of an expensive computational model. Once trained, the surrogate can replace the expensive model and thereby enable tasks that require many model evaluations, e.g. detailed assessment of forward uncertainty propagation, or Bayesian model calibration.

Techniques to construct surrogates that are well-established in the UQ domain include Non-Intrusive Spectral Projection (based on Polynomial Chaos Expansion), interpolating polynomials resulting from stochastic collocation, and Gaussian Process regression (also known as “kriging”). They are not specifically aimed at the multi-model setting, however. A step forward was the semi-intrusive approach (detailed in section 6) where it was shown that these existing techniques can be successfully used as elements in a multi-model UQ framework.

Below we give an overview of newly developed, advanced techniques to obtain a surrogate model $\tilde{\mu}$ from an original model μ . Specifically, we discuss:

1. Stochastic surrogates
2. Reduced surrogates

4.1 Data driven stochastic surrogates

When given parametric states can correspond to multiple μ model states methods, stochastic surrogate modelling (or stochastic parameterization) of the model are necessary to account for the uncertainty in the μ state. In the VECMA project, methods have been developed that resample μ data coming from a reference simulation, conditioned on given parametric states. In our case of multi scale modelling, μ is often a quantity derived from the expensive micro model, for which we wish to make a surrogate. Furthermore, let X be some collection of parametric. This could include the QoI Q , although not necessarily. In general, our surrogate $\tilde{\mu}$ takes the form of a conditional probability density function, i.e.

$$\tilde{\mu}_{j+1} \sim \mu_{j+1} \mid \tilde{X}_j, \tilde{X}_{j-1}, \tilde{X}_{j-2}, \dots, \tilde{\mu}_j, \tilde{\mu}_{j-1}, \tilde{\mu}_{j-2}, \dots \quad (1)$$

Here, the index j corresponds to a given time t_j . Thus, in addition to a stochastic nature, we also have the option of embedding memory into the surrogate by conditioning on multiple time steps into the past. This is especially relevant when there is no clear time scale separation between the submodels. In essence, by conditioning as $\mu_{j+1} \mid \tilde{X}_j, \tilde{X}_{j-1}, \tilde{X}_{j-2}, \dots, \tilde{\mu}_j, \tilde{\mu}_{j-1}, \tilde{\mu}_{j-2}, \dots$ we identify a subset of candidate μ_{j+1} reference samples, from which we randomly sample one value (i.e. $\tilde{\mu}_{j+1}$) to be used as the prediction for the next time step t_{j+1} .

Eq. (1) describes a class of different models. Within VECMA, we have implemented a model based on the so-called ‘‘binning’’ concept from [33]; see [34]. Here, the space of conditioning variables is discretized into a set of non-overlapping bins, where each bin contains a given number of reference samples from μ . This is a direct way to identify the required subset of reference samples, since the conditioning variables will lie inside a single bin at every time step. The results of the implementation were positive [34]. Notwithstanding this, a downside of the approach is that it is subject to the curse of dimensionality, since the number of bins grows exponentially with the number of time-lagged conditioning variables.

To circumvent this problem, we have developed a conditional resampling model based on probabilistic classification via machine learning [35]. Now, instead of binning the conditioning variables, the output (i.e. the reference μ samples) is placed into K non-overlapping bins. The advantage is that this avoids the curse of dimensionality, since we do not include any memory in the output, i.e. the number of bins remains equal to K . A neural network is used to learn a discrete probability mass function (PMF) over the K output bins, conditional on the time-lagged macroscopic input features. At any timestep, they can sample a bin index from this PMF, and subsequently resample μ reference data from the designated bin, see Figure 3.

They applied these stochastic surrogates to problems in the context of climate modelling. As mentioned earlier, the goal here is to obtain a surrogate such that the overall, time-averaged statistics of the macroscopic solver are accurately captured. The results obtained to date are positive, when applied to a simplified atmospheric model [35] and to a more complex two-dimensional ocean circulation model [34]. Furthermore, the neural network approach has been extended to include a kernel-mixture network [36], enabling construction of a continuous Probability Density Function (PDF) instead of the discrete PMF used until now.

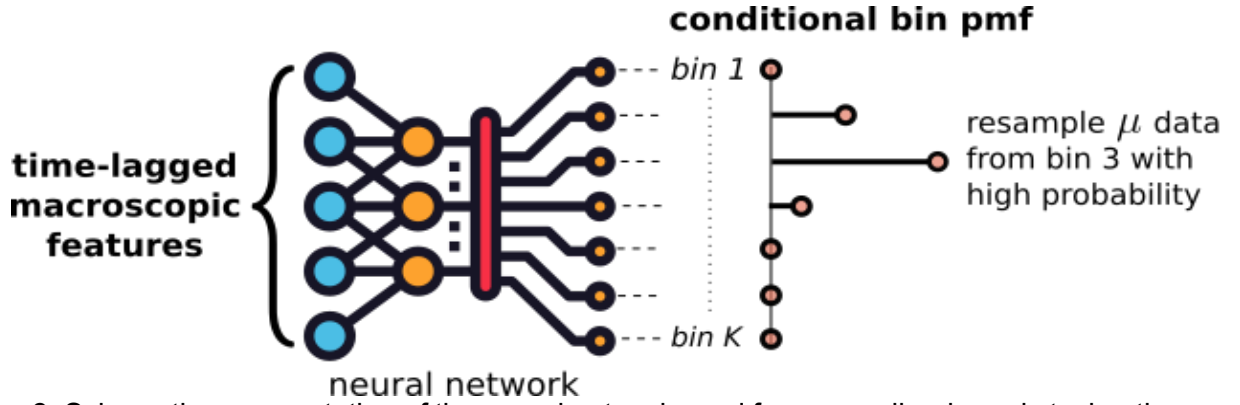


Figure 3: Schematic representation of the neural network used for resampling-based stochastic surrogate modelling, as proposed in [35].

4.2 Data-driven reduced surrogates

Multiscale models often have a high dimensional state space. As a consequence, the target of a surrogate model, for instance a subgrid-scale term in a turbulence simulation, also has a high number of degrees of freedom. That said, despite this high dimension, the QoI could just be a function that takes the high dimensional code output, and produces a single scalar. For instance, in a climate context, it is not uncommon that the QoIs are global, spatially integrated, quantities. Within VECMA, we have developed so-called reduced surrogate models that exploit such a massive difference in size between the model state and the QoIs.

A model state has fixed parametric dimension. Hence, a surrogate model must have the same dimension as the original model. However, the *unclosed component* the surrogate models can be controlled. The unclosed component is the only part which must be learned from data, as the closed component is fully determined from known variables. In the VECMA project, a procedure has been developed where the unclosed component of the surrogate model is of the same size as a set of *a priori* defined integrated QoIs. This can be viewed as a pre-processing procedure which generates new training data that is reduced in size by several orders of magnitude compared to the original surrogate model μ . For instance, if we have a 2D model with 64 points in each spatial direction, and our QoIs are 4 scalar time series (computed from the high-dimensional model state), we can reduce the training data size of each snap shot in time from 64^2 to 4, without any significant loss of accuracy in our QoIs during the training phase. Effectively, instead of creating a surrogate for a high-dimensional dynamic field, we only need to create a surrogate for a small number of scalar time series, as far as accuracy in our pre-defined set of QoIs is concerned. The methodology is described in [37,38]. Briefly, the surrogate model is given by the following expansion:

$$\tilde{\mu}(x, y, t) = \sum_{i=1}^d \tau_i(t) P_i(x, y, t)$$

Here, $\tau_i(t)$ are the generated new training data for which a surrogate must be learned, and the $P_i(x, y, t)$ are dynamic fields which are completely made up of known, (macroscopic) variables. Hence, the P_i do not need to be learned from data, and can be computed without reference to the expensive micro model. In principle, any type of surrogate can be trained on the generated τ time series data.

Thus far, we have only tested this method on two-dimensional problems. The reduction in training data size for three-dimensional problems will even be greater (e.g. 64^3 to 4 in our example). However, our current focus lies on training surrogates on the generated reduced training data, and solving the equations with a trained reduced (microscopic) surrogate in place. This is a challenging problem, as a surrogate (in general) is trained offline to fit the training data. It is not directly trained to perform well in an online coupled modelling environment, in which there is a two-way interaction between the surrogate and the

(macroscopic) governing equations. This does not have to be a problem (see e.g. [35]), although we (and others) have also observed that this can yield incorrect results. To circumvent this, we are currently investigating the effectiveness of a second, online training phase, see the work of [68] for the general methodology.

5 Semi-intrusive uncertainty quantification for multi-model applications

The semi-intrusive methods for multiscale UQ are a family of algorithms which employ the structure of the multiscale and multi-physics codes in order to perform an efficient UQ, that is, estimating the uncertainties with comparable quality as the black box MC method, but with a substantially reduced execution time. According to the Multiscale Modelling and Simulation Framework [55], instead of considering the whole code as a black-box, the code can be seen as a collection of coupled single scale black-box sub-models. Thus, the semi-intrusiveness of the methods boils down to a limited inspection of the multi-model code, which is only up to the level of single scale components and their coupling. Below the main ideas behind the semi-intrusive UQ methods is described.

5.1 Semi-intrusive Monte Carlo

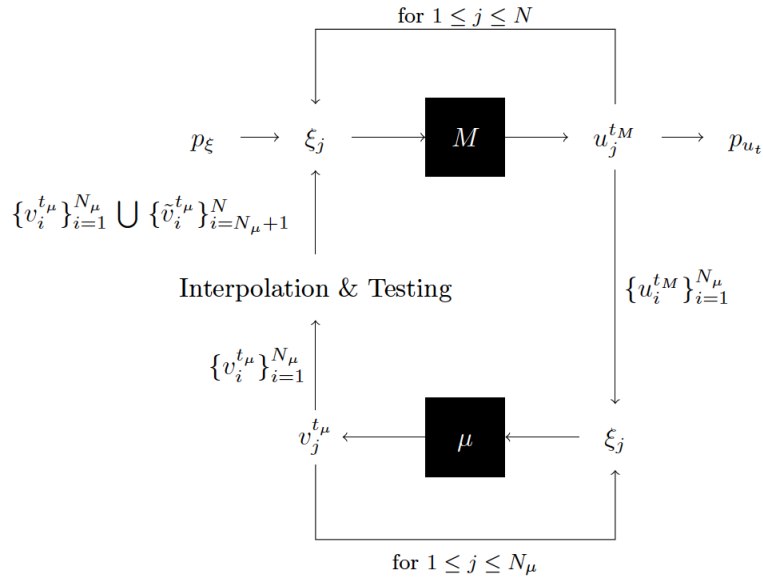


Figure 4: Semi-intrusive Monte Carlo method applied to a coupled-model application consisting of submodels M and μ . A smaller number of samples of the expensive submodel μ are simulated using advanced sampling.

Semi-intrusive Monte Carlo (SIMC) is a Monte Carlo method with a reduced number of samples of the expensive component of the multiscale model, see Figure 4. The remaining samples are obtained by interpolation. Usually the interpolation method produces results which are not exact to the micro model response. Therefore, a statistical cross-validation is applied to test whether the interpolation does not lead to a large error in the estimates of uncertainty: the error is compared to the confidence interval of the N_μ MC estimate, and then, our algorithm accepts the SIMC results when the error is smaller than the confidence interval and the MC results. All details can be found in [20].

5.2 Metamodeling of expensive sub-model

Surrogate modelling is a common approach to perform an efficient UQ for computationally intensive systems at a reduced amount of time [56, 18]. The idea of these methods is to substitute the original system

by its surrogate, much like the ones discussed in section 4, which produces a similar output, but their computational time is lower. In the semi-intrusive multiscale metamodeling method, these techniques are applied to a single scale component, which takes the largest portion of the computational time [22]. In this way, the error introduced by the approximation is expected to be small when estimating the uncertainties of the multiscale model.

Figure 5 shows an example where the micro model is substituted by a surrogate. The rest of the multiscale model has the original form. However, since the micro model produces an approximate result, the output of the macro model is not the same as the original model. In this method, the error will always depend on the details of the model. It depends on the properties of the micro model, for example, smoothness, which determines how difficult it will be to approximate the original single scale model. Additionally, the error in the estimates of uncertainty also depends on how sensitive the result of the macro model is to the output of the micro model which is replaced by a surrogate. If, for instance, this sensitivity is low, it is reasonable to expect that the error introduced by the approximation is small. Of course, the error also depends on the method with which the surrogate is build.

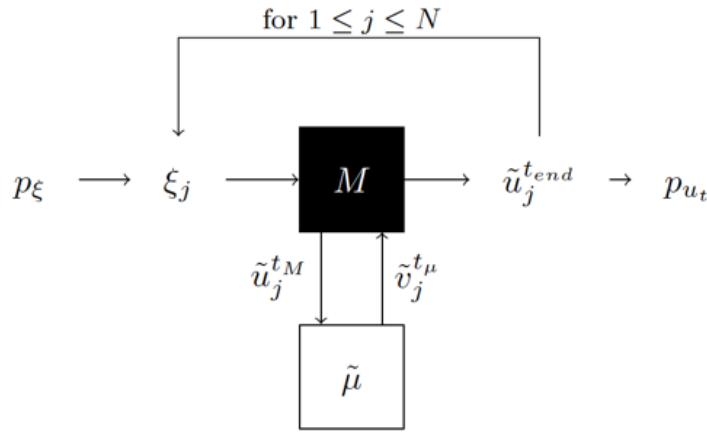


Figure 5: Semi-intrusive multiscale metamodeling uncertainty quantification. The expensive submodel μ is replaced by a cheaper surrogate model $\tilde{\mu}$ when computing ensembles of simulation of the complete application to perform UQ.

6 Uncertainty quantification toolkit for high-performance computing

Recent advances in the scale of computational resources available, and the algorithms designed to exploit them, mean that it is increasingly possible to conduct the additional sampling required by UQ even for highly complex calculations and workflows. EasyVVUQ is being developed as part of the VECMA project. The aim is to define stable interfaces and data formats that facilitate VVUQ in the widest range of applications. This would then provide the platform to support complex multi-solver workflows. Several software packages or libraries are already available for performing VVUQ (as shown in the next paragraph), but in many cases these rely on closed source components and none of them provide the separation of concerns needed to allow the analysis of both small local computations and highly compute intensive kernels (potentially using many thousands of cores and GPUs on HPC or cloud resources). Consequently, the design of EasyVVUQ is focused on making a wide range of VVUQ techniques available for scientists employing unmodified versions of existing applications. In particular, key considerations for us are the ability to support HPC codes, large job counts of the kind necessary for ensembles, as well as the robustness and restartability of workflows.

Several other toolkits share a subset of the added values that VECMAtk provides. In the area of VVUQ, a well-known toolkit is Design Analysis for Optimization and Terascale Applications (DAKOTA,

<https://dakota.sandia.gov>) [40], which provides a suite of algorithms for optimization, UQ, parameter studies, and model calibration. DAKOTA is a powerful tool but has a relatively steep learning curve due to the large number of tools available and offers no way to coordinate resources across concurrent runs [41]. Similarly, there are other toolkits that help with UQ directly, such as UQTK [42] and UQLab (<https://www.uqlab.com>) [43]. In the area of VVUQ using HPC, there are several other relevant tools. OpenTURNS [44] focuses on probabilistic modelling and uncertainty management, connects to HPC facilities, and provides calibration/Bayesian methods and a full set of interfaces to optimization solvers. Uranie leverages the ROOT framework (<http://root.cern.ch>) to support a wide range of UQ and sensitivity analyses (SA) activities using local and HPC resources. A key requirement for performing many types of UQ and SA is the ability to effectively run large ensembles of simulation runs. The “pilot job” mechanism allows a user to claim a large portion of a supercomputer into which a large and often complex set of individual jobs are submitted to form a workflow. In addition to QCG-PJ developed as part of the VECMAtk there are tools such as RADICAL-Cybertools [45] that can be used to initiate and manage large simulation ensembles on peta and emerging exascale supercomputers.

In the area of surrogate modelling, GPM/SA [46] helps to create surrogate models, calibrates them to observations of the system, and give predictions of the expected system response. At the Turing Institute, a Python package for fitting Gaussian Process Emulators to computer simulation results call MOGP is being developed (<https://github.com/alan-turing-institute/mogp-emulator>). There is also a portfolio of available solutions for rapidly processing user-defined experiments consisting of large numbers of relatively small tasks. The examples are Swift/T [47] and Parsl [48], both of which support execution of data-driven workflows. Another range of relevant related tools include more statistically oriented approaches. For instance, Uncertainpy [49] is a UQ and SA library that supports qMC and polynomial chaos expansions (PCE) methods. PSUADE [50] is a toolbox for UQ, SA and model calibration in non-intrusive ways [51], while DUE [52] assesses uncertain environmental variables, and generates realisations of uncertain data for use in uncertainty propagation analyses. PyMC3 [53] is a Python package for Bayesian statistical modelling and probabilistic machine learning which focuses on Markov Chain MC approaches and variational fitting. Similarly, SimLab (<https://ec.europa.eu/jrc/en/samo/simlab>) offers global UQ-SA based on non-intrusive MC methods. UQLab and SAFE [54] are MATLAB-based tools that provide support for UQ (using e.g. PCE) and SA (using e.g. Sobol’s method) respectively.

It is worth mentioning that capabilities of Uncertainpy have been integrated in EasyVVUQ. Indeed, it is possible to integrate many kinds of capabilities within EasyVVUQ, as it is designed to host VVUQ arbitrary applications that may be of interest now or in the future. This should be particularly convenient if currently un-featured UQ techniques are to be considered such as multilevel MC, or the mentioned HDMR techniques.

7 Review of UQ attempts on plasma fusion codes

The application of UQ to fusion simulation codes has been described in several papers, including "Validation in fusion research: Towards guidelines and best practices"[70], "Verification and validation for magnetic fusion" [3] and "Validation metrics for turbulent plasma transport" [4].

Although the UQ field has undergone rapid development over the past few years, its applications to plasma physics mainly focus on the two limits of Vlasov [71-73] and MHD [74, 75] with standard stochastic settings. Apart from the work in fluid dynamics [76, 77], to the best of the authors' knowledge, only limited work has been conducted on the propagation of uncertainty in multi-scale plasma physics.

Recently, the plasma community has recognized the importance of UQ in the validation and prediction of magnetically confined plasma turbulence [4]. Within the computational power afforded by current supercomputers at the time, the plasma community has explored the inclusion of UQ in the analysis of

reduced models, such as trapped-gyro-Landau-fluid (TGLF) equations, while UQ analysis in compute-intensive nonlinear simulations, e.g., gyrokinetic simulations, remains a challenging task. There is some previous literature concerning the inclusion of UQ in reduced model assessments and in the fitting of experimental measurements which includes but is not limited to [78, 79].

In more recent work, Calleja et al. [80] address a very concrete scenario: the analysis of first wall installations on the [DEMO](#) installation. An initial Monte Carlo study of the first wall is performed to develop understanding of the complex effects of tile misalignment. The Matlab toolkit [COSSAN](#) is used to perform the SA of the SMARRDA plasma modelling and simulation code. Another concrete application of UQ to DEMO by Lux et al [81] uses the [PROCESS](#) fusion power plant systems code. A multi-parameter Monte-Carlo method together with single parameter studies are performed to investigate individual impacts of performance parameters (net electric output and pulse length) on the fusion gain. Lakhilili et al. [82] performed the first UQ attempt on a multi-model (multiscale) fusion workflow, coupling a transport model of plasma profiles, a turbulence model of fluxes and an equilibrium model of plasma geometries. The UQ was performed using non-intrusive the polynomial chaos expansion.

Other advanced sampling techniques have been applied directly to plasma fusion simulations. Sensitivity-driven adaptive sparse stochastic approximations in plasma microinstability analysis was performed by Farcas et al [83]. They leveraged Sobol decompositions and introduced a sensitivity scoring system to drive the adaptive process. Their second test case was a real-world example stemming from a particular validation study for the [ASDEX Upgrade](#) experiment. They carried out a two-step analysis, initially considering three uncertain inputs characterizing the ions and electrons, and then 12 stochastic parameters associated with the particle species and the magnetic geometry. The results showed that the proposed approach has an accuracy comparable to the standard adaptive approach at significantly reduced computational cost; for example, for the 12D scenario, up to 13.3 fewer Gene evaluations.

Xiao et al. [84] introduced a stochastic kinetic scheme for multi-scale plasma transport with uncertainty quantification. They focused on the emergence, propagation and evolution of randomness from gyrations of charged particles in magnetohydrodynamic simulations. Solving Maxwell's equations with the wave-propagation method, the evolutions of ions, electrons and the electromagnetic field are coupled throughout the simulation. They combined the advantages of SG and SC methods with the construction principle of kinetic schemes, and obtained an efficient and accurate scheme for a cross-scale BGK-Maxwell system with uncertainties. Randomly initial inputs of both flow and electromagnetic fields are considered. Finally, point collocation methods have been used by Vaezi et al. [85] on simulations from a validation study of drift-wave turbulence in the CSDX linear plasma device experiment using [BOUT++](#) [86].

8 Conclusions

We have introduced three types of UQ approaches according to their intrusiveness with respect to the simulated application. At the single-model level, we discussed non-intrusive and intrusive methods. When considering applications coupling multiple models, we presented semi-intrusive methods which enable to save significant computational timing while avoiding interfering with the models equations. Non-intrusive and semi-intrusive methods appear to be interesting candidate keeping in mind that the Neptune projects seeks to promote the separation of concerns. While intrusive methods could circumvent the so-called curse of dimensionality, they would entail to implement UQ on a per-application basis. We have focused on two specific types of non-intrusive UQ methods: enhanced sampling and surrogate modelling.

Both techniques are already available from Neptune partners, that is UCL and the Turing Institute with tools such as EasyVVUQ, EasySurrogate and MOGP Emulator. We introduced these tools as part of a

larger review of (VV)UQ toolkits available which implement such methods and render these available for widespread use on high-performance computing infrastructures.

This report lays the foundations of methods that will be further investigated and tested during the duration of this 6-month project. Following the first meetings on the theme of UQ which were held mid-January (workshop and hackathon), enhanced sampling techniques as well as actionability of EasyVVUQ workflows has been tested by a subset of Neptune application partners. On the basis of their experience as well as expectations from the whole project's community, we will attempt to conclude on a precise list of methods and toolkits to integrate UQ at the heart of the future Neptune code.

9 References

- [1] Oberkampf, William L., and Timothy G. Trucano. "Verification and validation in computational fluid dynamics." *Progress in Aerospace Sciences* 38.3 (2002): 209-272. doi:10.1016/S0376-0421(02)00005-2
- [2] Terry, P. W., et al. "Task group on verification and validation, US burning plasma organization, and US transport task force." *Phys. Plasmas* 15 (2008): 062503.
- [3] Greenwald, Martin. "Verification and validation for magnetic fusion." *Physics of Plasmas* 17.5 (2010): 058101. doi:10.1063/1.3298884
- [4] Holland, C. "Validation metrics for turbulent plasma transport." *Physics of Plasmas* 23.6 (2016): 060901. doi:10.1063/1.4954151
- [5] A.G. Hoekstra, B. Chopard, D. Coster, S. Portegies Zwart, P. V. Coveney, Multiscale computing for science and engineering in the era of exascale performance, *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 377 (2019) 20180144. doi:10.1098/rsta.2018.0144.
- [6] S. Alwayyed, D. Groen, P. V. Coveney, A.G. Hoekstra, Multiscale Computing in the Exascale Era, *J. Comput. Sci.* 22 (2017) 15–25. doi:10.1016/j.jocs.2017.07.004.
- [7] A.G. Hoekstra, B. Chopard, P. V. Coveney, Multiscale modelling and simulation: a position paper, *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 372 (2014) 20130377. doi:10.1098/rsta.2013.0377.
- [8] P.M.A. Sloot, A.G. Hoekstra, Multi-scale modelling in computational biomedicine, *Br. Bioinform.* 11 (2010) 142–152. doi:10.1093/bib/bbp038.
- [9] S. Karabasov, D. Nerukh, A.G. Hoekstra, B. Chopard, P. V. Coveney, Multiscale modelling: approaches and challenges, *Philos. Trans. R. Soc. A.* 372 (2014) 20130390. doi:10.1098/rsta.2013.0390.
- [10] R.H. Johnstone, E.T.Y. Chang, R. Bardenet, T.P. de Boer, D.J. Gavaghan, P. Pathmanathan, et al., Uncertainty and variability in models of the cardiac action potential: Can we build trustworthy models?, *J. Mol. Cell. Cardiol.* 96 (2016) 49–62. doi:10.1016/J.YJMCC.2015.11.018.
- [11] R.C. Smith, *Uncertainty quantification: theory, implementation, and applications*, SIAM, 2013.
- [12] O.P. Le Maître, O.M. Knio, *Spectral Methods for Uncertainty Quantification*, Springer Netherlands, Dordrecht, 2010. doi:10.1007/978-90-481-3520-2.
- [13] X. Wan, G.E. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, *J. Comput. Phys.* 209 (2005) 617–642. doi:10.1016/J.JCP.2005.03.023.
- [14] Groen, D., et al. "VECMAtk: A Scalable Verification, Validation and Uncertainty Quantification toolkit for Scientific Simulations." *arXiv preprint arXiv:2010.03923* (2020).
- [15] Richardson, R. A., Coveney, P.V., et al. "EasyVVUQ: A Library for Verification, Validation and Uncertainty Quantification in High Performance Computing." *Journal of Open Research Software* 8.1 (2020). doi:10.5334/JORS.303
- [16] R.C. Smith, *Uncertainty quantification: theory, implementation, and applications*, SIAM, 2013.

- [17] Babuska, Ivo, Raúl Tempone, and Georgios E. Zouraris. "Galerkin finite element approximations of stochastic elliptic partial differential equations." *SIAM Journal on Numerical Analysis* 42.2 (2004): 800-825. doi:10.1137/S0036142902418680
- [18] Ghanem, Roger G., and Pol D. Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.
- [19] Giraldi, Loïc, et al. "To be or not to be intrusive? The solution of parametric and stochastic equations---the "plain vanilla" Galerkin case." *SIAM Journal on Scientific Computing* 36.6 (2014): A2720-A2744. doi:10.1137/130942802
- [20] Nikishova, Anna, and Alfons G. Hoekstra. "Semi-intrusive uncertainty propagation for multiscale models." *Journal of Computational Science* 35 (2019): 80-90. doi:10.1016/j.jocs.2019.06.007
- [21] Giles, Michael B. "Multilevel monte carlo path simulation." *Operations research* 56.3 (2008): 607-617. doi:10.1287/opre.1070.0496
- [22] Jantsch, Peter, Clayton G. Webster, and Guannan Zhang. "On the Lebesgue constant of weighted Leja points for Lagrange interpolation on unbounded domains." *IMA Journal of Numerical Analysis* 39.2 (2019): 1039-1057. doi:10.1093/imanum/dry002
- [23] Babuška, Ivo, Fabio Nobile, and Raúl Tempone. "A stochastic collocation method for elliptic partial differential equations with random input data." *SIAM Journal on Numerical Analysis* 45.3 (2007): 1005-1034. doi: 10.1137/050645142
- [24] Barthelmann, Volker, Erich Novak, and Klaus Ritter. "High dimensional polynomial interpolation on sparse grids." *Advances in Computational Mathematics* 12.4 (2000): 273-288. doi:10.1023/A:1018977404843
- [25] Blatman, Géraud, and Bruno Sudret. "Adaptive sparse polynomial chaos expansion based on least angle regression." *Journal of computational Physics* 230.6 (2011): 2345-2367. doi:10.1016/j.jcp.2010.12.021
- [26] Migliorati, Giovanni, et al. "Approximation of quantities of interest in stochastic PDEs by the random discrete L2 projection on polynomial spaces." *SIAM Journal on Scientific Computing* 35.3 (2013): A1440-A1460.
- [27] Eldred, Michael, and John Burkardt. "Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification." 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition. 2009. doi:10.2514/6.2009-976
- [28] Bellman, R. "Dynamic Programming". Princeton University Press, Princeton, New Jersey (1957).
- [29] Chkifa, Abdellah, Albert Cohen, and Christoph Schwab. "High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs." *Foundations of Computational Mathematics* 14.4 (2014): 601-633. doi:10.1007/s10208-013-9154-z
- [30] Narayan, Akil, and John D. Jakeman. "Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation." *SIAM Journal on Scientific Computing* 36.6 (2014): A2952-A2983. doi:10.1137/140966368
- [31] Hackbusch, Wolfgang. "Numerical tensor calculus." *Acta Numer.* 23 (2014): 651-742.
- [32] Uschmajew, André. "Zur Theorie der Niedrigrangapproximation in Tensorprodukten von Hilberträumen." (2013).
- [33] N. Verheul, D. Crommelin, Data-driven stochastic representations of unresolved features in multiscale models, *Communications in Mathematical Sciences*, Vol. 14 (2016), pp 1213 – 1236. doi:10.4310/CMS.2016.v14.n5.a2
- [34] W. Edeling, D. Crommelin, Towards data-driven dynamics surrogate models for ocean flow. In: *Proceedings of the Platform for Advanced Scientific Computing Conference 2019 (PASC '19)*, Zurich, Switzerland. ACM, New York, NY, USA. doi:10.1145/3324989.3325713
- [35] D. Crommelin, W. Edeling, Resampling with neural networks for stochastic parameterization in multiscale systems, 2020. arXiv preprint arXiv:2004.01457
- [36] Ambrogioni, L., Güçlü, U., van Gerven, M. A., & Maris, E. The kernel mixture network: A nonparametric method for conditional density estimation of continuous random variables, 2017. arXiv preprint arXiv:1705.07111.

- [37] W. Edeling, D. Crommelin, Reduced model-error source terms for fluid flow, Proceedings of UNCECOMP 2019 conference. doi:10.13140/RG.2.2.36773.35048
- [38] W. Edeling, D. Crommelin, Reducing data-driven dynamical subgrid scale models by physical constraints, Computers and Fluids, 2020. doi:10.1016/j.compfluid.2020.104470
- [39] Groen, Derek, et al. "Introducing VECMAtk-verification, validation and uncertainty quantification for multiscale and HPC simulations." International Conference on Computational Science. Springer, Cham, 2019. doi:10.1007/978-3-030-22747-0_36
- [40] Adams, Brian M., et al. "DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.0 user's manual." Sandia National Laboratories, Tech. Rep. SAND2010-2183 (2009). doi:10.2172/991841
- [41] Foley, Samantha S., et al. "Many-task applications in the Integrated Plasma Simulator." 2010 3rd Workshop on Many-Task Computing on Grids and Supercomputers. IEEE, 2010. doi:10.1109/MTAGS.2010.5699425
- [42] Debusschere, Bert, et al. UQTK: A Flexible Python/C++ Toolkit for Uncertainty Quantification. No. SAND2018-4226C. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2018.
- [43] Marelli, Stefano, and Bruno Sudret. "UQLab: A framework for uncertainty quantification in Matlab." Vulnerability, uncertainty, and risk: quantification, mitigation, and management. 2014. 2554-2563. doi:10.1061/9780784413609.257
- [44] Baudin, Michaël, et al. "Open TURNS: An industrial software for uncertainty quantification in simulation." arXiv preprint arXiv:1501.05242 (2015).
- [45] Balasubramanian, Vivek, et al. "Radical-cybertools: Middleware building blocks for scalable science." arXiv preprint arXiv:1904.03085 (2019).
- [46] Gattiker, J., et al. "Gaussian process-based sensitivity analysis and Bayesian model calibration with GPMSA." Handbook of uncertainty quantification (2017): 1-41. doi:10.1007/978-3-319-11259-6_58-1
- [47] Wozniak, Justin M., et al. "Swift/t: Large-scale application composition via distributed-memory dataflow processing." 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing. IEEE, 2013. doi:10.1109/CCGrid.2013.99
- [48] Babuji, Yadu, et al. "Parsl: Pervasive parallel programming in python." Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing. 2019. doi:10.1145/3307681.3325400
- [49] Tennøe, Simen, Geir Halnes, and Gaute T. Einevoll. "Uncertainpy: a python toolbox for uncertainty quantification and sensitivity analysis in computational neuroscience." Frontiers in neuroinformatics 12 (2018): 49. doi: 10.3389/fninf.2018.00049
- [50] Lawrence Livermore National Laboratory. 2016. Non-Intrusive Uncertainty Quantification: PSUADE. <https://computing.llnl.gov/projects/psuade-uncertainty-quantification>
- [51] Hittinger, J. A., B. I. Cohen, and R. I. Klein. Uncertainty Quantification in the Fusion Simulation Project Verification and Validation Activity. No. LLNL-TR-458089. Lawrence Livermore National Lab. (LLNL), Livermore, CA (United States), 2010. doi:10.2172/1119966
- [52] Brown, James D., and Gerard BM Heuvelink. "The Data Uncertainty Engine (DUE): A software tool for assessing and simulating uncertain environmental variables." Computers & Geosciences 33.2 (2007): 172-190. doi:10.1016/j.cageo.2006.06.015
- [53] Salvatier, John, Thomas V. Wiecki, and Christopher Fonnesbeck. "Probabilistic programming in Python using PyMC3." PeerJ Computer Science 2 (2016): e55. doi:10.7717/peerj-cs.55
- [54] Pianosi, Francesca, Fanny Sarrazin, and Thorsten Wagener. "A Matlab toolbox for global sensitivity analysis." Environmental Modelling & Software 70 (2015): 80-85. doi:10.1016/j.envsoft.2015.04.009
- [55] B. Chopard, J. Borgdorff, A.G. Hoekstra, A framework for multi-scale modelling, Philos. Trans. R. Soc. A. 372 (2014) 20130378. doi:10.1098/rsta.2013.0378.
- [56] R. Archibald, M. Chakoumakos, T. Zhuang, Characterizing the elements of Earth's radiative budget: Applying uncertainty quantification to the CESM, Procedia Comput. Sci. 9 (2012) 1014–1020. doi:10.1016/J.PROCS.2012.04.109.
- [57] N.E. Owen, P. Challenor, P.P. Menon, S. Bennani, Comparison of Surrogate-Based Uncertainty Quantification Methods for Computationally Expensive Simulators, SIAM/ASA J. Uncertain. Quantif. 5 (2017) 403–435. doi:10.1137/15M1046812.

- [58] A. Nikishova, L. Veen, P. Zun, A.G. Hoekstra, Semi-intrusive multiscale metamodelling uncertainty quantification with application to a model of in-stent restenosis, *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 377 (2019) 20180154. doi:10.1098/rsta.2018.0154.
- [59] Rabitz, Herschel, and Ömer F. Aliş. "General foundations of high-dimensional model representations." *Journal of Mathematical Chemistry* 25.2 (1999): 197-233.
- [60] Ma, Xiang, and Nicholas Zabarar. "An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations." *Journal of Computational Physics* 229.10 (2010): 3884-3915.
- [61] Ma, Xiang, and Nicholas Zabarar. "An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations." *Journal of Computational Physics* 228.8 (2009): 3084-3113.
- [62] Witteveen, Jeroen AS, and Gianluca Iaccarino. "Simplex stochastic collocation with ENO-type stencil selection for robust uncertainty quantification." *Journal of Computational Physics* 239 (2013): 1-21.
- [63] Edeling, Wouter Nico, Richard P. Dwight, and Paola Cinnella. "Simplex-stochastic collocation method with improved scalability." *Journal of Computational Physics* 310 (2016): 301-328.
- [64] Constantine, Paul G., Eric Dow, and Qiqi Wang. "Active subspace methods in theory and practice: applications to kriging surfaces." *SIAM Journal on Scientific Computing* 36.4 (2014): A1500-A1524.
- [65] Liu, Xiaoyu, and Serge Guillas. "Dimension reduction for Gaussian process emulation: An application to the influence of bathymetry on tsunami heights." *SIAM/ASA Journal on Uncertainty Quantification* 5.1 (2017): 787-812.
- [66] Tripathy, Rohit, and Ilias Biliotis. "Deep active subspaces: A scalable method for high-dimensional uncertainty propagation." *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2019.
- [67] Tripathy, Rohit K., and Ilias Biliotis. "Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification." *Journal of computational physics* 375 (2018): 565-588.
- [68] Rasp, Stephan. "Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: general algorithms and Lorenz 96 case study (v1. 0)." *Geoscientific Model Development* 13.5 (2020): 2185-2196.
- [69] Edeling, Wouter, et al. "Model uncertainty and decision making: Predicting the Impact of COVID-19 Using the CovidSim Epidemiological Code." (2020).
- [70] Greenwald, Martin. "Beyond benchmarking—how experiments and simulations can work together in plasma physics." *Computer physics communications* 164.1-3 (2004): 1-8.
- [71] Hu, Jingwei, Shi Jin, and Ruiwen Shu. "A Stochastic Galerkin Method for the Fokker–Planck–Landau Equation with Random Uncertainties." *XVI International Conference on Hyperbolic Problems: Theory, Numerics, Applications*. Springer, Cham, 2016.
- [72] Jin, Shi, and Yuhua Zhu. "Hypocoercivity and Uniform Regularity for the Vlasov–Poisson–Fokker–Planck System with Uncertainty and Multiple Scales." *SIAM Journal on Mathematical Analysis* 50.2 (2018): 1790-1816.
- [73] Ding, Zhiyan, and Shi Jin. "Random regularity of a nonlinear Landau Damping solution for the Vlasov-Poisson equations with random inputs." *International Journal for Uncertainty Quantification* 9.2 (2019).
- [74] Phillips, Edward G., and Howard C. Elman. "A stochastic approach to uncertainty in the equations of MHD kinematics." *Journal of Computational Physics* 284 (2015): 334-350.
- [75] Yamazaki, Kazuo. "Stochastic Hall-magneto-hydrodynamics system in three and two and a half dimensions." *Journal of Statistical Physics* 166.2 (2017): 368-397.
- [76] Hu, Jingwei, and Shi Jin. "Uncertainty quantification for kinetic equations." *Uncertainty quantification for hyperbolic and kinetic equations*. Springer, Cham, 2017. 193-229.
- [77] Liu, Liu, and Shi Jin. "Hypocoercivity based sensitivity analysis and spectral convergence of the stochastic Galerkin approximation to collisional kinetic equations with multiple scales and random inputs." *Multiscale Modeling & Simulation* 16.3 (2018): 1085-1114.

- [78] White, A. E., et al. "Measurements of the cross-phase angle between density and electron temperature fluctuations and comparison with gyrokinetic simulations." *Physics of Plasmas* 17.5 (2010): 056103.
- [79] Chilenski, M. A., et al. "Improved profile fitting and quantification of uncertainty in experimental measurements of impurity transport coefficients using Gaussian process regression." *Nuclear Fusion* 55.2 (2015): 023012.
- [80] Calleja, D. C., et al. "Strategy for Sensitivity Analysis of DEMO first wall." Proceedings of the joint ICVRAM ISUMA UNCERTAINTIES conference, Florianopolis, SC, Brazil. 2018.
- [81] Lux, H., et al. "Implications of uncertainties on European DEMO design." *Nuclear Fusion* 59.6 (2019): 066012.
- [82] Lakhili, J., et al. "Uncertainty Quantification for Multiscale Fusion Plasma Simulations with VECMA Toolkit." International Conference on Computational Science. Springer, Cham, 2020.
- [83] Farçaş, I., et al. "Sensitivity-driven adaptive sparse stochastic approximations in plasma microinstability analysis." *Journal of Computational Physics* 410 (2020): 109394.
- [84] Xiao, T., and Frank, M. "A stochastic kinetic scheme for multi-scale plasma transport with uncertainty quantification." *Journal of Computational Physics* (2021): 110139.
- [85] Vaezi, P., and Christopher Holland. "An improved approach to uncertainty quantification for plasma turbulence validation studies." *Fusion Science and Technology* 74.1-2 (2018): 77-88.
- [86] Vaezi, P., et al. "Validation study of a drift-wave turbulence model for CSDX linear plasma device." *Physics of Plasmas* 24.9 (2017): 092310.