Uncertainty quantification for multiscale fusion plasma simulations with VECMA toolkit

Jalal Lakhlili, Olivier Hoenen, Onnie O. Luk, and David P. Coster

Max-Planck Institute for Plasma Physics,
Boltzmannstrasse 2, 85748 Garching, Germany
jalal.lakhlili@ipp.mpg.de

Abstract. Within VECMAtk platform we perform Uncertainty Quantification (UQ) for multiscale fusion plasmas simulations. The goal of VECMAtk is to enable modular and automated tools for a wide range of applications to archive robust and actionable results. Our aim in the current paper is to incorporate suitable features to build UQ workflow over the existing fusion codes and to tackle simulations on high performance parallel computers.

Keywords: Fusion plasmas · Uncertainty quantification · Sensitivity analysis · Polynomial chaos expansion · Quasi-Monte Carlo method · Multiscale modeling · High performance computing.

1 Introduction

Thermonuclear fusion can potentially provide a carbon free and safe solution to the provision of base load electricity. Heat and particle transport will play the key roles in determining the size and efficiency of future fusion power reactor, so understanding their mechanisms is an important milestone towards the realization of fusion-based electricity. Our present understanding is that turbulence at small space and time scales is responsible for much of this transport, but the profiles of temperature and density evolve over much larger space and time scales. To study these phenomena, multiscale and multiphysics applications have been developed throughout the years. Notable efforts in Europe lean toward implementing such applications by following a workflow approach, in which several *single-scale* (or single-physics) codes are coupled together [1, 2].

When one element in such a workflow is a turbulence code, whose outputs are inherently noisy due to the chaotic nature of the turbulence (intrinsic uncertainties), or if input data such as external sources contain uncertainties (extrinsic), these need to be propagated through different models of the workflow. The goal, therefore, is to produce temperature and density profiles, along with their confidence intervals. This information will then allow for an improved validation of the simulation results against the experimental measurements that come with error bars.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_53

Uncertainty quantification in the existing fusion workflow is implemented with the VECMA open source toolkit[1], which provides tools to facilitate the verification, validation and uncertainty quantification (VVUQ) process in multiscale, multiphysics applications [3]. So far we have integrated two elements from this toolkit: the EasyVVUQ library [4] to incorporate uncertainty quantification (UQ) and sensitivity analysis (SA) in a non-intrusive manner, and the QCG Pilot-Job middleware [5] to execute efficiently the large number of tasks or samples prepared by EasyVVUQ.

In this work we focus on the different sampling methods used to quantify the uncertainties, their results and performance for two different black box use cases: on the entire workflow for extrinsic uncertainties coming from the heating source terms and boundary conditions for temperature profiles (on electron and ions), and on the turbulence model for uncertain ions and electron temperatures values and gradients. This paper is organized as follows. After a brief overview on the background theory for uncertainty quantification, we present the various sampling methods used in our application that are also available in the EasyVVUQ. We then describe our two use cases together with UQ and SA results for the method that demonstrates the best performance. Finally, we conclude with future works related to more intrusive approaches.

2 Theory overview

We consider a numerical model $\mathcal{M}$ that has a number d of uncertain input parameters $Q_1, Q_2, ..., Q_d$, and gives the output Y:

$$Y = \mathcal{M}(x, Q). \tag{1}$$

Here, Q is the vector $[Q_1, Q_2, ..., Q_d]$ and x is a space variable. The model can also depends on other variables (e.g. time) and on additional but invariant input parameters. In this study we focus on the uncertain parameters $(Q_i)_{i=1}^{d}$ and how the uncertainty is propagated through the model $\mathcal{M}$. That way we can quantify the uncertainty in the output Y and identify the relative contribution of each Q_i into this uncertain result.

First, we assume that the uncertain parameters are statistically independent components, and each one has an univariate probability density function f_{Q_i}, hence the vector Q can be described by the joint probability density function:

$$f_Q = \prod_{i=1}^{d} f_{Q_i}. \tag{2}$$

Thereafter, we seek relevant properties of the output Y that characterize the uncertainty propagation and where possible, we build an approximation of the unknown output distribution f_Y.

[1] www.vecma-toolkit.eu

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_53

2.1 Statistical moments

The principal aim of UQ is to describe the output distribution f_Y through useful statistics [6, 7]. Several metrics are based on statistical moments, and among these the most commonly used are the mean $\mathbb{E}$ and the variance $\mathbb{V}$:

$$\mathbb{E}[Y] = \int_{\Omega_Y} y f_Y(y) dy, \tag{3}$$

$$\mathbb{V}[Y] = \int_{\Omega_Y} (y - \mathbb{E}[Y])^2 f_Y(y) dy. \tag{4}$$

Here, Ω_Y is the output space.

The mean indicates the expected value of the model output Y and the variance measures how much the output varies around the mean. In general, instead of using the variance, we take its square root to obtain the standard deviation σ:

$$\sigma[Y] = \sqrt{\mathbb{V}[Y]}. \tag{5}$$

The advantage of using σ instead of the variance is that, like the mean, σ has the same units as Y and that makes it easier to interpret.

Another useful measure for UQ is percentiles which can be used to define confidence intervals [6].

2.2 Sensitivity Analysis

The principal aim of SA is to evaluate the contribution of different inputs parameters $(Q_i)_{i=1}^{d}$ to the uncertainty in output Y. Among the various sensitivity metrics that exist, we choose in this study the variance-based sensitivity analysis by computing the Sobol indices [8, 9]. It is the most commonly used metric as it allows to measure sensitivity across the whole input space and can cope with nonlinear outputs.

There are several orders of Sobol indices. Here, we select two of the most-used indices in UQ: the first-order Sobol index S_i and the total order index S_{Ti}. They measure the direct effect of the parameter Q_i on the variance of Y and the interactions between Q_i and other parameters, respectively:

$$S_i = \frac{\mathbb{V}[\mathbb{E}[Y|Q_i]]}{\mathbb{V}[Y]}, \tag{6}$$

$$S_{Ti} = 1 - \frac{\mathbb{V}[\mathbb{E}[Y|Q_{-i}]]}{\mathbb{V}[Y]}. \tag{7}$$

We denote by $\mathbb{E}[Y|Q_i]$ the conditional mean, which represents the expected value of Y for a fixed value of Q_i, and by Q_{-i} all uncertain parameters except Q_i.

The first-order Sobol index, also known as the main effect index, cannot exceed one, and if it is close to zero we can claim that its input parameter have a small or no effect on the output. The higher the index value is, the more influence the associated parameter has on the resulting uncertainty. For the total Sobol indices, their sum is equal to or greater than one, and it is only equal to one if there is no interaction between the parameters [10].

3 Sampling approaches

Sampling is the usual computational method used to estimate uncertainty measures, statistical moments and sensitivity as they have been described in the previous section. We can choose between several widely used sampling methods for a non-intrusive (or black box) approach: Quasi-Monte Carlo (QMC) [11] or Polynomial Chaos Expansion (PCE) [12]. This choice can be motivated by the numerical cost of such estimation, which is proportional to the number of samples and mainly depends on the number of uncertain input parameters.

3.1 Quasi-Monte Carlo

Monte Carlo is one of the most widespread method due to its simplicity and ease of implementation. The basic idea of the method is a purely random selection of sample evaluations without assumptions about the model. The number of samples is independent of the number of uncertain parameters, but a very high number is required in order to obtain reliable statistics, thus UQ estimations can rapidly become computationally expensive.

QMC methods are different approaches that aim at improving Monte Carlo method. They are based on variance-reduction techniques to reduce the number of model evaluations needed [11, 13]. In this work, instead of random selection of samples from f_Q, we use Sobol sequences which have a low-discrepancy sequence [8]. Using N_s samples to evaluate the model $\mathcal{M}$, which gives the output results $Y = [Y_1, Y_2, ..., Y_{N_s}]$, the mean and the variance can approximated by:

$$\mathbb{E}[Y] \approx \frac{1}{N_s} \sum_{i=1}^{N_s} Y_i, \tag{8}$$

$$\mathbb{V}[Y] \approx \frac{1}{N_s - 1} \sum_{i=1}^{N_s} (Y_i - \mathbb{E}[Y])^2. \tag{9}$$

For the SA, we use Saltelli's procedure [9,14]. It is also based on Sobol sequences and to obtain a full set of main and total sensitivity indices, it is sufficient to take:

$$N_s = \frac{(d+2)N}{2}, \tag{10}$$

where N is the number of samples required to get a given accuracy.

3.2 Polynomial Chaos Expansion

The PCE method is one of most competitive alternatives to (Quasi-)Random methods. The output Y is expanded into a series of orthogonal polynomials $(P_j)_j$ of degree p, which are functions of the input parameters Q and are chosen such that they are orthogonal to the input distributions [7, 15–17]:

$$Y \approx \widehat{Y}(Q) = \sum_{j=1}^{N_p} c_j P_j(Q). \tag{11}$$

Here, N_p is the number of expansion factors, which depends on both the degree p and the number of the uncertain parameters d, and is given by the binomial coefficient:

$$N_p = \binom{d+p}{d}; \tag{12}$$

And $(c_j)_j$ are the expansion coefficients, which can be estimated using tensor product quadrature or linear regression. These two methods that will be detailed later in this section.

By exploiting the properties of the orthogonal polynomials, the expectation value and the variance for the output model are:

$$\mathbb{E}[Y] \approx c_1, \tag{13}$$

$$\mathbb{V}[Y] \approx \sum_{j=2}^{N_p} \gamma_j c_j \tag{14}$$

Here, γ_j is a normalization factor and it is defined as: $\gamma_j = \mathbb{E}[P_j^2(Q)]$.

For the sensitivity analysis, the first and total-order Sobol indices can also be computed using Sobol's variance decomposition and by exploiting the polynomial chaos expansion properties [16, 18].

The estimation of the expansion coefficients is the most important step in the PCE construction. As mentioned before, we use two of the most common methods: spectral projection with tensor product quadrature and linear regression.

Spectral projection In this PCE variant, each expansion coefficient c_j is estimated by projecting the solution onto the output space Ω_Y. This is analogous to the calculation of Fourier coefficients in the approximation of periodic function. By exploiting the orthogonality of the polynomial, equation 11 yields:

$$c_j = \frac{1}{H_j} \int_{\Omega_Y} Y(q) P_j(q) f_Q(q) dq, \tag{15}$$

where H_j is a normalization factor associated with the polynomial P_j.

To compute the integral from equation 15, we use quadrature generators with weights $(\omega_k)_k$ and nodes $(q_k)_k$, and then we evaluate the outputs at those nodes. As the integrands are smooth polynomials, c_j can be approximated with a good accuracy by the expression:

$$c_j \approx \frac{1}{H_j} \sum_{k=1}^{N_s} Y(q_k) P_j(q_k) \omega_k. \tag{16}$$

The number of samples N_s in this case is determined by the quadrature rule. For our applications, we use tensored quadrature with Gaussian schemes and in this case N_s is:

$$N_s = (p+2)^d. \tag{17}$$

DOI: 10.1007/978-3-030-50436-6_53

As the required number of samples increases exponentially with the number of uncertain parameters, this method is beneficial only for a relatively small number of parameters.

Regression This variant, also known as point collocation, uses a single linear least square approximation of the form:

$$\begin{bmatrix} P_1(q_1) & P_2(q_1) & \dots & P_{N_p}(q_1) \\ P_1(q_2) & P_2(q_2) & \dots & P_{N_p}(q_2) \\ \vdots & \vdots & \ddots & \vdots \\ P_1(q_{N_s}) & P_2(q_{N_s}) & \dots & P_{N_p}(q_{N_s}) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N_p} \end{bmatrix} = \begin{bmatrix} Y(q_1) \\ Y(q_2) \\ \vdots \\ Y(q_{N_s}) \end{bmatrix} \tag{18}$$

The set of linear equations above results from the expansion of equation 11 at a set of collocation nodes $(q_k)_{k=1}^{N_s}$. To get stable nodes, we use pseudo-random samples generated from the input distribution f_Q with specific rules as indicated in Chaospy [19]. To avoid an under-determined system, we should choose a number of samples such that $N_s \geq N_p$. According to [20], it is recommended that we use at least $2N_p$ samples. Therefore, we set:

$$N_s = 2\frac{(p+d)!}{p!d!}. \tag{19}$$

In this case, we obtain a good least square approximation which becomes significantly more affordable than the full tensor product quadrature used in the spectral projection.

3.3 Implementation and choice

All these sampling methods and variants are available, among others, in the EasyVVUQ[2] library; it provides a simple decomposition of UQ and SA as generic steps (*sampling, encoder, decoder, collation, analysis*) in a modular workflow approach. Sampling and analysis methods are mainly based on existing Python packages such as Chaospy [19], which provides probability distributions, statistics, quadrature generators and SA for PCE, and SALib [21] for SA with the QMC method. This decomposition allows users to exchange sampling methods easily without impacting the overall structure of the application.

In terms of sample sizes, we can use PCE with spectral projection as long as the number of uncertain parameters is low, typically smaller than 6. For a moderate number of parameters, PCE with regression becomes advantageous, and for a high number of parameters, quasi-Monte Carlo is the only method that is applicable on current computing platforms. In terms of accuracy, for smooth problems, PCE with spectral projection is expected to be more efficient compared with the regression variant or QMC. A more in-depth analysis and comparison between these methods and others can be found in [16, 19, 20].

[2] github.com/UCL-CCS/EasyVVUQ

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_53

4 Numerical applications and results

To perform UQ and SA on any multiscale application that couples single-scale models as a workflow, various level of intrusiveness can be considered. In a non-intrusive approach, either the entire workflow is treated as a black box, or only one single-scale model is being studied and treated as a black box. In a semi-intrusive approach, each single scale code is placed inside a black box, and UQ and SA are performed on the entire workflow.

In this section, we will present the two use cases for non-intrusive approach with the fusion multiscale application, and discuss their results as well as some performance considerations.

4.1 Multiscale fusion workflow as a black box

The multiscale fusion workflow[3] was created to study the effects of turbulence happening at the micro time and space scales on the evolution of temperature profiles at the macro level of the fusion device. Currently, the workflow combines three main models as shown in Fig. 1: an equilibrium code that describes the plasma geometry, a turbulence code that approximates transport coefficient, and a transport solver that evolves temperature profiles at the macro time scale.

When considering the entire workflow as a black box, we use a simpler analytical model for the turbulence to reduce drastically the computational cost of the simulation. The extrinsic uncertainties we consider are:

- Boundary conditions defining the electron temperature at the plasma edge.
- Simplified heating sources, for which electron heating power Gaussian distributions are characterized by their amplitude, width, and position.

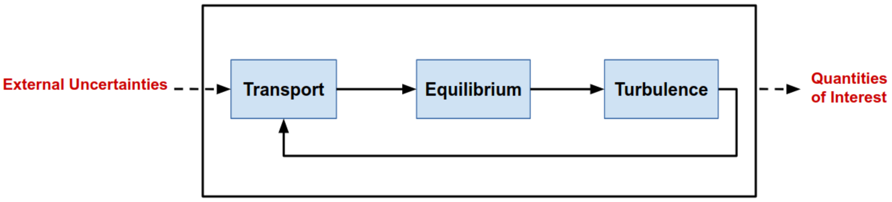

Fig. 1. Sketch of the targeted fusion workflow diagram

We select the PCE method, described in section 3, and assume that each of the uncertain parameters has a normal distribution in the range of ±20% around its original value. The workflow runs within a black-box for multiple time iterations until the plasma reaches steady state [2, 22]. The quantities of interest

[3] github.com/vecma-ipp/MFW

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_53

are the outputs Te and Ti, the electron and ion temperatures respectively, coming from the macroscopic transport model. The outcome from the EasyVVUQ analysis is plotted in Fig. 2.

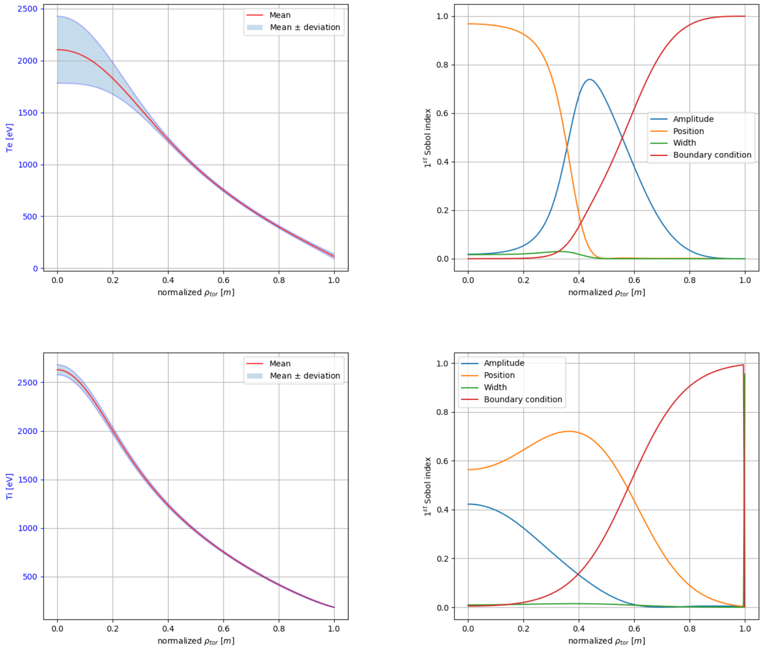

Fig. 2. Descriptive statistics and sensitivity analysis for electron and ion temperatures: on the left hand-side, the expected value and standard deviation are shown with respect to the normalized toroidal flux coordinate ρ_{tor}, and on the right hand-side the first-order Sobol indices are plotted for each of the uncertain parameters.

The standard deviation indicates that the ion temperature varies weakly since the uncertainties are carried by the electrons sources. The sensitivity analysis reveals that the variance in the electron and ion temperatures is mainly due to the uncertainty from three parameters: the position and amplitude parameters of the sources at core region of the plasma ($\rho_{tor} = 0$) and, as expected, boundary condition parameter at the edge region ($\rho_{tor} = 1$). The parameter width has no direct effect on the variance of the two quantities, so according to [23], this parameter can be neglected and then the number of samples can be reduced while keeping the same variance behavior.

We did the same simulation using uncertain parameters from heating sources and temperature boundary conditions for ions, and both electrons and ions. In all cases the UQ and SA results are qualitatively the same.

This use case was performed using both PCE variants: spectral projection and regression. They required 1296 and 140 samples, respectively, because we have 4 uncertain parameters and use quadratic polynomials. The results are qualitatively the same, and the differences between output means and variances of each method are at the order of $\mathcal{O}(10^{-6})$.

4.2 3D gyrofluid turbulence as a black box

In the workflow, given a plasma state and the geometry, the turbulence code provides the transport coefficients by computing the turbulent fluxes of particles and energy. Due to the chaotic nature of the turbulence model, these outputs are inherently noisy. And running such stochastic model is computationally expensive. Therefore, we begin the UQ study by isolating a 3D gyrofluid turbulence model that computes heat and particle fluxes [24] and consider one flux tube. We vary the electron and ion temperatures, Te and Ti, and their respective gradients, ∇Te and ∇Ti, at the position of the flux tube.

We introduce four uncertain input parameters into the black box. Similar to the previous study, we assume that each parameter has a normal distribution in the range of ±20% around its original value. Finally, we perform the sensitivity analysis to explore how these parameters affect the resulting heat fluxes that determine the transport coefficients. It will also allow us to reduce the number of uncertain inputs that have negligible effect on the output variance [16, 25].

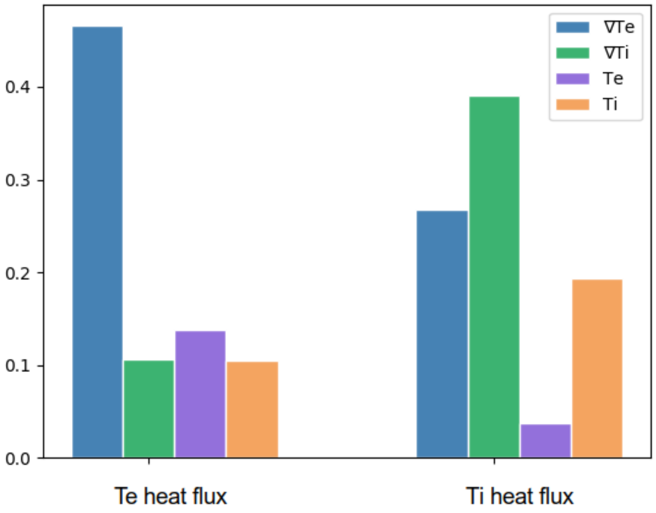

Fig. 3. First-order Sobol indices electron and ion heat fluxes.

DOI: 10.1007/978-3-030-50436-6_53

As indicated in Fig. 3, the variance in the electrons and ion heat fluxes are mainly due to the uncertainty in temperature gradients. However, we should be cautious about the impact coming from temperature uncertainties, especially Ti on ion heat flux.

In this case, we also use PCE method with spectral projection and with regression. As in the previous case, we also set the polynomial order to 4 and have 4 uncertain parameters, hence, the number of samples requrired are 1296 and 140, respectively. The parallel execution of the samples, whether the black box contains serial or parallel codes, is done via QCG PilotJob [5]. This middleware considers each sample as an independent job, but schedule them dynamically through a single batch allocation. For example, using 32 cores for each sample, we notice that the time to run UQ simulation with regression is 10 times less than the spectral projection.

5 Conclusion

In this paper, we presented results from the introduction of uncertainty quantification and sensitivity analysis methods in our multiscale fusion workflow application. We focused so far on two black box use cases: the entire coupled workflow with simple analytical model for turbulence, and singled-out 3D gyrofluid turbulence model. Integration of the non-intrusive methods was done with the the EasyVVUQ library and the parallel execution of samples was performed in a single batch allocation with QCG-PilotJob. Both tools, available as part of the VECMA toolkit and in conjunction with the coupled workflow nature of our application, have proven to be simple to use and to allow a large variety of experiments with different methods and models.

A next step will be to use quantified uncertainties in order to improve the validation of our simulation results against experimental measurements. We also want to explore semi-intrusive methods, by coupling black boxes with propagation of uncertainties or by replacing the expensive 3D model with a cheap surrogate, in order to reduce the computational cost for quantifying the uncertainties such as it can be performed routinely in our simulation. Other optimizations (e.g parallelization of the sampling step with a closer EasyVVUQ–QCG-PilotJob integration) and scalability tests are being performed and will be the subject of a more in-depth performance study.

Acknowledgements

This work was supported by the VECMA project, which has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 800925. We also acknowledge EUROfusion for the computing grant on MARCONI-Fusion supercomputer which allowed us to run all simulations presented in this paper.

References

1. Falchetto, G.L., Coster, D., Coelho, R., Scott, B., Figini, L., Kalupin, D., Nardon, E., Nowak, S., Alves, L.L., Artaud, J.F., et al.: The European Integrated Tokamak Modelling (itm) effort: achievements and first physics results. Nuclear Fusion **54**(4) (2014) 043018
2. Luk, O., Hoenen, O., Bottino, A., Scott, B., Coster, D.: Compat framework for multiscale simulations applied to fusion plasmas. Computer Physics Communications (2019)
3. Groen, D., Richardson, R.A., Wright, D.W., Jancauskas, V., Sinclair, R., Karlshoefer, P., Vassaux, M., Arabnejad, H., Piontek, T., Kopta, P., Bosak, B., Lakhlili, J., Hoenen, O., Suleimenova, D., Edeling, W., Crommelin, D., Nikishova, A., Coveney, P.V.: Introducing VECMAtk - Verification Validation and Uncertainty Quantification for Multiscale and HPC Simulations. In: Computational Science – ICCS 2019. Springer International Publishing (2019) 479–492
4. Richardson, R.A., Wright, D.W., Jancauskas, V., Lakhlili, J., Edeling, W.: Easyvvuq: A library for verification, validation and uncertainty quantification in high performance computing (2019) Submitted to Journal of Open Research Software. Under review.
5. Piontek, T., Bosak, B., Ciżnicki, M., Grabowski, P., Kopta, P., Kulczewski, M., Szejnfeld, D., Kurowski, K.: Development of science gateways using qcg — lessons learned from the deployment on large scale distributed and hpc infrastructures. Journal of Grid Computing **14**(4) (Dec 2016) 559–573
6. Schmidt, P.C.: C. w. gardiner: Handbook of stochastic methods for physics, chemistry and the natural sciences, springer-verlag, berlin, heidelberg, new york, tokyo 1983. 442 seiten, preis: Dm 115,-. Berichte der Bunsengesellschaft für physikalische Chemie **89**(6) (1985) 721–721
7. Xiu, D.: Numerical Methods for Stochastic Computations: A Spectral Method Approach. Princeton University Press, USA (2010)
8. Sobol, I.: On quasi-monte carlo integrations. Mathematics and Computers in Simulation **47**(2) (1998) 103 – 112
9. Saltelli, A., Annoni, P. In: Sensitivity Analysis. Springer Berlin Heidelberg, Berlin, Heidelberg (2011) 1298–1301
10. Saltelli, A.: Global sensitivity analysis: the primer. John Wiley (2008)
11. Lemieux, C.: Monte Carlo and Quasi-Monte Carlo Sampling. Springer Series in Statistics. Springer, Dordrecht (2009)
12. Sullivan, T.J.: Introduction to Uncertainty Quantification. Volume 63. Springer (2015)
13. Hammersley, J.M.: Monte carlo methods for solving multivariable problems. Annals of the New York Academy of Sciences **86**(3) (1960) 844–874
14. Saltelli, A.: Making best use of model evaluations to compute sensitivity indices. Computer Physics Communications **145**(2) (2002) 280 – 297
15. Xiu, D., Karniadakis, G.E.: The wiener–askey polynomial chaos for stochastic differential equations. SIAM Journal on Scientific Computing **24**(2) (2002) 619–644
16. Eck, V.G., Donders, W.P., Sturdy, J., Feinberg, J., Delhaas, T., Hellevik, L.R., Huberts, W.: A guide to uncertainty quantification and sensitivity analysis for cardiovascular applications. International Journal for Numerical Methods in Biomedical Engineering **32**(8) (2016) e02755 e02755 cnm.2755.

17. Preuss, R., von Toussaint, U.: Uncertainty quantification in ion–solid interaction simulations. Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms **393** (2017) 26 – 28 Computer Simulation of Radiation effects in Solids Proceedings of the 13 COSIRES Loughborough, UK, June 19-24 2016.
18. Sudret, B.: Global sensitivity analysis using polynomial chaos expansions. Reliability Engineering and System Safety **93**(7) (2008) 964 – 979 Bayesian Networks in Dependability.
19. Feinberg, J., Langtangen, H.P.: Chaospy: An open source tool for designing methods of uncertainty quantification. Journal of Computational Science **11** (2015) 46–57
20. Hosder, S., Walters, R., Balch, M.: Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. In: 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. (04 2007)
21. Herman, J., Usher, W.: SALib: An open-source python library for sensitivity analysis. The Journal of Open Source Software **2**(9) (jan 2017)
22. Coster, D.P., Basiuk, V., Pereverzev, G., Kalupin, D., Zagórksi, R., Stankiewicz, R., Huynh, P., Imbeaux, F., Members of the Task Force on Integrated Tokamak Modelling: The european transport solver. IEEE Transactions on Plasma Science **38**(9) (Sep. 2010) 2085–2092
23. Nikishova, A., Veen, L., Zun, P., Hoekstra, A.G.: Semi-intrusive multiscale metamodeling uncertainty quantification with application to a model of in-stent restenosis. Philosophical Transactions A (2018)
24. Scott, B.D.: Free-energy conservation in local gyrofluid models. Physics of Plasmas **12**(10) (2005) 102307
25. Nikishova, A., Veen, L., Zun, P., Hoekstra, A.G.: Uncertainty quantification of a multiscale model for in-stent restenosis. Cardiovascular Engineering and Technology **9**(4) (Dec 2018) 761–774