



D4.2: Report on the implementation of non-intrusive VVUQ techniques

Due Date	14 December 2019
Delivery	05 December 2019
Submission of updated version	N/A
Lead Partner	MPG
Dissemination Level	Public
Status	Final
Approved	Executive Board
Version	V1.0



DOCUMENT INFO

Date and version number	Author	Comments
30.09.2019 v0.1	Onnie Luk	Skeleton and additional texts
18.10.2019 v0.2	All	Report from all applications
28.10.2019 v0.3	Onnie Luk	Reorganize report with additional texts
30.10.2019 v0.3	Wouter Edeling	Review of v0.3
08.11.2019 v0.4	All	Reorganize report with additional texts
26.11.2019 v0.5	Onnie Luk, Olivier Hoenen	Make further modifications to address internal reviewers' comments
04.12.2019 v1.0	Onnie Luk	Finalized report

CONTRIBUTORS

- Onnie Luk (MPG) – Main Author
- Olivier Hoenen (MPG) – Editor
- Jalal Lakhilili (MPG) – Author
- Wouter Edeling (CWI) – Author
- Anna Nikishova (UvA) – Author
- Dongwei Ye (UvA) – Author
- Diana Suleimenova (Brunel) – Author
- Derek Groen (Brunel) – Author
- Robin Richardson (UCL) – Author
- David Wright (UCL) – Author
- Robert Sinclair (UCL) – Author
- Michal Kulczewski (PSNC) - Author

DISCLAIMER

This document's contents are not intended to replace consultation of any applicable legal sources or the necessary advice of a legal expert, where appropriate. All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user, therefore, uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors' view.

TABLE OF CONTENTS

1 Executive summary4

2 Implementation of non-intrusive VVUQ techniques4

 2.1 UQ methods and implementation into applications5

 2.1.1 Polynomial Chaos.....5

 2.1.2 Quasi-Monte Carlo8

 2.1.3 Stochastic Collocation10

 2.1.4 Parameter sweeps and Ensemble bootstrapping14

 2.2 Lessons learned and ongoing work16

 2.2.1 Continuum blood flow17

 2.2.2 Urban air pollution17

3 Conclusions18

4 References18

LIST OF FIGURES

Figure 1: Comparison between experimental measurements and simulation results 7

Figure 2: Left: Uncertainty estimation results: (i) mean value, aleatory and total uncertainty 10

Figure 3: An illustration of input parameter samples for Mali 13

Figure 4: Graph of the data shown in the above table. The measured Young’s modulus 15

LIST OF TABLES

Table 1: List of uncertain inputs from the ISR2D model..... 9

Table 2: Sobol indices for the two-dimensional ocean model..... 12

Table 3: FabFlee and EasyVVUQ input parameter exploration results 14

Table 4: Raw output from an EasyVVUQ bootstrap analysis, measuring the Young’s modulus 15

1 Executive summary

This deliverable reports on achievements from Task 4.2, which focused on the fast-track implementation of validation, verification and uncertainty quantification (VVUQ) techniques into applications from Work Package 4 (WP4). Here the fast-track label relates to the use of non-intrusive techniques only, which require less effort from the application developers as the models and their implementations remain effectively unchanged. While the underlying sampling and analysis methods are well known in the uncertainty quantification (UQ) community (Polynomial Chaos, Monte Carlo, Stochastic Collocation, Sobol sensitivity analysis) and were implemented as is in the VECMA toolkit (VECMAtk), a large part of the task for WP4 was to provide inputs to the technical work packages WP2, WP3 and WP5 and feedback on their produced algorithms, tools and middleware. Such inputs were reported already through previous deliverables (D4.1, D2.1, D3.1, D3.2) and helped shape the VECMAtk tutorials (<https://www.vecma-toolkit.eu/tutorials/>) while feedback happened mostly through issues on the GitHub pages of the various tools being used.

In this report we show that, from a portfolio of 8 applications, 6 have demonstrated results with the range of UQ methods implemented within the VECMAtk (only 4 were initially labelled as fast-track candidates). From these, 2 applications (ISR and BAC) already had implementations of UQ techniques before the start of the project, and have been extensively driving the definitions of uncertainty quantification patterns (UQP) by WP2 as well as the implementation of the toolkit by WP3. At the time of writing, 5 applications (Fusion, Climate, Migration, Materials and BAC) integrate parts of the VECMAtk to quantify the uncertainties of the entire multiscale application or to perform sensitivity analysis (SA) on one or several of their single-scale models. In addition, plans have been laid out to perform non-intrusive UQ on single-scale models by using VECMAtk in the Continuum Blood Flow application, as well as in the Urban Air pollution application (external to VECMA), indicating that the toolkit is already getting some uptake from other communities and application groups.

While most of the results presented in this document concern the SA and quantification of uncertainty, a couple of early results in verification and validation (VV) are mentioned as well as an action plan for speeding-up this effort in the coming period.

2 Implementation of non-intrusive VVUQ techniques

In VECMA, WP4 is composed of 8 multiscale applications which come from various scientific and societal domains and have been described previously in the D4.1 deliverable. The main purpose of these applications is to steer the developments from the technical work packages (WP2 — methods and algorithms, WP3 — software implementation, WP5 — infrastructure) and serve as demonstrators showing that such developments are useful in real applications.

In order to deliver initial developments as fast as possible, the work was structured into fast-track and deep-track. This report focuses on the fast-track implementation of VVUQ techniques within WP4 applications and the early results.

When referring to the fast-track we are considering only non-intrusive VVUQ techniques, which mean that the code of the application remains unchanged. For UQ this lies under the UQP1 label (cf. D2.1 deliverable for all details on UQP) for which the method is applied to a black-box (either a function, an entire multiscale application or just one submodel). For the use cases where UQP1 is applied to a single-scale submodel (typically to perform SA on a given component of a larger application), it does not correspond to a multiscale simulation anymore. Nevertheless it demonstrates the flexibility and genericity of the technique and provides valuable knowledge to the application developers with respect to the integration of UQ within full multiscale application in the future.

This section is divided into two parts: section 2.1 describes briefly the 4 targeted UQ methods implemented in the EasyVVUQ as instances of the Sampler (cf. D3.1 for details on the EasyVVUQ architecture and the rest of the VECMAtk developed through WP3). For each method, practical examples are given by describing its implementation within one or more applications from the WP4 portfolio. Note that the various analysis methods (e.g Sobol indices) being used in these examples are not described here as the sampling remains the main focus for applications in order to determine the computational costs of the method. Then, in section 2.2 we discuss some of the lessons learned throughout Task 4.2 as well as non-intrusive work that is still ongoing.

2.1 UQ methods and implementation into applications

The non-intrusive UQ methods employed by the WP4 applications so far are the Polynomial Chaos (PC), quasi-Monte Carlo (qMC), Stochastic Collocation (SC), and parameter sweeps followed by a bootstrap analysis on the resulting ensemble. The EasyVVUQ library within the VECMAtk [1] has UQ elements that guide users through the VVUQ process, from defining the uncertain input parameters, to analyzing the probability distributions in the quantities of interest (QoI). The library has all 4 UQ methods as Sampler and Analysis options.

The following sections introduce the UQ methods one by one, with detailed example(s) from WP4 applications on how to implement the method and a breakdown of the analysis.

2.1.1 Polynomial Chaos

The PC method approximates a model Y by a polynomial expansion, where the polynomials are orthogonal to the probability density function of the inputs:

$$Y \approx \hat{Y}(x, t, Q) = \sum_{n=0}^{N_p-1} c_n(x, t) P_n(Q).$$

P_n , c_n , and N_p are the polynomials, expansion coefficients, and number of expansion factors, respectively [2].

The advantage of PC is that it requires fewer samples than qMC and the Monte Carlo (MC) methods. However, that advantage only holds true when the number of uncertain parameters is below 20. [3,4,5] It is important to note that the PC expansion can only be estimated with a finite number of terms; there is always truncation error in the estimates of uncertainty. In particular, the variance, standard deviation and Sobol sensitivity indices tend to be underestimated since, in their computation, the summation of the squares of the coefficients is involved.

2.1.1.1 Fusion application

The multiscale fusion workflow (MFW) was created to study the effects of microscale turbulence on the overall plasma transport in a tokamak (an experimental device that confines plasma at very high temperatures with magnetic fields). Currently, the workflow combines three single-scale models that simulate the plasma geometry (equilibrium model), turbulence, and overall transport.

Uncertainties in this application come from applied heating sources and boundary conditions (epistemic) and from the stochastic behavior of the turbulence (aleatoric). Since fast-track considers the entire workflow as a black-box, we only include the epistemic uncertainties: boundary conditions defined by the electron and ion temperatures (T_e and T_i , respectively) at the plasma edge, and characteristics of the applied heating sources (e.g. amplitude, width, and position for both ion and electron heating power distributions).

We use the PC method to perform UQ studies in a non-intrusive manner by following UQP1 on the entire application. Six samples along a single uncertain dimension are considered and then expanded to a multi-dimensional PC grid via a tensor product construction. For example, if all 8 uncertain parameters are considered, then 6^8 , or about 1.7 million samples are generated. This is true when full tensor is utilized to fill the sampling space. However, the number of PC samples grows exponentially with the number of uncertain parameters, and processing all of these samples can become cumbersome. Therefore, qMC [6] or PC with a sparse grid [7] become better options in generating samples.

Once the uncertain input parameters are defined, the MFW runs within a black-box for multiple time iterations until the plasma reaches steady state. The QoI from this simulation are the output T_e and T_i coming from the macroscopic transport model. The EasyVVUQ library carries out the UQ analysis and provides the mean, standard deviation, and variance to the QoI. Sobol SA, which is also implemented

in the EasyVVUQ, determines how sensitive the QoI are to uncertainties introduced by the input parameters. For this study, the 1st order and total Sobolj sensitivity indices are calculated for each uncertain input parameter. The former quantifies the direct effect on the output variance from individual input parameter. The latter evaluates the sensitivity due to interactions between a given parameter and all combinations of the other parameters [8].

We take the MFW to simulate an ASDEX Upgrade¹ (AUG) tokamak scenario. We use EasyVVUQ to generate samples under the PC method, encode samples, execute simulations, decode simulation results, and analyze UQ outputs. The outcome from Analysis is plotted in Figure 1, in which we show the expected value and standard deviation to the QoI: T_e (left) and T_i (right) with respect to the normalized toroidal flux coordinate “rho_tor_norm”.

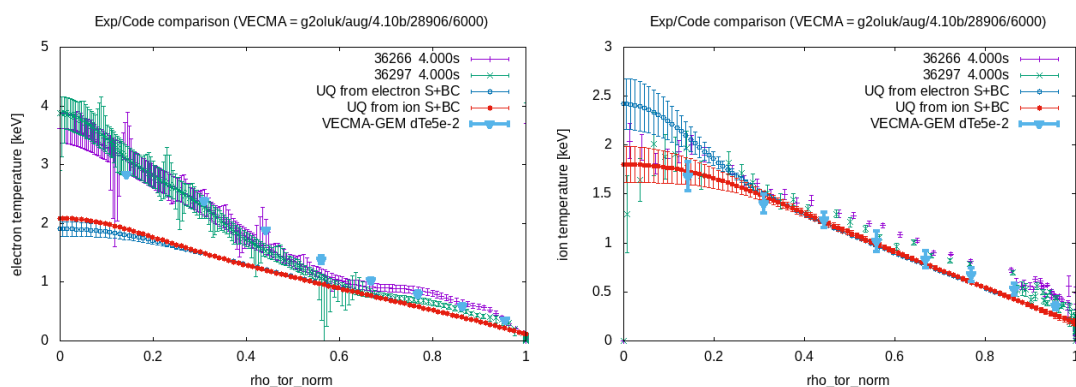


Figure 1: Comparison between experimental measurements and simulation results of the electron (left) and ion (right) temperature profiles in the AUG tokamak. The measured temperatures with error bars are labeled by their 5-digit shot numbers. The simulated temperature profiles with the electron (ion) boundary condition and the characteristics of the electron (ion) heating source as the input uncertain parameters are shown in thin blue (red) line. The simulation results using *GEM* are averaged over 200 time steps at the quasi-steady state. The error bars of all simulation results represent the standard deviation.

To validate the UQ results, measurements coming from 2 experimental shots are plotted in the figure as well. While T_e profiles obtained from the UQ studies do not agree at the inner region (with lower rho_tor_norm values), the outer region have more agreeable outcome.

As part of the model verification effort, different turbulence models are implemented within the MFW. We run the MFW without any UQ using a turbulence model named *GEM*, which is a gyrofluid electromagnetic flux-tube model that determines heat and particle fluxes at each flux-tube in a field-aligned shifted-metric coordinate system [9]. For each of these simulation runs, the MFW uses adaptive time steps and runs until the plasma reaches quasi-steady state [10]. The output temperatures of such a production-sized run are then averaged over and are plotted in the figure

¹ <https://www.ipp.mpg.de/16195/asdex>

above. On the other hand, the UQ results discussed earlier used a simplified local flux-tube model to simulate the turbulence.

Overall, the simulation results show that the turbulence model *GEM* has better agreement with experiments than the simplified flux-tube model. This is especially obvious in the T_e profiles. As for the T_i profiles, the simulations agree fairly well with the experiment.

2.1.2 Quasi-Monte Carlo

In order to study uncertainty of the model response, the MC method samples randomly the values of uncertain inputs from their probability distribution function and propagates them through the model. The number of samples N_s tends to be a large number in the MC sampling method. On the other hand, qMC can lower N_s with variance-reduction techniques such as the Sobol sequence that allows more uniform sampling to avoid clustering of the samples [11]. The simulation model runs through the entire set of sampling parameters to calculate statistical inferences to the QoI $Y = [Y_1, Y_2, \dots, Y_{N_s}]$. For example, the expected value of Y is [12]:

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

Importantly, the convergence rate of qMC method is of the order of N^{-1} instead of $N^{-1/2}$ from the MC method. Convergence of some quasi-random sequence may depend on the dimension of the uncertain input [6]. However, research shows that in practice, for instance, qMC with the Sobol sequence outperforms the purely random MC method, e.g. [13].

2.1.2.1 In-stent restenosis (2D) application

The ISR2D model is a two-dimensional representation of the in-stent restenosis process. Its domain is a longitudinal section of the artery, in which five subprocesses take place: stent deployment, post-deployment smooth muscle cell proliferation into the lumen, blood flow through the lumen, reendothelialization, and diffusion of antiproliferative drugs from the stent into the tissue. As these processes take place at different temporal scales, the ISR2D model is built into multiscale model [14]. There are three submodels: an initial condition model which offers the initial geometry after the strut is deployed, an agent-based model of Smooth Muscle Cell (SMC) growth into the artery lumen, and a simulation of blood flow through the artery. SMC growth changes the geometry of the artery, which affects the blood flow, while the wall shear stress (WSS) caused by the blood flow inhibits SMC growth when it is above a threshold value. The final output of the model is the cross-sectional area of neointima.

The model is subject to aleatoric and epistemic uncertainty. Aleatory uncertainty is due to natural variability of the cell cycle, the relative orientation of daughter cells, and the pattern of

reendothelialization in the SMC model. Epistemic uncertainty is due to imperfect measurements of the values of flow velocity, maximum deployment depth and endothelium regeneration time. The list of uncertain inputs is shown in Table 1, where a uniform distribution is assumed between the minimum and maximum values. Instead of uniform random numbers, we used the Sobol sequence. The Sobol sequence is low-discrepancy, which means it covers input space X more evenly. To estimate the uncertainty in model response, we applied a black-box qMC method [15]. The simulation was developed and tested on the Distributed ASCI Supercomputer (DAS-5) using Python scripts. Similar work has also been carried out using the semi-intrusive metamodeling method [16].

Parameter	Unit	Min value	Max value
Flow velocity	m s^{-1}	0.432	0.528
Maximum deployment depth	mm	0.09	0.13
Endothelium regeneration time	days	15	23

Table 1: List of uncertain inputs from the ISR2D model.

From the UQ and SA (Figure 3) we find that the largest contribution to the uncertainty comes from the endothelium regeneration speed, followed by the deployment depth. This is similar to results obtained in earlier studies for a similar 3D model [17]. There it was also demonstrated that a difference in reendothelialization speed changes growth to a large extent, while the effects of the deployment depth are smaller, but still quite pronounced. As the re-endothelialization has such significant impact on the neointimal area, this part of the ISR model requires further study and validation. We are currently undertaking more detailed modelling, based on controlled experiments of endothelial cell migration on substrates. The results once more confirm our hypothesis that the development of restenosis is very much driven by the inability to quickly regenerate a functional endothelium. However, these results cannot be considered final. The model parameters were based on a previous publication [18] which is considered a largely simplified model geometry, and some other parameters also were assumed without a thorough investigation on their distribution or variability range. For instance, the effect of the flow velocity requires a deeper study; we are looking in detail at the expected physiological variability (instead of the 10% variability that we now assumed).

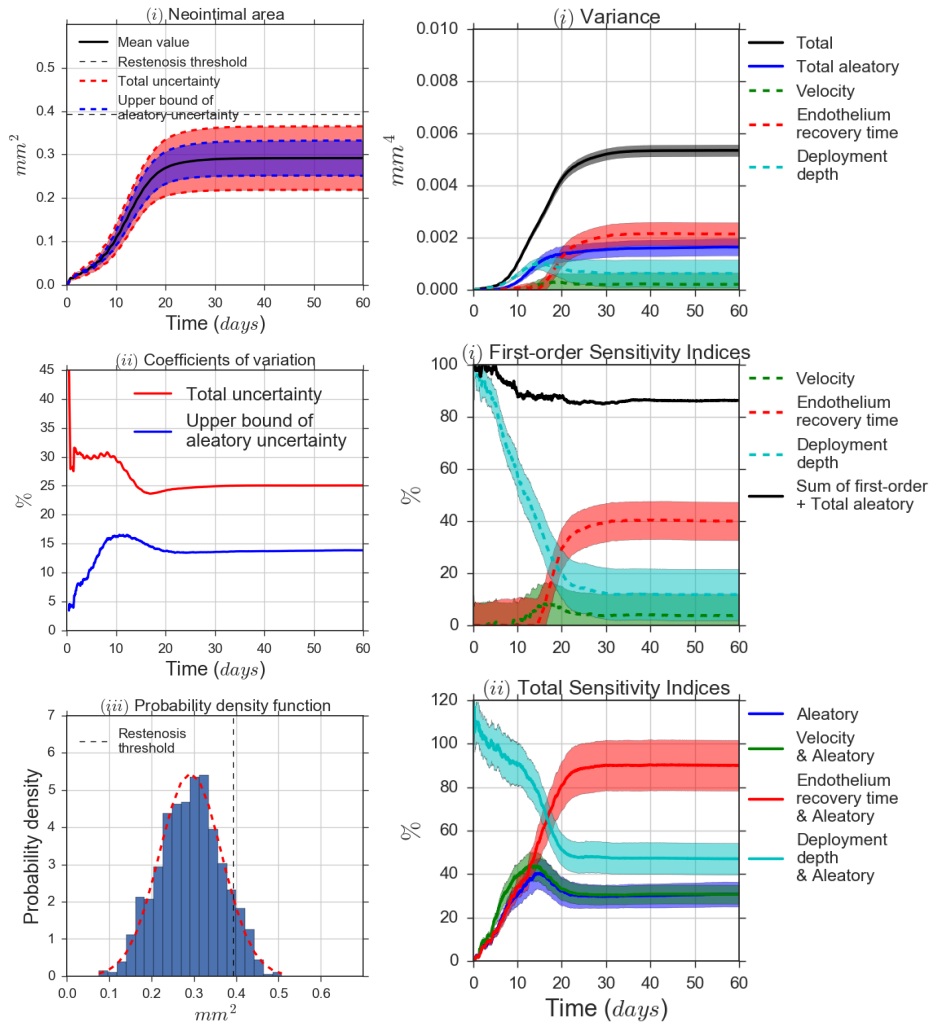


Figure 2: Left: Uncertainty estimation results: (i) mean value, aleatory and total uncertainty of the neointimal area as a function of time; (ii) relative total uncertainty (red line) and relative aleatory uncertainty (blue line) as a function of time; (iii) Probability Density Function (PDF) of the neointimal area at the final time step, where red dashed curve is a fit of a normal distribution. Right: Sensitivity Analysis with the Bootstrap test results: (i) partial and total variances; (ii) first-order sensitivity indices of uncertain inputs, where solid black line denotes the sum of the first order effect of uncertain inputs plus the total effect of the model stochastic variables; (iii) total sensitivity indices of uncertain parameters together with the stochastic variables. The shadow areas are one standard deviation of the estimators.

2.1.3 Stochastic Collocation

The SC method creates polynomial approximation of a quantity of interest q in the (multi-dimensional) stochastic space via the following expansion, written concisely as:

$$q(x, t, \xi) \approx \sum_{j=1}^{N_p} q_j(x, t) L(\xi)$$

Here, q_j are the code samples computed on a structured multi-dimensional grid, and N_p is the total number of collocation points. The samples q_j are interpolated to an arbitrary point within the stochastic space by means of Lagrange interpolation polynomials $L(\xi)$. For interpolation in multiple dimensions, $L(\xi)$ is built as a tensor product of one-dimensional Lagrange polynomials. The SC method, as well as the PC method, are well-known and we refer to [19] for more details on these techniques.

2.1.3.1 Climate application

Computational models for atmospheric and oceanic flows are central to climate modeling and weather forecasting. As these models have finite resolutions, they employ simplified representations, so-called parameterizations, to account for the impact of unresolved processes on larger scales. An example is the treatment of atmospheric convection and cloud processes, which are important for the atmospheric circulation and hydrological cycle, but are unresolved in global atmospheric models. These parameterizations are a source of uncertainties: they have parameters that can be difficult to determine, and even their structural form can be uncertain.

Although our main focus within VECMA is the development of a toolkit of surrogate models (deep-track) for the unresolved processes of aforementioned geophysical flow problems, we also performed some implementation of fast-track UQ methods. Specifically, we implemented the SC method (including sparse-grid capability and Sobol index computation) into EasyVVUQ. To run the EasyVVUQ samples on HPC resources, we created an interface between EasyVVUQ and FabSim3 together with the developers from WP3, see [20]. FabSim3 is a Python3-based, command-line automation toolkit for scientific computing on large-scale and distributed e-infrastructures [21]. It facilitates amongst others the execution of remote (ensemble) jobs, and automates the transfer of data to and from those resources, all from the local command line. Once the interface is set up, only some small changes to the code are required. This differs to an EasyVVUQ Campaign, which runs on the localhost, see [20] for a tutorial.

This fast-track implementation will allow us to perform non-intrusive UQ on the level of resolved scales, after that the subgrid-scale term will be replaced by a surrogate. Typical uncertain parameters of the resolved scale equations could for instance be the value of viscosity or forcing term coefficients in the ocean models, which are normally manually tuned by the user. Instead, we can use EasyVVUQ to prescribe input distributions, and compute the output uncertainty or the Sobol sensitivity indices via the SC Sampler.

To test our method, we prescribed uniform distributions for two parameters (i.e. the viscosity ν and forcing term coefficient μ) of a two-dimensional ocean model. Our output QoI were the time-averaged total energy and enstrophy of the system, as a function of the uncertain coefficients,

computed by running each sample for a simulated time of 8 years. We used the developed EasyVVUQ-FabSim3 interface [20] in order to run the corresponding EasyVVUQ samples via FabSim3 on the PSNC Eagle supercomputer. As an example, below we show all Sobol sensitivity indices for both uncertain parameters, from which we can infer that the viscosity ν is the most influential model parameter (a value close to 1 means that parameter explains most of the variance in the outputs).

Parameter	Sobol indices of the energy
viscosity ν	0.8945
forcing term coefficient μ	0.1022
interaction ν and μ	0.0033

Parameter	Sobol indices of the enstrophy
viscosity ν	0.7494
forcing term coefficient μ	0.2393
interaction ν and μ	0.0113

Table 2: Sobol indices for the two-dimensional ocean model.

In addition to the surrogate modelling effort, some follow-up steps along the described non-intrusive direction are also planned. We will consider a more expensive model, namely a three-dimensional large eddy simulation of the atmosphere, and study the effect of propagating the uncertainty of several model parameters, whose influence is rarely investigated.

2.1.3.2 Multiscale migration prediction application

Forced migration reached record levels in 2019 with 25.9 million refugees spread worldwide [22]. Predicting forced migration is crucial, as governments and NGOs can save refugees' lives and allocate humanitarian resources efficiently [23]. We use an agent-based modelling tool, namely FLEE (<https://github.com/djgroen/flee-release>), to forecast the distribution of incoming refugees across destination camps for African countries [24]. To automate the construction, execution and analyses of ensemble simulations of refugee movements [25], we have established a FabSim3-based FabFlee (<https://github.com/djgroen/fabflee>) plugin. It provides an environment for researchers and organisations to construct and modify refugee simulations, instantiate and execute multiple runs for different policy decisions, as well as to validate and visualise the obtained results against existing data [26].

Within VECMA, we focus on the input data uncertainty, SA of simulation runs, validation of obtained simulation results and verification of our initial model. Currently, we perform non-intrusive UQ

procedures with SC and Sobol SA on input parameters of multiscale migration models. To make this possible, we have integrated the FabFlee plugin with EasyVVUQ using a separate set of automation definitions that can be linked or unlinked by developers. In it, we re-use the SC approach used for the climate application in the context of migration modelling.

We perform SA on location move chances (such as camp, conflict and default locations), which vary between 0.0001 to 1.0, and speed limits of refugee movements to understand in what regime and scenario our simulation approach perform well.

With FabFlee and EasyVVUQ we perform SA on input parameters, and rely on QCG Pilot Jobs (<https://github.com/vecma-project/QCG-PilotJob>) to facilitate the execution of large ensemble runs. We also plan to explore agent awareness levels and other simulation parameters, as well as basic one-at-a-time input UQ for multiscale migration application. In addition, we are in the process of exercising these analyses for a multiscale macro-micro model of the South Sudan situation.

We run the Mali conflict scenario using a combined function of FabFlee and EasyVVUQ for three parameters (see Figure 4). We apply the SC method [27] and obtain the results illustrated in Table 3. The camp_move_chance parameter is more sensitive in our model compared to the other two parameters since camps are the destination locations for refugees fleeing from conflict locations. We also find that our models are not sensitive to the combination of these parameters.

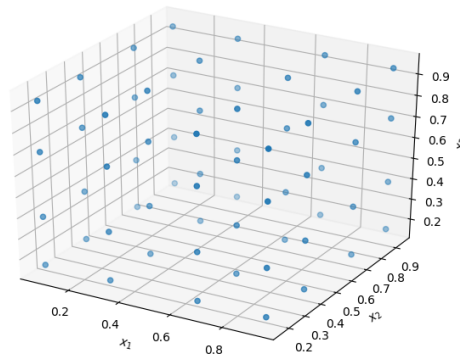


Figure 3: An illustration of input parameter samples for Mali, where x_1 is camp_move_chance, x_2 is conflict_move_chance and x_3 is default_move_chance.

Parameters	Sobol indices
camp_move_chance (0)	0.88814118
conflict_move_chance (1)	0.05990016
default_move_chance (2)	0.04551298
0, 1	0.00030681
0, 2	0.00357396
1, 2	0.00243685

0, 1, 2	0.00012808
---------	------------

Table 3: FabFlee and EasyVVUQ input parameter exploration results

Lastly, as part of the migration tutorial for the M12 release (see <https://github.com/digroen/FabFlee/blob/master/doc/Tutorial.md>) and in collaboration with the HiDALGO CoE project (<http://hidalgo-project.eu>) we have introduced a very basic form of UQ that doesn't quite fit in our existing categories. We require conflict event forecasts if we wish to predict forced population displacements, but creating an accurate or predictive conflict simulation give rise to major ethical problems. So, instead, we developed a *conflict generator* that creates a range of realistic (but not necessarily correct) conflict evolution scenarios. We then use a large sample of these scenarios to give an estimate of the uncertainty in camp arrivals in the migration model, based on variations induced by the different instances of the (highly stochastic) conflict model.

2.1.4 Parameter sweeps and Ensemble bootstrapping

Along with more advanced UQ techniques (such as PC and SC) the VECMAtk also provides functionality for basic parameter sweeps (generating samples by cycling through a user defined range of values on one or more variables) using the "SweepSampler" element from the EasyVVUQ library. This allows also for an ensemble of "replica" runs to be generated for each group of input parameters, on which a bootstrap analysis can then be performed, using the EnsembleBoot analysis element (also from EasyVVUQ).

2.1.4.1 Materials application

In the full Materials application we have a cyclically coupled heterogeneous multiscale model with two scales - the (continuum, finite element) macroscale, and the (discrete, molecular dynamics) nanoscale [28,29]. The local strain (deformation) at the macroscale is calculated using the stress response. This stress response is calculated at the nanoscale using an ensemble of molecular dynamics simulations, in order to better approximate a continuum response. UQ for the fully coupled Materials model is a deep-track activity, so in this deliverable our work concentrates on non-intrusive approaches to estimate the uncertainty obtained from nanoscale only.

The uncertainty in this nanoscale stress response is both epistemic - in terms of the molecular dynamics forcefield parameters - and aleatoric - in terms of the natural variation between one (finite) MD box of molecular structure and another. We are initially interested in testing the sensitivity of our overall stress calculation to both of these sources of error. The preliminary sensitivity test is a simple parameter sweep across multiple sizes of the MD box, multiple different times at which the

straining begins, and multiple different polymer networks. In total this results in around 1500 different runs.

We use the EasyVVUQ library to implement this workflow, in particular the “SweepSampler” VVUQ element. As there are approximately 1500 runs to execute (taking 96 cores each, and over a week to complete), we use the restartability functionality of EasyVVUQ to separate the sampling and analysis stages of the workflow to be implemented in two separate scripts.

	YM		
	boot	high	low
box_size			
30	42537.398920	12157.451562	73692.602069
40	34968.506380	16546.698645	51600.928269
60	34841.820061	23016.567640	45162.132650
80	34631.485156	28096.543163	41497.338792
100	34175.572852	28879.032293	38808.258953

Table 4: Raw output from an EasyVVUQ bootstrap analysis, measuring the Young’s modulus of an ensemble different sized simulations of epoxy polymers.

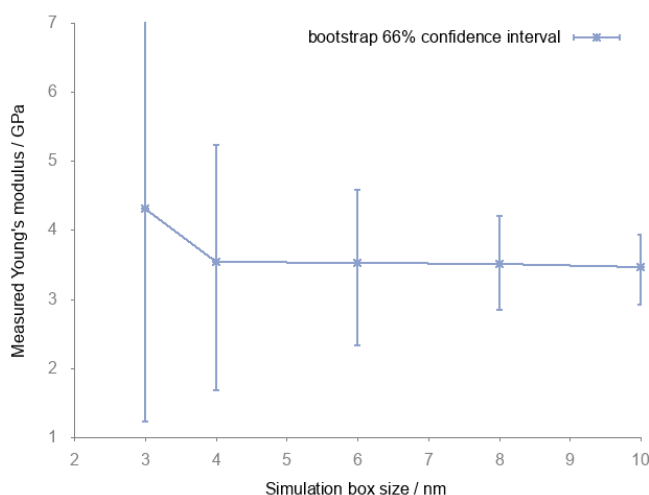


Figure 4: Graph of the data shown in the above table. The measured Young’s modulus is independent of system size for systems with box length over 4 nm.

We find that the measured Young’s modulus is only affected on simulation sizes below 4 nm. This indicates that on measuring the stress due to small deformations (0.5% strain), MD simulations must be above a minimum size. We expect this size to be related to the characteristic length scale of the

polymer network. We also see, from the above table and graph, that the uncertainty in the measured quantity is greatly affected by simulation size.

2.1.4.2 Binding affinity calculator application

The Binding Affinity Calculator (BAC) application uses large ensembles of molecular dynamics simulations to predict the binding affinities of ligands (such as drug molecules) with proteins [30]. It was already in production use at the start of the VECMA project (including UQ) and has been used to drive development of features in EasyVVUQ. All the code for ensemble statistics calculation has now been captured in EasyVVUQ. The workflow essentially performs a parameter sweep over the input ligand (with ensembles of simulations based on the same structure but with randomized initial velocities). Ideal bootstrap analysis is then performed on the results, to obtain the binding affinities with accompanying confidence intervals. This approach has been used in the analysis of papers using both computationally cheap end point approaches (known as ESMACS protocols) [31, 32] and more rigorous alchemical binding free energy calculations (known as TIES protocols) [33].

The BAC workflow was used to develop the bootstrap analysis and SweepSampler elements in EasyVVUQ, which are now used within the application. BAC is used within the INSPIRE project to compute binding free energies of inhibitors binding to different kinase sequences. In this context the RADICAL Cybertools pilot job manager [34] was used as the middleware to deploy simulations on the world's largest supercomputer, Summit at the Oak Ridge Leadership Computing Facility in the US.

2.2 Lessons learned and ongoing work

Although the first public release of VECMAtk was less than 6 months ago (at M12), it has already attracted interest from external application groups, within and outside the scientific domains covered in the WP4 portfolio. As this release implements mostly well-known UQ methods that are already available through other software suites, such early uptake shows that there is room for solutions like EasyVVUQ, which focuses on simplicity and modularity for providing an end-to-end solution using a high-level framework in Python. We expect that the toolkit will gain even more scrutiny during the second phase of the project when advanced methods for semi-intrusive UQ and meta-modelling become available.

Throughout the first half of the project, the focus has been more towards UQ rather than VV (for which only a few preliminary experiments have been performed), which may be more closely related to the specificities of the application and its domain. This makes VV harder to design through generic solutions. In addition, the delay in VV implementation can be explained by the fact that it depends heavily on the effective quantification of uncertainties in the target application. Now that most applications have UQ results, in order to bring VV up-to-speed within the project, a dedicated

development team comprised of a small number of WP3 and WP4 members has been created. It aims at developing quick prototypes for measuring the distance (e.g Kullback-Leibler entropy and Hellinger distance) between distributions obtained through UQ.

In addition, two more applications are still working towards the integration of non-intrusive methods for studying UQ from one of their models, as described below.

2.2.1 Continuum blood flow

The continuum blood flow application (deep-track) at UCL concerns the coupling of haemodynamic simulations of the arterial and venous trees (obtained through MRI scans), handled with a lattice-Boltzmann solver (HemeLB [35]), and coupling with a full electro-physiological model of the human heart, handled with the Alya solver. This is to allow study of transport of particles (cells or drugs) through the body's bloodstream.

This is a challenging case due to the very high computational expense of such a coupled model. As such, the initial UQ related approaches are likely to involve basic SA over given heart parameters, such as heart rate, or rigidity of the vessel walls (this functionality is currently being added to HemeLB). We intend to use EasyVVUQ for UQ matters initially, although surrogate models may be employed at a later date due to the high computational costs involved.

2.2.2 Urban air pollution

The UrbanAir application at the Poznań Supercomputing and Networking Center is an external application to the VECMA project. It is tailored towards simulating air quality – pollutants concentration and dispersion – over complex urban areas. The multiscale model is provided as a workflow of two different models: i) numerical weather prediction, which simulates chemical transportation and mixing at mesoscale to city scale, and ii) a numerical solver for city-scale geophysical flows for accurate simulation of contaminants transportation around and over buildings. So far, we have focused on UQ techniques for the latter model. Its uncertain parameters are related to the pollutants emission line and area sources, coming from vehicle transportation and household appliances respectively. Taking line sources as an example, the uncertain parameters are: the ratio of petrol to diesel cars and the fuel usage or petrol density. These parameters are selected via a uniform distribution. Once selected, QCG-PilotJob [1] is used to run the air quality model for each of the samples generated (ensemble). The EasyVVUQ library carries out the UQ analysis to the QoI. We envision carrying out SA for each of the uncertain input parameters to evaluate interactions between them.

For the numerical weather prediction model, we want to focus on wind resource characterization and forecast, which is crucial for pollutants transportation. The uncertain parameters concern

surface and planetary boundary layers, e.g. surface roughness level, turbulence kinetic energy, the Prandtl number, Monin-Obukov stability parameter or von Karman's constant [36]. Using qMC sampling and EasyVVUQ, we will provide SA to determine the most important model parameters.

3 Conclusions

In this deliverable, the fast-track implementation of VVUQ into WP4 applications is presented, along with example results on SA and UQ. Out of the 8 applications within our portfolio, 6 have implemented non-intrusive UQ methods at the time of writing, and work is ongoing on 2 other applications (including one external). This exceeds our expectations since there were only 4 applications listed on the fast-track initially. This achievement is credited to the efficient interaction between the applications (WP4) and technical work packages (WP2-WP3-WP5), which resulted in the early release of high quality software (e.g. VECMAtk). The toolkit has been proven to be simple to use for implementing non-intrusive methods into our applications across a range of different domains.

While there remain important challenges to be tackled within the VECMAtk, we have decided to shift the development effort away from software and focus more on the applications. In particular, we are creating application-specific development teams that allow application and software developers to work together and integrate desired tools into the simulation models. Such collaboration shall help the applications as they implement the more-complex semi-intrusive UQ techniques and systematically introduce VV techniques.

4 References

- [1] Groen D. et al. (2019) Introducing VECMAtk - Verification, Validation and Uncertainty Quantification for Multiscale and HPC Simulations. In: Rodrigues J. et al. (eds) Computational Science – ICCS 2019. ICCS 2019. Lecture Notes in Computer Science, vol 11539. Springer, Cham
- [2] R. Preuss and U. von Toussaint, "Investigations on Bayesian uncertainty quantification with two examples," in AIP Conference Proceedings, 2016.
- [3] Xiu, Dongbin, and Jan S. Hesthaven. "High-order collocation methods for differential equations with random inputs." *SIAM Journal on Scientific Computing* 27.3 (2005): 1118-1139.
- [4] Crestaux, Thierry, Olivier Le Maître, and Jean-Marc Martinez. "Polynomial chaos expansion for sensitivity analysis." *Reliability Engineering & System Safety* 94.7 (2009): 1161-1172.
- [5] Eck, Vinzenz Gregor, et al. "A guide to uncertainty quantification and sensitivity analysis for cardiovascular applications." *International journal for numerical methods in biomedical engineering* 32.8 (2016): e02755.
- [6] Sobol, I. M. (1998), "On quasi-Monte Carlo integrations", *Mathematics and Computers Simulation* 47 (2-5), 103–112
- [7] Smolyak, S. (1963). Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR* 148, 1042–1045.
- [8] Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., and Tarantola, S. (2010). "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index". *Comput. Phys. Commun.* 181, 259–270.

- [9] Scott, Bruce D. "Free-energy conservation in local gyrofluid models." *Physics of Plasmas* 12.10 (2005): 102307.
- [10] O.O. Luk, O. Hoenen, A. Bottino et al. ComPat framework for multiscale simulations applied to fusion plasmas, *Computer Physics Communications* (2019), <https://doi.org/10.1016/j.cpc.2018.12.021>.
- [11] Sobol', Il'ya Meerovich. "On the distribution of points in a cube and the approximate evaluation of integrals." *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* 7.4 (1967): 784-802.
- [12] Lemieux, C. "Monte Carlo and Quasi-Monte Carlo Sampling. 2009."
- [13] Sobol, I.M. and Asotsky, D. I. "One more experiment on estimating high-dimensional integrals by quasi-Monte Carlo methods." *Mathematics and Computers in Simulation* 62.3-6 (2003): 255-263.
- [14] Caiazzo A, Evans D, Falcone JL, Hegewald J, Lorenz E, Stahl B, Wang D, Bernsdorf J, Chopard B, Gunn J, Hose R, Krafczyk M, Lawford P, Smallwood R, Walker D, Hoekstra A. A complex automata approach for in-stent restenosis: two-dimensional multiscale modelling and simulations. 2011;2(1):9-17. <https://doi.org/10.1016/j.jocs.2010.09.002>.
- [15] Nikishova, A., et al. "Semi-intrusive multiscale metamodelling uncertainty quantification with application to a model of in-stent restenosis." *Philosophical Transactions of the Royal Society A* 377.2142 (2019): 20180154.
- [16] Nikishova, Anna, et al. "Uncertainty quantification of a multiscale model for in-stent restenosis." *Cardiovascular engineering and technology* 9.4 (2018): 761-774.
- [17] Zun, P. S., T. Anikina, A. Svitenkov, and A. G. Hoekstra. A comparison of fully-coupled 3D in-stent restenosis simulations to in-vivo data. *Front. Physiol.* 8(May):284, 2017
- [18] Tahir, H., C. Bona-Casas, and A. G. Hoekstra. Modelling the effect of a functional endothelium on the development of in-stent restenosis. *PLoS ONE* 8(6):1–9, 2013.
- [19] M. Eldred, J. Burkardt, "Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification", 47th AIAA aerospace sciences meeting, 2009.
- [20] W. Edeling, D. Groen, "FabUQCampaign", GitHub repository, <https://github.com/wedeling/FabUQCampaign>, 2019.
- [21] D. Groen et al., "FabSim: facilitating computational research through automation on large-scale and distributed e-infrastructures", *Computer Physics Communications*, 207, 375--385, 2016.
- [22] UNHCR, "Figures at a Glance," United Nations High Commissioner for Refugees. Available at: <https://www.unhcr.org/uk/figures-at-a-glance.html>
- [23] D. Groen, "Simulating Refugee Movements: Where would You Go?," *Procedia Computer Science*, vol. 80, pp. 2251-2255, 2016.
- [24] D. Suleimenova, D. Bell and D. Groen, "A generalized simulation development approach for predicting refugee destinations," *Scientific Reports*, vol. 7, no. 1, p. 13377, 2017.
- [25] D. Suleimenova, D. Bell and D. Groen, "Towards an automated framework for agent-based simulation of refugee movements," in 50th Winter Simulation Conference (WSC17), Las Vegas, 2017.
- [26] D. Suleimenova and D. Groen, "How policy decisions affect refugee journeys in South Sudan: A study using automated ensemble simulations," *Journal of Artificial Societies and Social Simulation (JASSS)*, (submitted).
- [27] G. Tang, G. Iaccarino and M.S. Eldred, "Global Sensitivity Analysis for Stochastic Collocation," in 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Orlando, Florida, 2010.
- [28] Vassaux, M., Richardson, R. A., & Coveney, P. V. (2019). The heterogeneous multiscale method applied to inelastic polymer mechanics. *Philosophical Transactions of the Royal Society A*, 377(2142), 20180150.

- [29] Vassaux, M., Sinclair, R. C., Richardson, R. A., Suter, J. L., & Coveney, P. V. (2019). The Role of Graphene in Enhancing the Material Properties of Thermosetting Polymers. *Advanced Theory and Simulations*, 2(5), 1800168.
- [30] S. K. Sadiq, D.W. Wright, S. J. Watson, S. J. Zasada, I. Stoica, P. V. Coveney, "Automated molecular simulation based binding affinity calculator for ligand-bound HIV-1 proteases.", *J Chem Inf Model*. 2008 Sep;48(9):1909-19. doi: 10.1021/ci8000937.
- [31] David W. Wright, Shunzhou Wan, Christophe Meyer, Herman van Vlijmen, Gary Tresadern, Peter V. Coveney, "Application of ESMACS binding free energy protocols to diverse datasets: Bromodomain-containing protein 4", *Scientific Reports* 9, 6017, (2019), DOI:10.26434/chemrxiv.7327019
- [32] Wright, David; Hussein, Fouad; Wan, Shunzhou; Meyer, Christophe; Van Vlijmen, Herman; Tresadern, Gary; et al., "Application of the ESMACS Binding Free Energy Protocol to a Highly Varied Ligand Dataset: Lactate Dehydrogenase A.", (2019), ChemRxiv. Preprint. DOI: 10.26434/chemrxiv.8398055.v1
- [33] S. Wan, G. Tresadern, L. Pérez-Benito, H. van Vlijmen, "Accuracy and Precision of Alchemical Relative Free Energy Predictions With and Without Replica-Exchange", *Advanced Theory and Simulations*, Accepted
- [34] V. Balasubramanian, S. Jha, A. Merzky and M. Turilli, RADICAL-Cybertools: Middleware Building Blocks for Scalable Science, arXiv:1904.03085
- [35] Patronis, A., Richardson, R. A., Schmieschek, S., Wylie, B. J., Nash, R. W., & Coveney, P. V. (2018). Modeling Patient-Specific Magnetic Drug Targeting Within the Intracranial Vasculature. *Frontiers in physiology*, 9, 331.
- [36] Berg, L., Yang, B., Qian, Y., Ma, P.-I., Wharton, S., Bulaevskaya, V., Yan, H., & Shaw, W. (2015, June). Understanding model uncertainty – an application of uncertainty quantification to wind energy.