D3.5: Documentation and tutorials accompanying the deep-track release of the VECMA toolkit

Due Date	14 August 2020
Delivery	11 August 2020
Submission of updated version	N/A
Lead Partner	UBRU
Dissemination Level	Public
Status	Final
Approved	Executive Board
Version	V1.2

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 800925.

DOCUMENT INFO

Date and version number	Author	Comments
17.07.2020 v1.0	Diana Suleimenova	First Draft
24.07.2020	Federica Gugole	Received review and feedback
30.07.2020	Vytautas Jancauskas	Received review and feedback
3.08.2020 v1.1	Diana Suleimenova	Addressed internal reviewers' comments
8.08.2020 v1.1	Peter Coveney	Sent final comments and approval
10.08.2020 v1.2	Diana Suleimenova	Addressed final comments and finalised report

CONTRIBUTORS

- Diana Suleimenova, UBRU – Editor, Author
- Derek Groen, UBRU – Author
- Hamid Arabnejad, UBRU – Author
- Xuanye Gu, UCL – Contributor
- Bartosz Bosak, PSNC – Contributor
- Hugh Martin, CBK – Contributor
- Peter Coveney, UCL – Contributor
- Wouter Edeling, CWI – Contributor
- Federica Gugole – Reviewer
- Vytautas Jancauskas – Reviewer

Disclaimer

Definition and Acronyms

Acronym	Definition
ADE	Advection Diffusion Equation
EasyVVUQ	A tool for constructing and performing Verification Validation and Uncertainty Quantification procedures, intended to be easy to use.
FabCovid19	Fabric for Flu and Coronavirus Simulations
FabSim3	Fabric for Simulation
FabMD	Fabric for Molecular Dynamics Domain
FabMogp	Fabric for Multi-Output Gaussian Process Emulator
FabUQCampaign	Fabric for Uncertainty Quantification Campaign of climate modelling
ISR3D	The 3D In-Stent Restenosis
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator
Mogp	Multi-Output Gaussian Process Emulator
MUSCLE3	MUltiScale Coupling Library and Environment
Ocean2D	Two-dimensional Ocean model
PCE	Polynomial Chaos Expansion
QCG	Quality in Cloud and Grid
QCGPJ	Quality in Cloud and Grid Pilot Job mechanism
SC	Stochastic Collocation
SMC	Smooth Muscle Cells
UQ	Uncertainty Quantification
VECMA	Verified Exascale Computing for Multiscale Applications
VECMAtk	Toolkit for Verified Exascale Computing for Multiscale Applications
VVUQ	Verification Validation and Uncertainty Quantification
WP	Work Package

Table of Contents

List of Figures

1 Executive Summary

This document reports on documentation and tutorials accompanying the deep-track release of the VECMA toolkit (VECMAtk), which includes six internal and two external applications from various domains using seven components of VECMAtk (i.e FabSim3, EasyVVUQ, QCG-PilotJob, QCG-Broker, QCG-Client, QCG-Now and MUSCLE3) to perform Verification, Validation and Uncertainty Quantification (VVUQ) analysis.

In this deliverable, we describe the structure of VECMAtk tutorials, and the strategy undertaken to create, organise and publicly share these tutorials with existing and new users.

To grow our user base and accommodate new learners from various domains, we developed static, interactive and video tutorials for end-users. Each of these categories have their distinguishing features for users choosing to learn all about VECMAtk. Specifically, the static tutorials are detailed read-only manuals following the same outline and visual structures and capturing all components and applications within the VECMA project. The interactive tutorials are easy to use without installation requirements of components and can be considered as a portable training platform using Jupyter Notebooks. Lastly, the tutorial videos are a new millennial-style learning experience, which can be accessed and watched on the go and at any time.

2 Introduction

This deliverable reports on documentation and tutorials accompanying the VECMA toolkit (VECMAtk) components and applications. We describe the structure and the strategy undertaken for this deliverable, highlight features and functionalities of the VECMAtk components, and provide links to all available documentation and tutorials of the VECMA applications. This deliverable is part of VECMA's Work Package 3 (WP3), whose aim is to develop a software toolkit that enables automated Verification, Validation and Uncertainty Quantification (VVUQ) of multiscale applications that can be deployed on exascale platforms. A detailed description of the VECMA software stack as a whole is given in D5.1 and D5.2, while a systematic overview of the VECMA applications is provided in D4.1. Hence, in this deliverable we will limit ourselves to providing a selection of all the material for scope and space reasons. The VECMAtk is one of the major outcomes of the VECMA project and documentation and tutorials described in this deliverable accompanied the major release (M24) with wide-ranging dissemination activities (WP6).

3 The VECMA toolkit documentation and tutorials

Software development requires documentation in order to communicate information to stakeholders or end-users on how to use and operate it. The purpose of documentation is to describe architecture and functionalities, as well as to provide instructions on installation, testing and troubleshooting [1]. Within the VECMA project, we develop documentation and tutorials for VECMAtk to guide and teach existing and new users on how to perform VVUQ analysis [2]. All documentation and tutorials are easily accessible, descriptive and illustrative with the VECMA applications ranging within various domains.

To attract users from different domains and provide distinctive experience, we developed static, interactive and video tutorials for end-users, as illustrated in Figure 1. The static tutorials provide informative read-only information for the VECMAtk components with internal and external application instances. The interactive tutorials are easy-to-use as installation of the VECMAtk components are not required and they are versatile because users are able to append with additional remarks. Video tutorials instead provide an audio-visual learning experience to users and in turn, users independently teach themselves about VECMAtk.

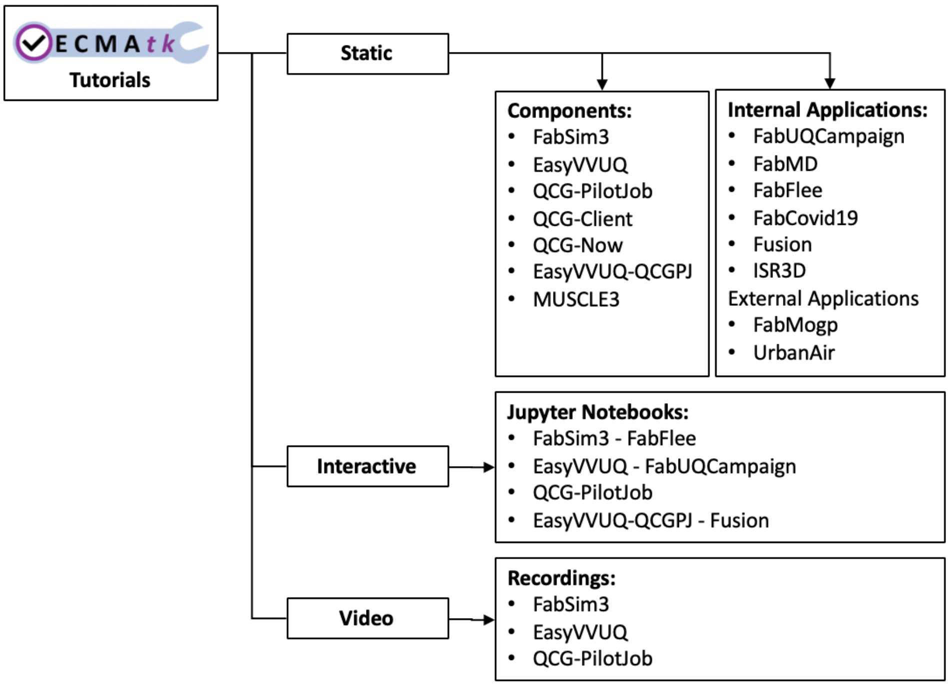


Figure 1: The structure of VECMAtk tutorials.

We attempt to provide various learning resources to the VECMAtk users, review and modify tutorials according to the user feedback and measure traffic usage from Jupyter Notebooks. In the following sections of this deliverable, we describe each tutorial type and provide references to these tutorials.

3.1 Static tutorials

The static tutorials are available for FabSim3, EasyVVUQ, Quality in Cloud and Grid (QCG) tools, EasyVVUQ-QCGPJ and MUSCLE3 components of VECMAtk. We provide tutorials for each of these components using the ReadTheDocs[1] platform or markdown language. We also describe installation and testing, as well as illustrate how to use and perform various computational tasks.

We present a range of tutorials pertaining **FabSim3[2]**, which is an automation toolkit written in Python 3 for organising input, output and environment information. It creates a consistent log and makes it possible to repeat or reproduce runs, enabling the execution of simulation and analysis tasks on supercomputers and featuring an integrated test infrastructure and a flexible plugin system [3]. There are several plugins available from a diverse range of scientific domains, which are described in the VECMAtk applications subsection. The FabSim3 static tutorial is available on the ReadTheDocs platform and it explains installation, testing, creation of new automation scripts and plugins for new applications, as well as configuration steps for remote executions. In Figure 2, we illustrate the initial page layout of FabSim3 (https://fabsim3.readthedocs.io), which provides general information about the automation tool and has an easy-to-navigate table of content.

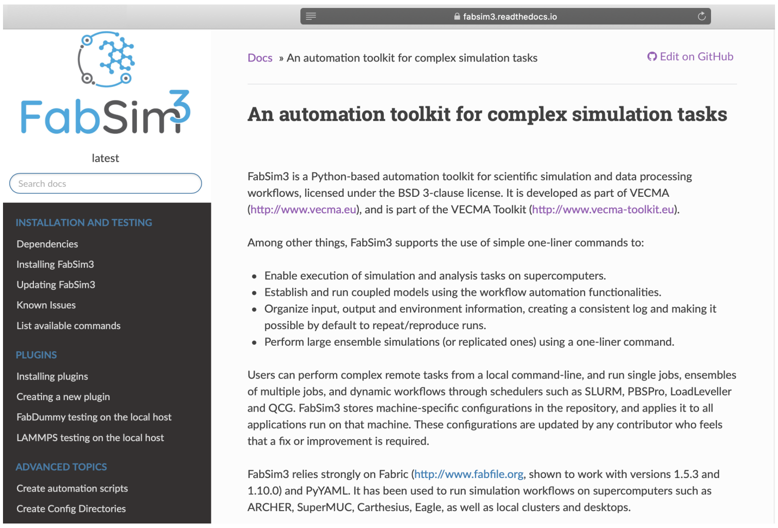


Figure 2: Overview of the FabSim3 static tutorial.

EasyVVUQ[3] is another component for which we provide a range of tutorials. It is a Python library designed to facilitate VVUQ of a wide variety of applications [4, 5]. The EasyVVUQ static tutorial[4] is also hosted on the ReadTheDocs platform and describes the installation and conceptual basis of the

[1] https://readthedocs.org
[2] https://github.com/djgroen/FabSim3
[3] https://github.com/UCL-CCS/EasyVVUQ
[4] https://easyvvuq.readthedocs.io/en/dev/index.html

framework. Moreover, this tutorial offers practical examples of various UQ sampling techniques that are available in EasyVVUQ and are applied to the VECMAtk applications (see Figure 3).

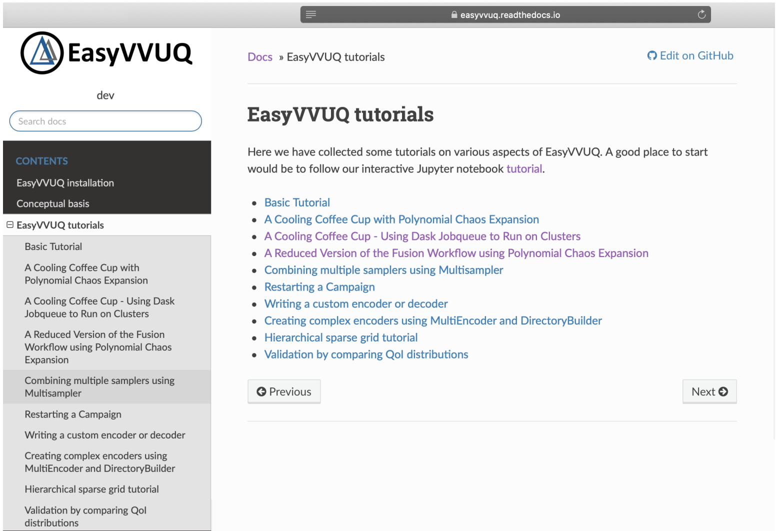


Figure 3: EasyVVUQ example tutorials list accompanied by the VECMA applications.

There are tutorials for four different UQ sampling techniques. The first UQ technique considered here is the stochastic collocation (SC), which is applied to a simplified two-dimensional ocean model (Ocean2D)[5], to the advection diffusion equation (ADE)[6] and to migration modelling[7] application instances.

The second UQ technique is the Polynomial Chaos Expansion (PCE) that has example tutorials with regard to fusion[8] [6] and migration modelling[8] applications. The Random Sampler technique is also available and has an example of an ensemble of LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) simulations[9]. The last technique is Latin Hypercube, which is applied to an earthquake model[10].

There are two example tutorials available on validation patterns, namely ensemble output validation[11] in the context of migration modelling and quantity of interest distribution[12] extraction in an application to fusion.

[5] https://github.com/wedeling/FabUQCampaign/blob/master/Tutorial_ocean.md
[6] https://github.com/wedeling/FabUQCampaign/blob/master/Tutorial_ADE.md
[7] https://github.com/djgroen/FabFlee/blob/master/doc/TutorialSensitivity.md
[8] https://easyvvuq.readthedocs.io/en/dev/fusion_tutorial.html
[9] https://github.com/UCL-CCS/FabMD/blob/master/doc/EasyVVUQ_FabMD_example.md
[10] https://github.com/edaub/vecma_workshop_tutorial/blob/master/Tutorial.rst
[11] https://github.com/djgroen/FabFlee/blob/master/doc/TutorialValidate.md
[12] https://github.com/UCL-CCS/EasyVVUQ/blob/dev/docs/validate_similarities.rst

A suite of components that we provide tutorials for are the Quality in Cloud and Grid (QCG) tools (see http://www.qoscosgrid.org), which are integrated systems offering advanced job and resource management capabilities and delivering supercomputer-like performance and structure to end-users [7]. QCG consists of the following tools:

- **QCG-PilotJob**[13] is a Pilot Job system that allows to execute many subordinate jobs in a single scheduling system allocation. Users can access the QCG-PilotJob tutorial (https://qcg-pilotjob.readthedocs.io) and read information about installation, usage and execution.
- **QCG-Broker**[14] is an open-source meta-scheduling framework that allows developers to build and easily deploy resource management systems.
- **QCG-Client**[15] is a command-line client for execution of computing jobs on the clusters.
- **QCG-Now**[16] is a desktop GUI client for easy execution of computing jobs on the clusters.

Tutorials for QCG-Broker, QCG-Client and QCG-Now are hosted on the QCG website. To showcase their integration and usage in the context of the VECMA project, there is an example tutorial of a simple cooling coffee cup model[17], which explains the execution process using QCG-Client and QCG-Now.

EasyVVUQ-QCGPJ[18] is a lightweight integration plugin for parallelizing EasyVVUQ with the QCG-PilotJob mechanism. The tool provides API that can be effortlessly integrated into typical EasyVVUQ workflows to enable parallel processing of demanding operations, in particular the simulation model's executions and encodings. It works regardless of whether one runs one's use-case on a multi-core laptop or on a large HPC machine. The EasyVVUQ-QCGPJ static tutorial[19] provides a more detailed description on how to install, test and use the software with an example guide as demonstrated in Figure 4.

---

13 https://github.com/vecma-project/QCG-PilotJob
14 http://www.qoscosgrid.org/trac/qcg-broker
15 http://www.qoscosgrid.org/trac/qcg/wiki/user_information
16 http://www.qoscosgrid.org/qcg-now/en/
17 https://easyvvuq-qcgpj.readthedocs.io/en/plugin/tutorials/cooling_cup/guide/tutorial.html#execution-with-qcg-client
18 https://github.com/vecma-project/EasyVVUQ-QCGPJ
19 https://easyvvuq-qcgpj.readthedocs.io/en/plugin/index.html

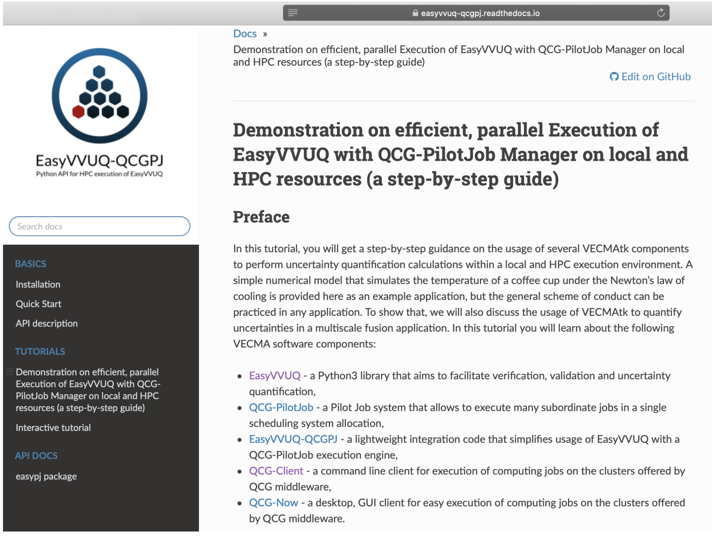


Figure 4: Overview of the EasyVVUQ-QCGPJ static tutorial.

An external VECMAtk component is **MUSCLE3**[20], which is the third incarnation of the Multiscale Coupling Library and Environment [8]. It also has a tutorial published on the ReadTheDocs platform with a complete user documentation https://muscle3.readthedocs.io. The static MUSCLE3 tutorial shows how it can be installed and used from three programming languages: Python, C++ and Fortran.

3.1.1 VECMA applications

The VECMA applications are categorised into internal and external applications. They use the components described above. We show the VECMAtk Tube Map for these applications and their integration with the various components in Figure 5. Within the internal applications, for most applications we provide FabSim3 plugins that begin with the term `Fab`. These applications have descriptive tutorials that have been developed using markdown language and can be found in their own GitHub repositories. The static tutorials for internal applications provide a generic introduction to the toolkit, display the interactions with the VECMAtk components by means of tube maps, and explain their installation and usage. These tutorials are:

- The FabSim3 plugin for climate modelling (FabUQCampaign)[21] aims to analyse ADE and Ocean2D models;
- In the material domain, FabMD plugin[22] simulates LAMMPS;
- Migration is simulated via the agent-based modelling toolkit Flee, which is purpose-built for simulating the movement of individuals across geographical locations using FabFlee plugin[23];

[20] https://github.com/multiscale/muscle3
[21] https://github.com/wedeling/FabUQCampaign
[22] https://github.com/UCL-CCS/FabMD
[23] https://github.com/djgroen/FabFlee/blob/master/doc/FabFlee.md

- The FabSim3 plugin for flu and coronavirus simulations (FabCovid19)[24] is an agent-based model that mimics the spread of flu and coronavirus in local regions [9].

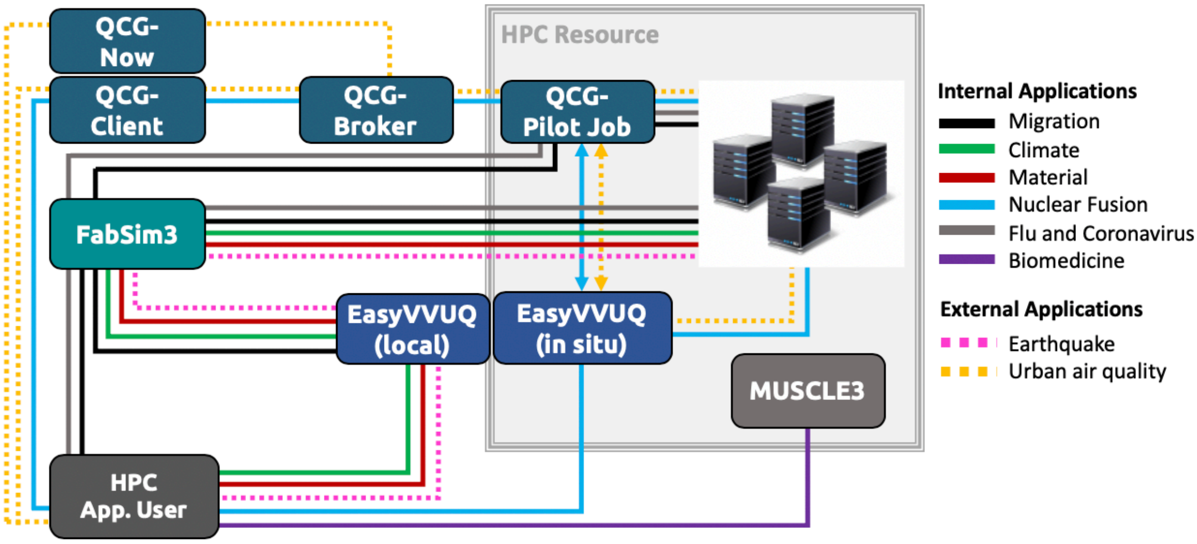


Figure 5: Tube Map showing which VECMAtk components are used in which tutorials. VECMAtk components are given in boxes, and the tutorials for internal and external applications are indicated using coloured lines. Note that, in particular, EasyVVUQ can be used both on the local desktop for ease of use, or on a remote HPC resource for improved performance.

The other internal applications are nuclear fusion[25], which studies the turbulence effects on plasma transport at larger scales, and the in-stent restenosis 3D (ISR3D) biomedical application[26] that simulates the smooth muscle cells (SMC) proliferation and restenosis process. Since these applications are in private GitHub repositories, users who have an interest in these applications are required to request an access. Nonetheless, introductory tutorials for these applications can be accessed in the VECMA tutorials repository (see https://github.com/vecma-project/VECMA-tutorials).

Currently, there are two external applications that use VECMAtk: FabMogp and UrbanAir. FabMogp[27] is an external FabSim3 plugin for earthquake modelling using the Mogp emulator, which is a Python package for fitting Gaussian Process Emulator to computer simulation results. The FabMogp static tutorial[28] is comprehensive and educates users by outlining installation and integration with the

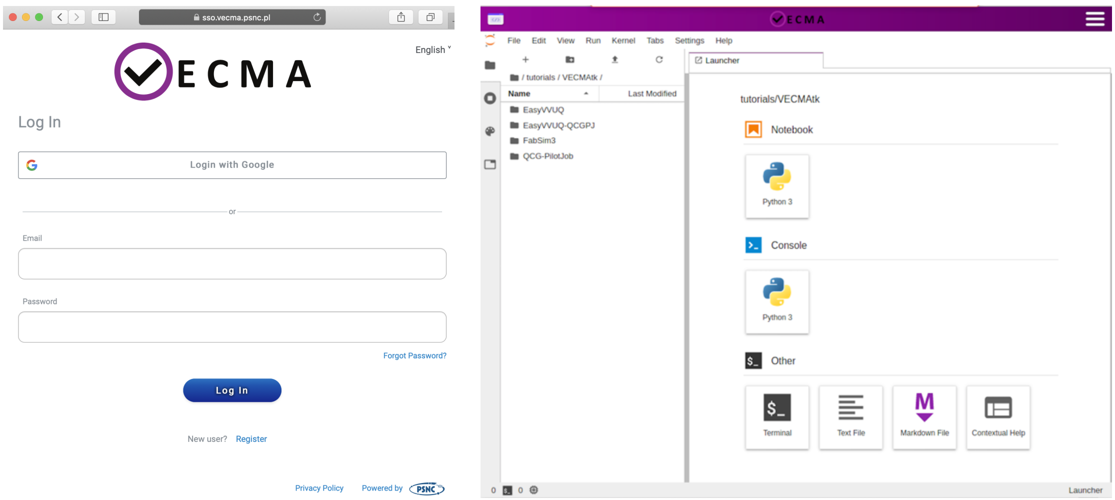
---

[24] https://github.com/djgroen/FabCovid19/blob/master/README.md
[25] https://www.vecma.eu/application-nuclear-fusion/
[26] https://github.com/vecma-project/VECMAtk/blob/master/VECMAtk_static_tutorials/ISR3D_installation_guide.md
[27] https://github.com/edaub/fabmogp
[28] https://github.com/edaub/fabmogp/blob/master/Tutorial.rst

VECMAtk components for execution and analysis of earthquake models. The other external application UrbanAir[29] assesses the air quality in complex urban areas at street level and is stored in a private repository with limited access. However, we created a basic tutorial for this external application that can be accessed in the VECMA tutorials repository.

3.2 Interactive tutorials

The interactive tutorials aim to provide a portable training environment without requiring the installation of the VECMAtk components. Specifically, these tutorials focus on trying out FabSim3, EasyVVUQ, QCG-PilotJob and EasyVVUQ-QCGPJ using example applications. These interactive tutorials offer unique opportunities to teach and learn independently on how to perform VVUQ analysis. We host interactive tutorials in the VECMA JupyterHub environment (https://jupyter.vecma.psnc.pl) as pictured in Figure 6 that is available to users upon registration or can be accessed using a Google account. This allows us to monitor and measure user traffic accessing and testing VECMAtk. Moreover, users are able to add explanatory text and their own code, perform analysis and share and visualise results with others. Importantly, the VECMA JupyterHub allows everyone to try out different components and applications of VECMAtk on a single platform.



a) Login page b) Main page

Figure 6: Overview of the VECMA JupyterHub for interactive tutorials.

[29] https://github.com/vecma-project/VECMAtk/blob/master/VECMAtk_static_tutorials/UrbanAir.md

3.3 Video tutorials

We provide video content to supplement written tutorials. Specifically, we created tutorial videos for FabSim3, EasyVVUQ and QCG-PilotJob to provide a successful learning experience by watching videos on the go. The VECMAtk tutorial videos are publicly shared on the personal YouTube channel VECMA FET-HPC[30]. In this channel, individuals can watch tutorial videos introducing VECMAtk, illustrating how to install and test FabSim3 on Linux, macOS and Microsoft Windows operating systems and explaining the differences between the various QCG tools.

4 Conclusion

In this deliverable, documentation and tutorials accompanying the deep-track release of VECMAtk are presented, along with the structure and strategy undertaken for a unified development and dissemination of tutorials. Existing and new users of VECMAtk are able to access three types of supporting content: static, interactive and video tutorials. These tutorials demonstrate and educate existing and new users on the VECMAtk components with eight application examples, such as migration, climate, material and others, aiming to accommodate different types of learners and scientists from various domains. It is important to note that we also intend to dedicate our efforts in the continuous improvement and revision of the VECMAtk tutorials as new functionalities and cross-component integrations emerge in the project, as well as in increasing the number of interactive and video tutorials between now and M36 release.

5 References

[1] Forward, A., and Lethbridge, T.C. (2002) The relevance of software documentation, tools and technologies: A survey. In Proceedings of the 2002 ACM symposium on Document engineering (DocEng '02). Association for Computing Machinery, New York, NY, USA, 26–33. DOI: 10.1145/585058.585065.

[2] 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. and Coveney, P. V. (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. DOI: 10.1007/978-3-030-22747-0_36.

[30] https://www.youtube.com/channel/UCDebwno9Fbm5azYA4tZjmUQ

[3] Groen, D., Bhati, A. P., Suter, J., Hetherington, J., Zasada, S. J. and Coveney, P. V. (2016). FabSim: Facilitating computational research through automation on large-scale and distributed e-infrastructures, Computer Physics Communications 207: 375–385. DOI: 10.1016/j.cpc.2016.05.020.

[4] Richardson, R. A., Wright, D. W., Edeling, W., Jancauskas, V., Lakhlili, J. and Coveney, P. V. (2020) EasyVVUQ: A library for verification, validation and uncertainty quantification in high performance computing, *Journal of Open Research Software*, 8(1), 11. DOI: 10.5334/jors.303.

[5] Wright, D. W. , Richardson, R. A. , Edeling, W. , Lakhlili, J., Sinclair, R. C., Jancauskas, V., Suleimenova, D., Bosak, B., Kulczewski, M., Piontek, T., Kopta, P., Chirca, I., Arabnejad, H., Luk, O. O., Hoenen, O., Weglarz, J., Crommelin, D., Groen, D., and Coveney, P. V. (2020) Building confidence in simulation: Applications of EasyVVUQ, *Advanced Theory and Simulations*, 1900246. DOI: 10.1002/adts.201900246.

[6] Lakhlili J., Hoenen O., Luk O. O. and Coster D. P. (2020) Uncertainty Quantification for Multiscale Fusion Plasma Simulations with VECMA Toolkit. In: Krzhizhanovskaya V. et al. (eds) Computational Science – ICCS 2020. ICCS 2020. Lecture Notes in Computer Science, vol 12143. Springer, Cham. DOI: 10.1007/978-3-030-50436-6_53.

[7] Piontek, T., Bosak, B., Ciżnicki, M., Grabowski, P., Kopta, P., Kulczewski, M., Szejnfeld, D. and Kurowski, K. (2016) Development of Science Gateways Using QCG — Lessons Learned from the Deployment on Large Scale Distributed and HPC Infrastructures, J Grid Computing. DOI: 10.1007/s10723-016-9384-9.

[8] Veen L.E. and Hoekstra A.G. (2020) Easing Multiscale Model Design and Coupling with MUSCLE 3. In: Krzhizhanovskaya V. et al. (eds) Computational Science – ICCS 2020. ICCS 2020. Lecture Notes in Computer Science, vol 12142. Springer, Cham. DOI: 10.1007/978-3-030-50433-5_33.

[9] Mahmood, I., Arabnejad, H., Suleimenova, D., Sassoon, I., Marshan, A., Serrano, A., Louvieris, P., Anagnostou, A., Taylor, S. J. E., Bell, D. and Groen, D. (in press) FACS: A geospatial agent-based simulator for analyzing COVID-19 spread and public health measures on local regions. *Journal of Simulation.*