State of the art
Methods and Tools in VVUQ

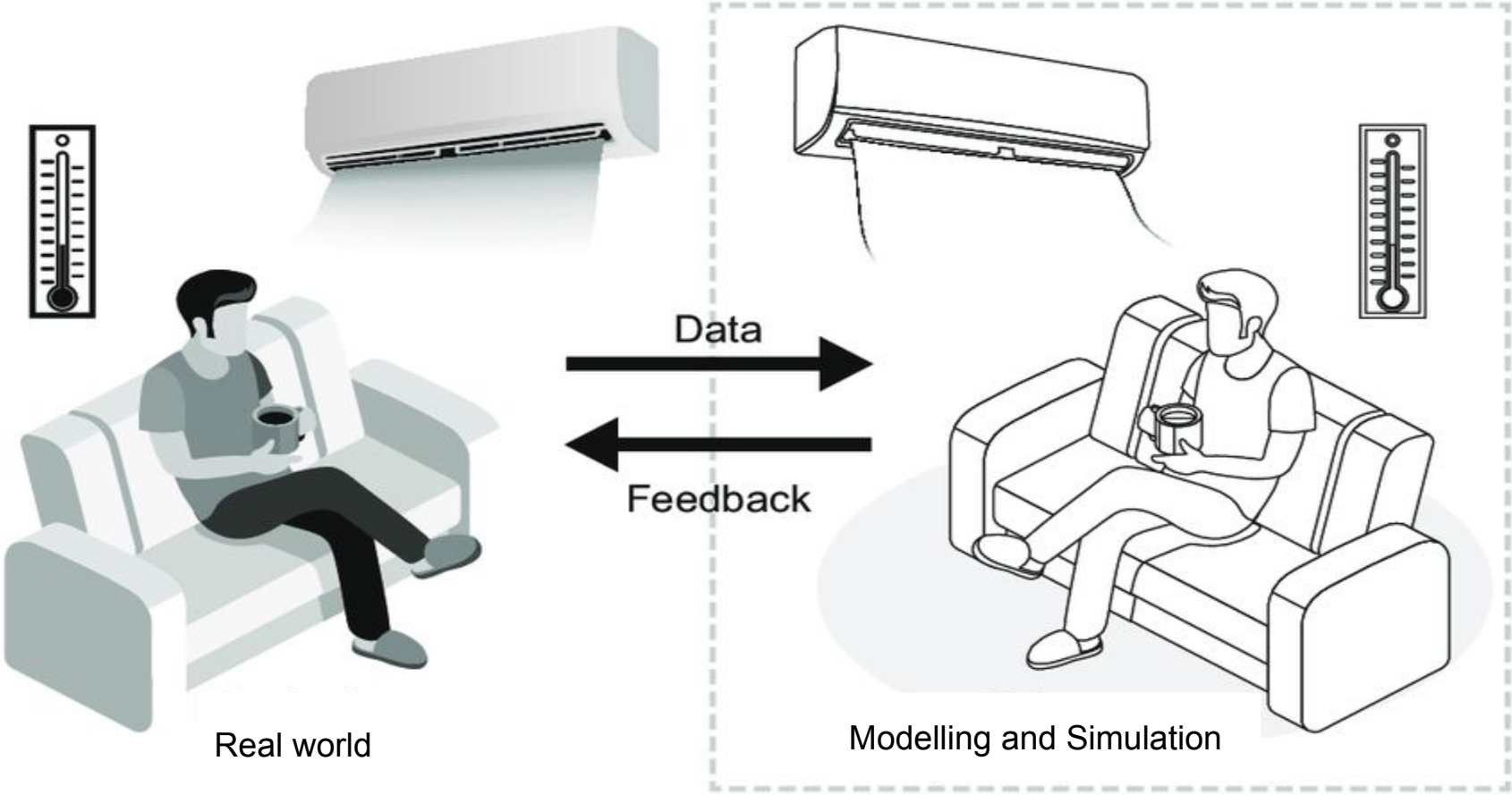
Diana Suleimenova
Department of Computer Science, Brunel University London

The Role of Uncertainty in Mathematical Modelling of Pandemics
8 February 2022

ECMA

Brunel University London

SEAVEA

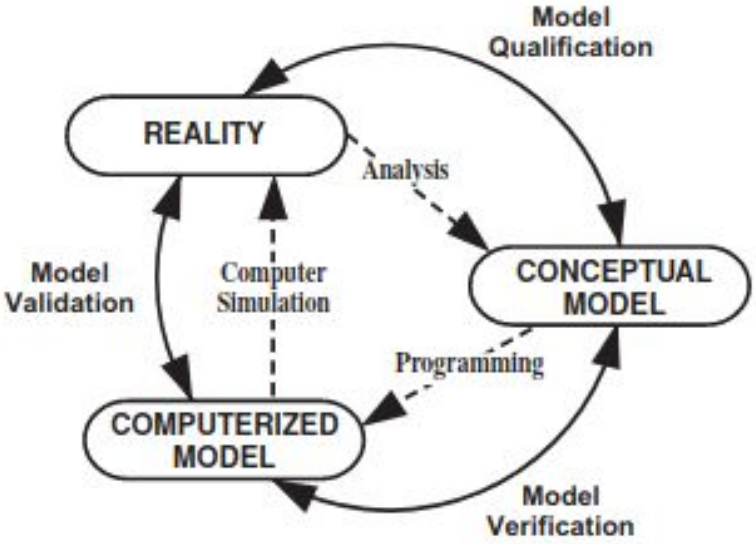


Source: Deguchi et al. (2020)

Concepts and terminology

Society of Computer Simulation (SCS):

- Qualification
  - “Determination of adequacy of the conceptual model to provide an acceptable level of agreement for the domain of intended application.”
- Verification
  - deals with the relationship between the conceptual model and the computerized model.
- Validation
  - deals with the relationship between the computerized model and reality.

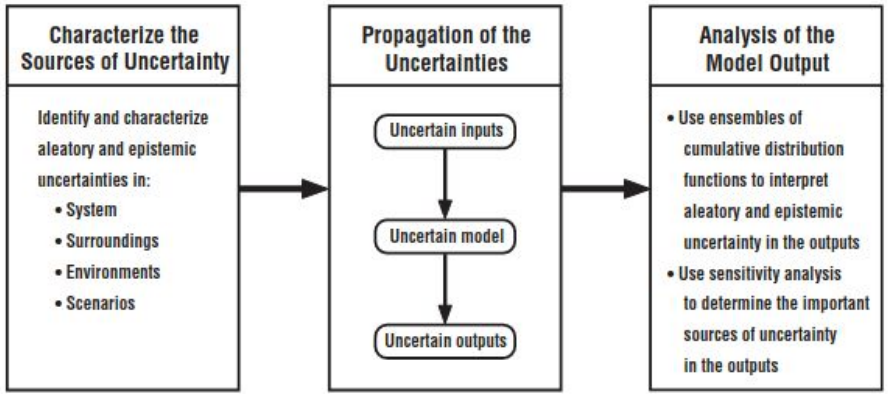


The role of Verification and Validation within the phases of modelling and simulation (Schlesinger, 1979).

Concepts and terminology (cont.)

Uncertainty:

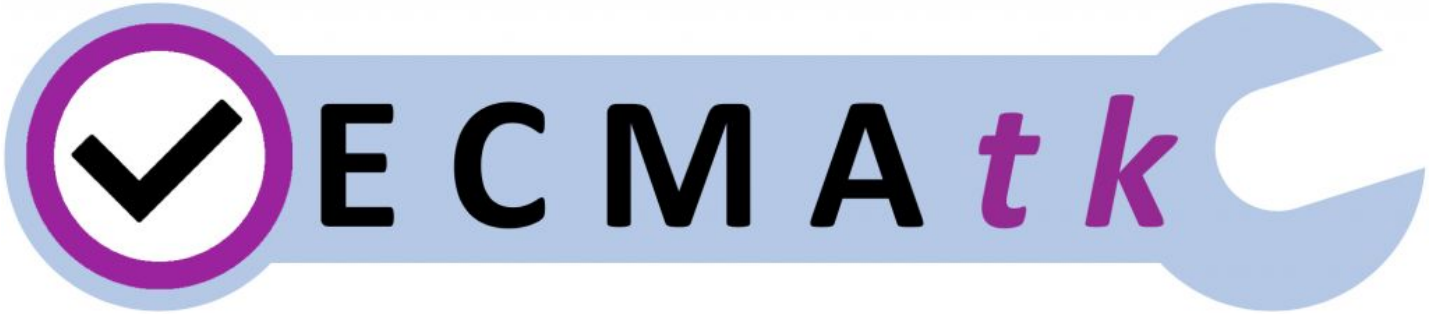
- Aleatory
  - “Uncertainty due to inherent randomness.”
  - describes the natural/intrinsic variability of a quantity of interest (QoI).
- Epistemic
  - “Uncertainty due to lack of knowledge.”
  - describes the lack of knowledge and is potentially reducible by acquiring more knowledge.

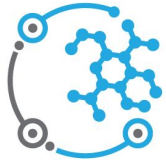


Three basic steps in an uncertainty analysis (Oberkampf and Roy, 2010).

VVUQ Tools

Name	Language	Main Features	License	Cross-platform
Dakota	C/C++[2]	General purpose: uncertainty propagation, surrogate modeling, sensitivity analysis, model calibration, reliability analysis, risk analysis, external code wrapping	LGPL	Yes
DiceDesign	R	Construction of experimental designs	GPLv3	Yes
DiceKriging	R	GP (Kriging) metamodeling	GPLv3	Yes
DiceOptim	R	GP (Kriging)-based optimization	GPLv3	Yes
FERUM	MATLAB	Reliability analysis (FORM, SORM, Subset simulation, etc.), reliability-based design optimization (RBDO), and global sensitivity analysis	GPLv3	Yes
mistral	R	Reliability analysis library (FORM, Importance Sampling, Subset Simulation, etc.)	CeCILLv2	Yes
MUQ	C++/Python	General purpose: surrogate modeling (PCE, GP), constrained optimization, Bayesian inversion)	GPLv2	Yes
OpenCossan	MATLAB	General purpose: uncertainty propagation, surrogate modeling, sensitivity analysis, reliability, robust optimization	LGPL	Yes
OpenTURNS	C++/Python	General purpose: uncertainty propagation, surrogate modeling, sensitivity analysis, reliability, optimization	LGPL	Yes
sensitivity	R	Sensitivity analysis (Sobol' indices, FAST, PCC, etc.) with support for multidimensional outputs	GPLv2	Yes
SIMLab	–	GUI-based sensitivity analysis (Sobol' indices, FAST, Morris, etc.)	Freeware	No (Windows)
SUMO Toolbox	MATLAB	Surrogate modeling (GP, SVM, neural networks, etc.) and surrogate-based optimization	AGPLv3	Yes
UQLab	MATLAB	General purpose: uncertainty propagation, surrogate modeling, sensitivity analysis, reliability analysis, Bayesian inversion, robust optimization, external code wrapping	3-Clause BSD	Yes
UQpy	Python	Uncertainty propagation, stochastic processes	MIT	Yes
UQ Toolkit (UQTk)	C++/Python	Uncertainty propagation, surrogate modelling, sensitivity analysis, Bayesian inversion, external code wrapping	LGPL	Yes





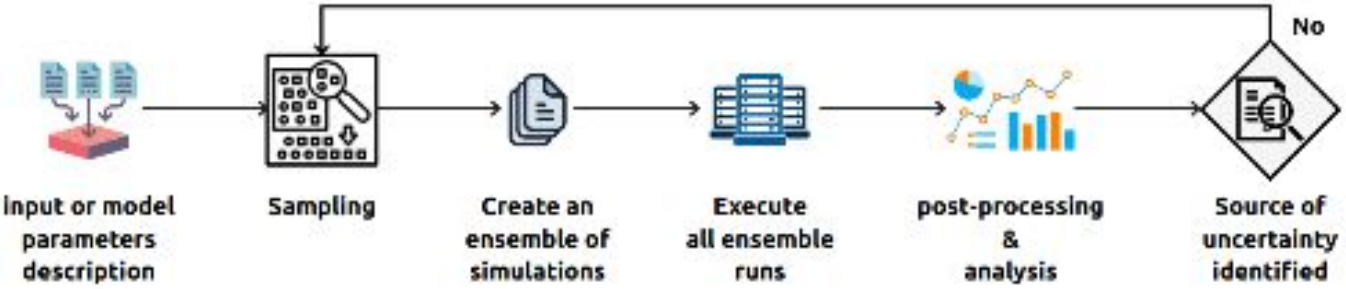
FabSim3

https://www.vecma-toolkit.eu

EasyVVUQ

- EasyVVUQ aims to make it as easy as possible to implement advanced techniques for uncertainty quantification for existing application codes (or workflows).
- Primary focus on non-intrusive UQ techniques, where many model instances are run to quantify uncertainties and analyse parameter sensitivities.

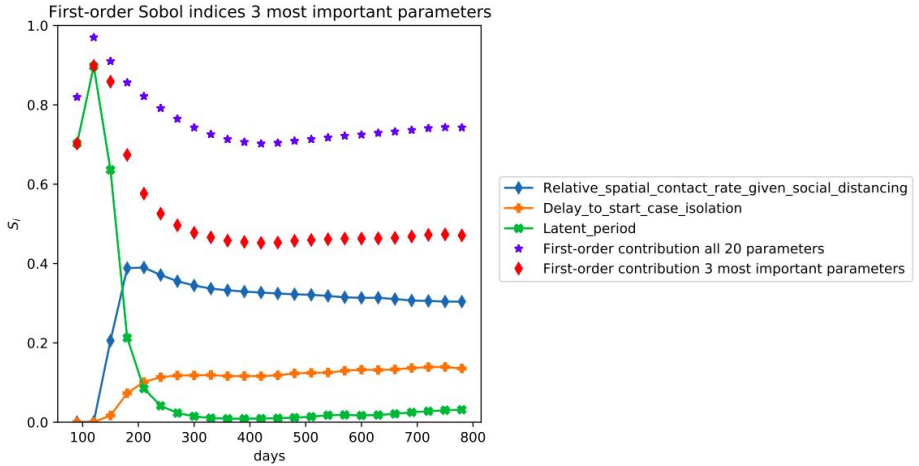
https://easyvvuq.readthedocs.io



Basic UQ workflow

EasyVVUQ samplers

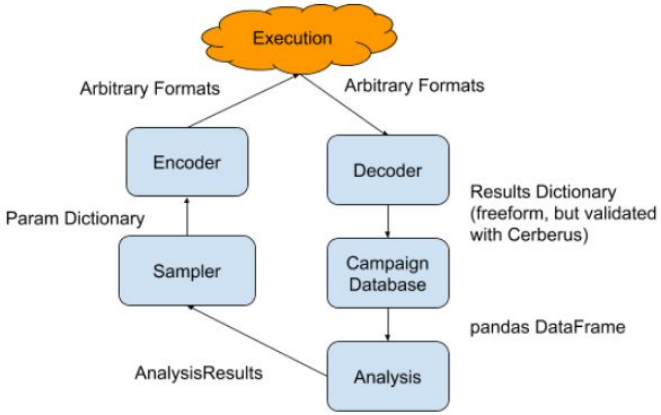
- Monte-Carlo
- Quasi Monte-Carlo
- Polynomial Chaos Expansion (PCE)
- Stochastic Collocation (SC)
- Adaptive SC sparse grids
- All of these can be used to calculate Sobol Sensitivity Indices



EasyVVUQ Enhances Workflow

- Integrated tightly with the rest of Python scientific computing infrastructure.
  - E.g. access through a wide range of sampling approaches using ChaosPy.
- Adapts to users existing code via extension of encoder and decoder classes.
- Minimal boiler-plate code required (minimal set-up overhead).
- Takes care of bookkeeping (creating folders for storing results for example), collects other meta-info in a database.

- Execution can be done:
  - Locally (no tools required)
  - Supercomputers
    (using FabSim3, Dask and/or QCG-PJ)
  - Cloud HPC (using Kubernetes).



FabSim3

- User-developer oriented automation/curation toolkit
  - Aim: reduce human effort required to create/modify/repurpose/salvage complex computational workflows.
- Uses generic pattern code (reusable in other tools) for UQ and V&V.
- Supports range of backends:
  - Localhost.
  - Range of supercomputers.
  - QCG broker & Pilot Jobs.
- FabSim3 is generic, but plugins provide application-specific advantages.

FabSim3 relies on bash one-liners, e.g.:

```
fabsim localhost run_amazingly_complex_app
fabsim eagle validate_flee:mali,cores=24,replicas=5
fabsim localhost install_plugin:FabFlee
```

https://fabsim3.readthedocs.io

Groen et al., (2016). Comp. Phys. Comm. vol. 207, pp. 375--385.

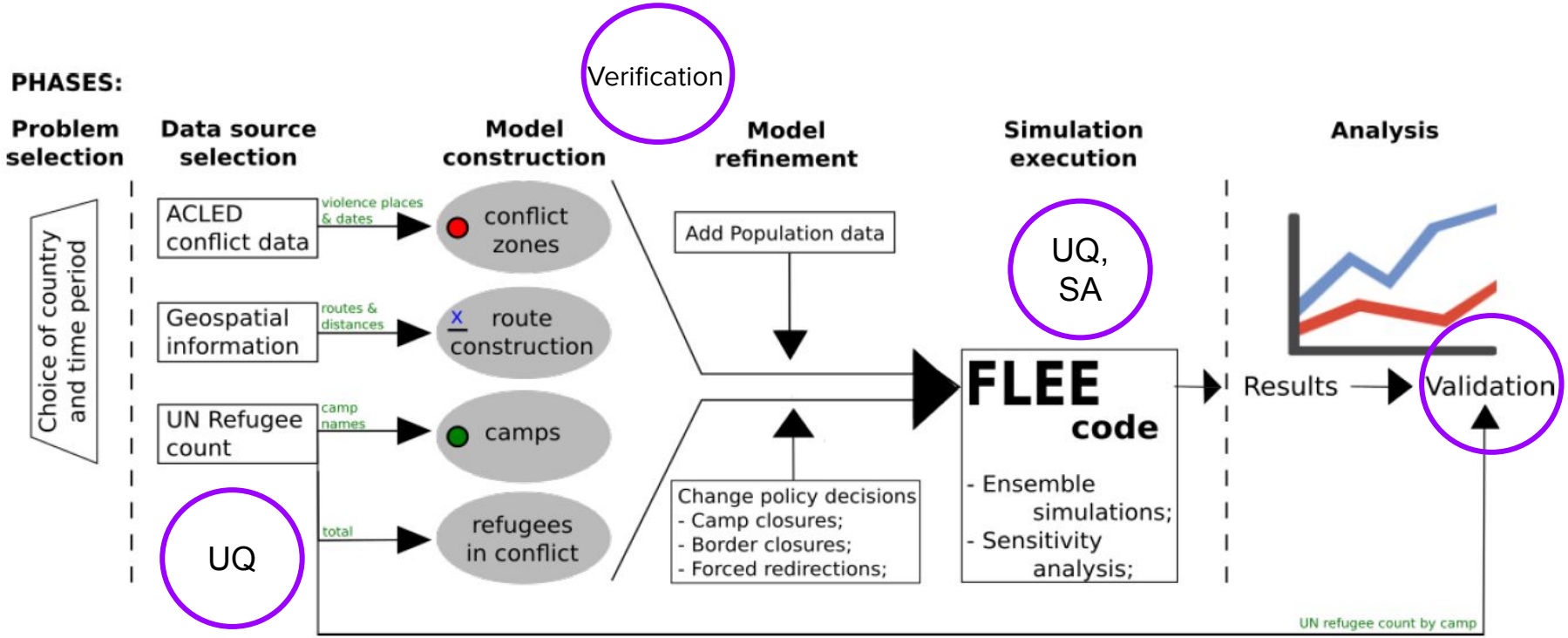
V&V patterns in FabSim3

Four prominent V&V patterns:

1. Stable Intermediate Forms (SIF):
    - monitors iterative simulation development, evaluating each intermediate step against the baseline.
2. Level of Refinement (LoR):
    - focuses asymptotic behaviour of QoIs upon changing certain model parameters, e.g. increasing grid resolution and the convergence of Sobol indices.
3. Ensemble Output Validation (EoV):
    - employs a sample testing function to compare output from different model simulations, where each simulation has its own output directory.
4. Quantities of Interest (QoI):
    - focuses on extracting a distribution of QoIs from the simulations by applying a similarity measure (e.g. Wasserstein distance).

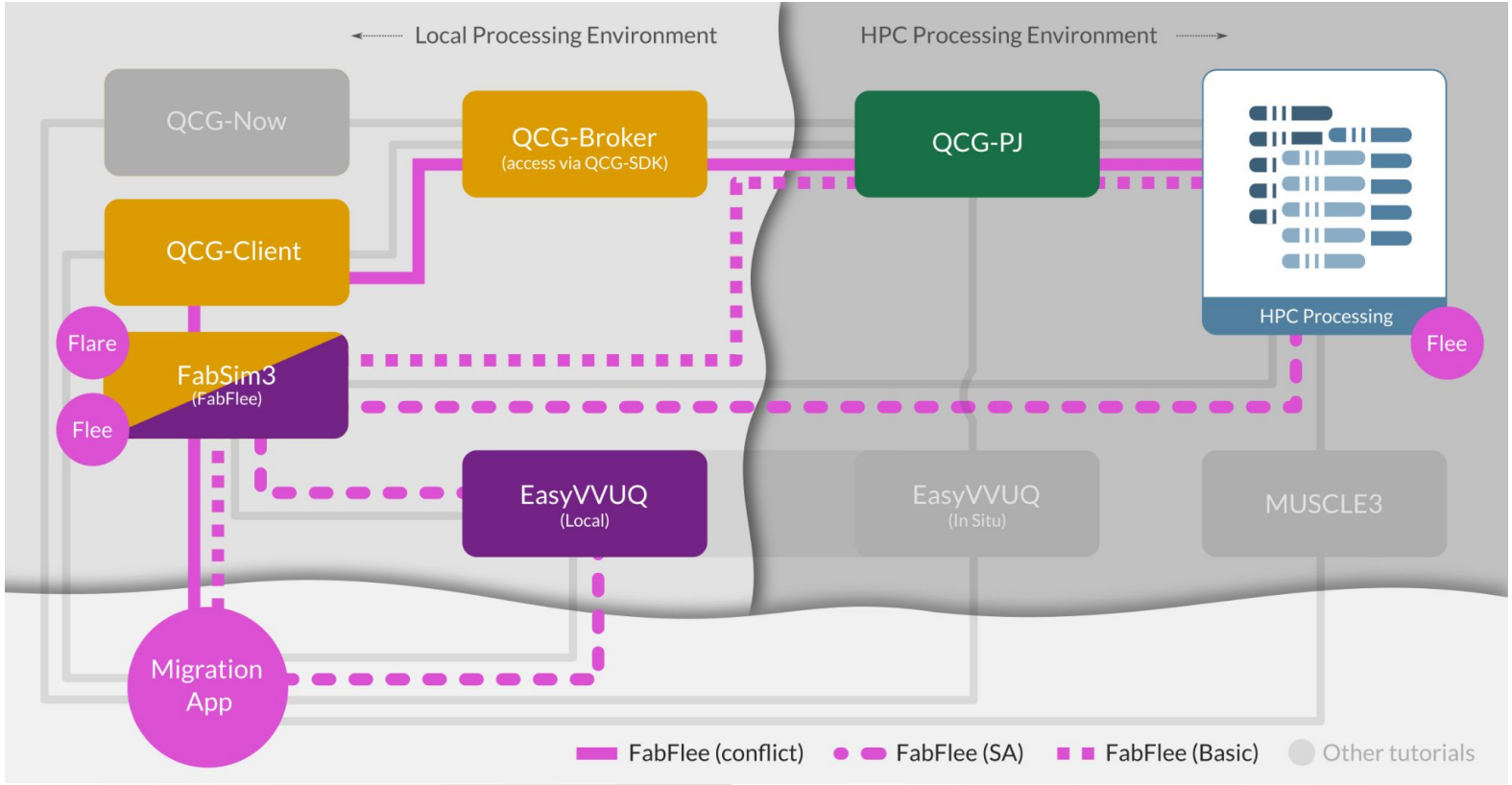
https://fabsim3.readthedocs.io/en/latest/patterns/#vvp

Example: VVUQ in Multiscale Migration Prediction



Suleimenova et al., (2021). Phil. Trans A. vol. 379 (2197).

VECMAtk Migration Tube Map



In-depth: Performing UQ analysis

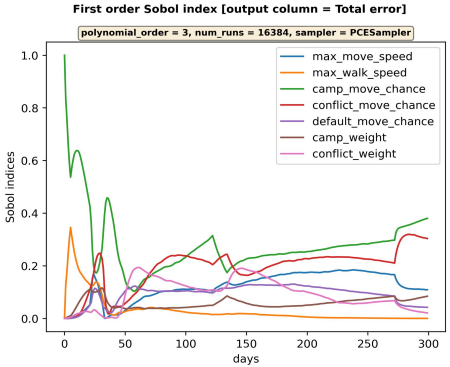
FabFlee FabSim3 plugin	• **Step 1** Definition of the space parameters.
EasyVVUQ	• **Step 2** Identification and assessment of uncertain input distributions. • **Step 3** Generation of required samples and runs.
Localhost/HPC	• **Step 4** Executions: evaluation of the application.
EasyVVUQ	• **Step 5** Aggregation of the outputs. • **Step 6** Analysis: calculation of UQ and SA measures.
FabFlee FabSim3 plugin	• **Step 7** Exploitation of the results: visualization, updating database, etc ...

Suleimenova et al., (2021). J. Comput. Sci. vol. 53 (101402).

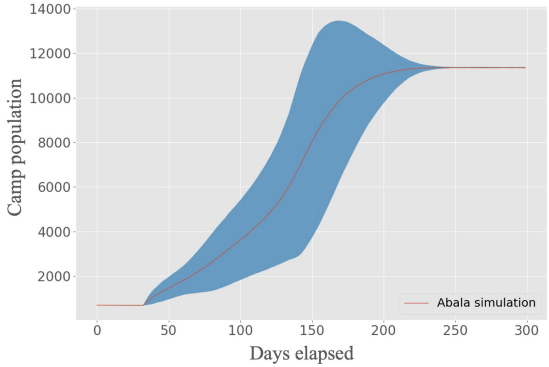
Migration VVUQ analysis

Semi-intrusive UQ analysis:

- **SA-based dimension reduction**: investigated 6-7 parameters with polynomial order of 3 using stochastic collocation and polynomial chaos expansion samplers with Sobol's method.
- **Efficient sampling:** measure the uncertainty caused by a stochastic model for evolving violent events on the output (populations in camps).



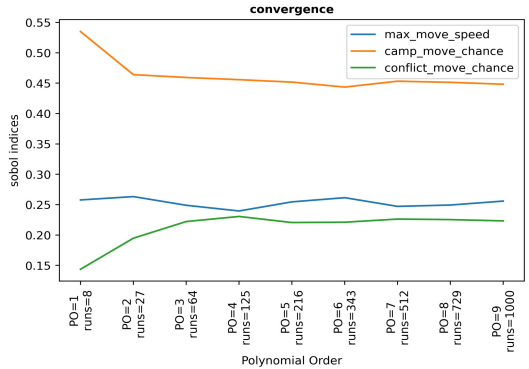
Simulations	Parameters	Mali	Burundi	South Sudan	CAR
SA iteration 1	6	0.4914	0.3832	0.4926	0.3809
SA iteration 2	7	0.4101	0.3361	0.4753	0.3257
Default values	6	0.3122	0.2571	0.5234	0.3378



Migration VVUQ analysis

Validation and Verification analysis:

- **Level of Refinement (LoR):** the Sobol indices for each parameters with different polynomial orders;
- **Ensemble Output Validation (EoV):** implemented within FabSim3 to calculate the validation metric and mean score of ensemble runs.




```
~/FabSim3/results/mali_training_vecma_1/RUNS/run_1
Validation run_1: 0.3392997798547688
~/FabSim3/results/mali_training_vecma_1/RUNS/run_7
Validation run_7: 0.2797604227928289
…
~/FabSim3/results/mali_training_vecma_1/RUNS/run_6
Validation run_6: 0.2841636513726309
Mean score: 0.29770008098431655
```

Questions?

Thank you for your attention.