CSMA 2026
17ème Colloque National en Calcul des Structures

An open source framework for the demonstration of simulation credibility, enabling qualification/certification by analysis supported by tests

Sébastien Bocquet[1], Ludovic Barrière[1], Florent Grotto[1], Stéphanie Miot[1], Clément Laboulfie[1]

[1] *IRT Saint Exupéry, CS34436, 3 Rue Tarfaya, 31400 Toulouse, sebastien.bocquet@irt-saintexupery.com*

Summary — We present VIMSEO, an open source Python framework to demonstrate simulation credibility and support critical decision-making. It is based on VV&UQ concepts, and facilitates the integration of models and methods to build analysis workflows. This framework is conceived to be collaborative and to gather several kind of users from decision makers to design office engineers and experts. Its objective is also to bridge the Machine Learning (ML) and Uncertainty Quantification (UQ) communities with the VV&UQ experts. It is generic to any kind of physics and can be easily extended with new mathematical methods and model wrappers. To support UQ activities with heavy models, this framework can scale both in terms of data management and computational ressources. Finally, it can also be specialised to specific domains by developing custom plugins.
Keywords — VV&UQ, certification by analysis supported by tests, Verification & Validation, uncertainty propagation, model credibility, virtual testing, model calibration, simulation database, HPC

1. Introduction

Reduction of development lead times and costs, management of late design changes and in-service repair has become a key target for the aeronautics industry. One enabler is the methodology of *Simulation-driven product development* or Qualification/Certification by analysis [1, 2] supported by tests. In these approaches, numerical simulations are part of the decision-making, which requires that the numerical models used are credible, particularly in the context of critical applications.

The ASME V&V 10-2019 [3, 4] guidelines describe a framework for Verification, Validation and Uncertainty Quantification (VV&UQ) that should be considered for rigorous credibility assessment. While these guidelines provide a foundation for the assessment of simulations credibility, their practical application still requires significant effort and knowledge from engineers and experts, including the definition and implementation of suitable organisation, methodologies, and infrastructure. Experts from industry or academia, engineers and also decision makers should share knowledge, capabilities, information and evidences to build the credibility of simulations. They need an operational and extensible VV&UQ framework.

2. Main requirements on a VV&UQ framework

We introduce two aspects of Certification by Analysis supported by Tests that will be main drivers regarding our proposal of a VV&UQ framework implemented in a software.

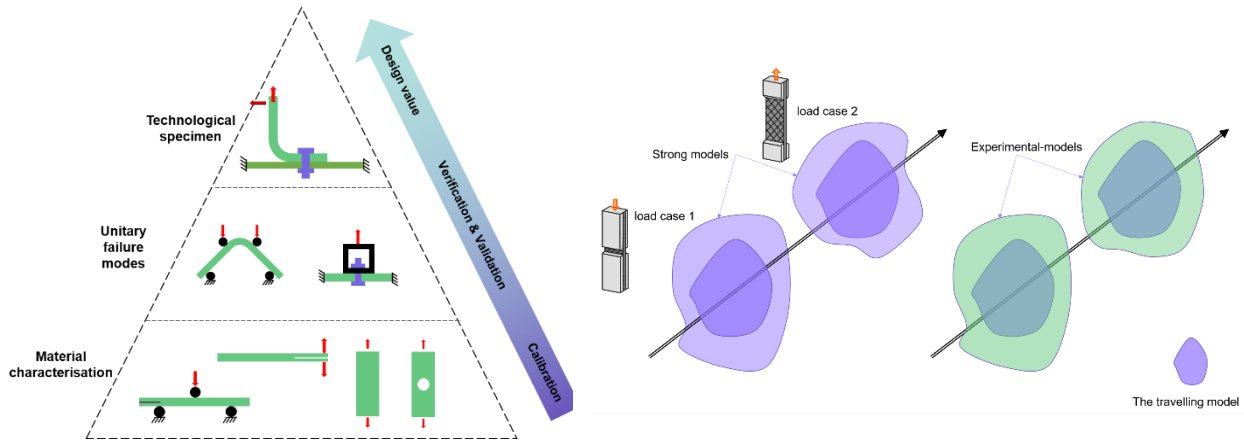

Figure 1 - A test pyramid applied to a Bolted-L-Angle specimen (left). The travelling model concept and its instantiation in strong models (right).

The first aspect concerns the modelling strategy. The building block approach, as shown in Figure 1, has supported the Certification by Analysis supported by Tests approach, and will continue during the next coming years. It involves a test pyramid in which a given material is used throughout the levels, allowing a synergy between levels (calibration is done at lower levels, validation and application of the same material is done at higher levels). The counterpart on the simulation side is that a so-called travelling model [5] is used throughout the pyramid. As shown in Figure 1, the "travelling" material must be equipped with auxiliary apparatus (grips, rollers) to be stressed through load cases (see the two experimental models colored in green). On the simulation side, the "travelling" material law must be complemented by auxiliary models and boundary conditions to enable its simulation and comparison to the experimental models (see the two strong models colored in purple). As discussed in Section 3.1, the strong model and the travelling model concepts drive the implementation of a generic model integration framework within the software.

The second aspect concerns the inherent collaborative way of using and developing a VV&UQ framework. As shown in Figure 2, several kind of users and developers are involved, which has an impact on the VV&UQ software features.

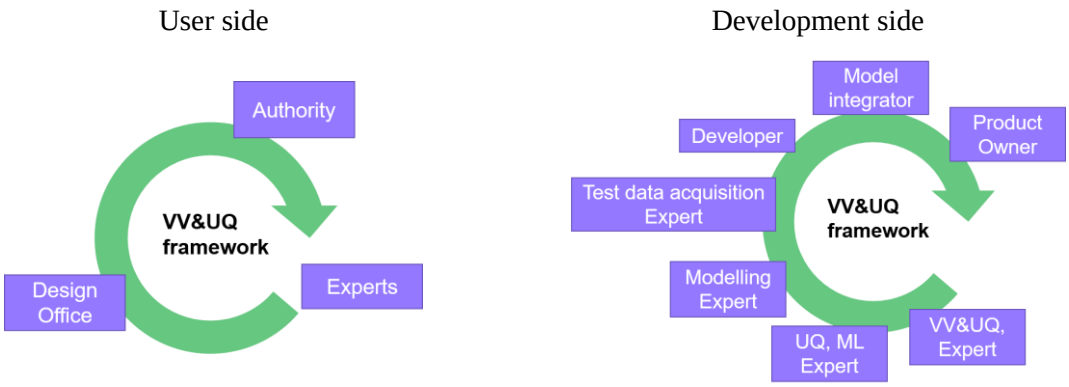

Figure 2 – A VV&UQ framework is inherently collaborative and gathers very different kind of users.

3. Main components of the VV&UQ software

3.1 Model Integration

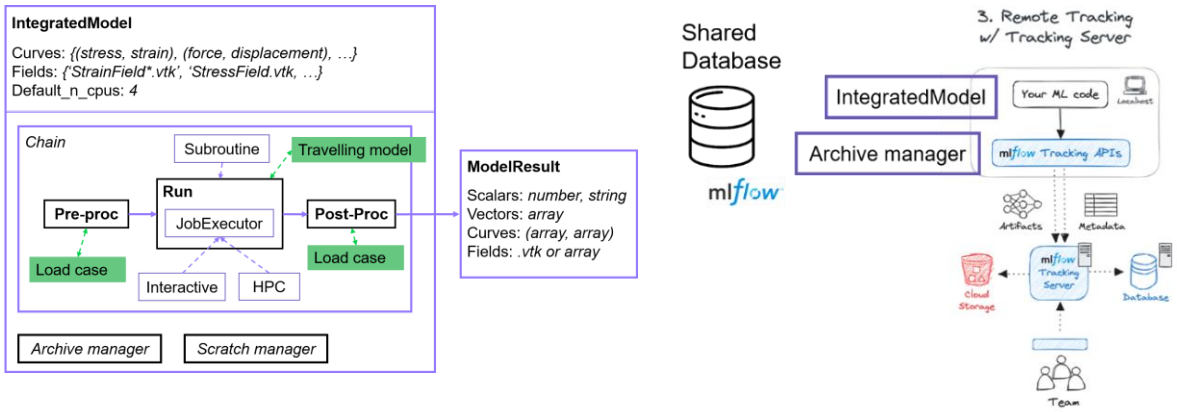

Figure 3 - Structure of an integrated model (left), and link to a shared database (right, illustration from https://mlflow.org/docs/latest/self-hosting/architecture/overview/).

As shown in Figure 3, a class IntegratedModel defines the framework for integrating models. It is composed of a chain of executable components. This approach allows the implementation of the strong model and travelling model concepts. The chain typically contains three steps: a pre-processor, a run, a post-processor:

- The run component contains the implementation of the travelling model. For Finite Element based modelling, it consists of a solver and a material law.
- The pre-processor generates the mesh, boundary conditions and auxiliary models like rollers or grips for a given load case. The pre and post-processors instantiate the travelling model for a given load case, creating a strong model that can be directly compared to an experiment.

This component-based approach allows to implement variants of a run component while using the same pre and post-processor. It is useful to implement different modelling approaches (Representative Volume Element, Finite Element using Abaqus solver, Code Aster solver, any in-house solver or script) while limiting code duplication, reducing the chance of bugs and facilitating the verification and maintainability of the software. The run component contains a JobExecutor. Two kind of JobExecutor components are implemented, one for interactive execution, and one for submitting jobs through a job scheduler on a HPC.

Another key aspect of model integration is the traceability and exploitation of the generated data. Indeed, properly tracking and being able to re-run VV&UQ workflows directly contributes to the model credibility demonstration. As shown in the IntegratedModel structure in Figure 3, a distinction is made between scratch data (raw simulation files) and archive data (useful post-processed outputs and metadata). Each data is handled by a different manager, allowing separate lifecycle and deletion policy. The post-processed data is formalised as a ModelResult object comprising metadata, scalars, vectors, curves and fields. Setting this structure is useful for providing generic utilities to visualise and compare results, between model runs or against external data, with high flexibility.

Finally, the archive data are stored in a database. The approach chosen is shown on the right of Figure 3. Rather than implementing a custom solution, we seek to reuse existing data tracking libraries. An adapter (the Archive Manager) maps the ModelResult towards a data format compatible with the data tracking library. The Open Source library *Mlflow* is chosen since its concepts can be well mapped for VV&UQ processes. This approach allows to exchange simulation data with a shared database, which is a key feature for the collaborative use of VIMSEO.

3.2 Analysis and workflows

Building bridges between communities of UQ, ML and VV&UQ can be highly beneficial to the development of VV&UQ methodologies. The idea behind the analysis and workflow framework of VIMSEO is to allow maturation and integration of UQ and ML methods into a single framework, offering the capability to define VV&UQ workflow based on these analysis building blocks. It acts as a bridge between VV&UQ, Certification by analysis experts and UQ, ML experts and mathematicians. For instance, VIMSEO has allowed the use of active-learning surrogates, Bayesian inference and sensitivity analysis within VV&UQ workflows on industrial-grade Finite Element models.

As shown in Figure 4, VIMSEO provides a frame to integrate analysis. It returns a Result object containing attributes (typically dictionaries, Pandas Dataframe, NumPy arrays) that need to be serialisable such that they can be written on disk, possibly in a readable format. Inputs and settings are Pydantic models, which allow to define arguments of a Python method in an Object-Oriented way. It is particularly suited to our case where we want extensible settings with no limitation on their type.

Figure 4 - Generic structure of an Analysis.

Similarly to model results, analysis results can be stored in a shared database by implementing adapters between analysis results and *Mlflow* concepts. VIMSEO currently implements the following analysis: code verification, solution verification, sensitivity analysis (Morris, HSIC), stochastic validation, surrogates with validation criteria and cross-validation plots, Bayesian inference, and model calibration based on a sequence of load cases with possible coupling of several load cases to identify a given set of parameters.

Then, VV&UQ workflows can be constructed based on several analysis, from which the user selects the inputs and the outputs to be exposed as a workflow step. An acyclic graph is generated based on the name of the step's inputs and outputs, as shown on the left of Figure 5.

In addition to the shared database, another key feature is a visual user interface. Indeed, making decisions from model-based analysis results requires a systematic way of visualising the data with some level of interactivity to gather experts, engineers and decision makers around the table. Choosing an appropriate library for developing these interfaces is not trivial, since it is a compromise between high performant (responsive, highly customized) framework but mastered by software specialists, and less performant framework but more accessible to scientific developers (full Python with no backend/frontend decoupling). Among the second option, *Streamlit* was found to be a good compromise and was chosen to develop VIMSEO dashboards. As shown in Figure 5, the dashboards allow to define workflows and to explore analysis results.

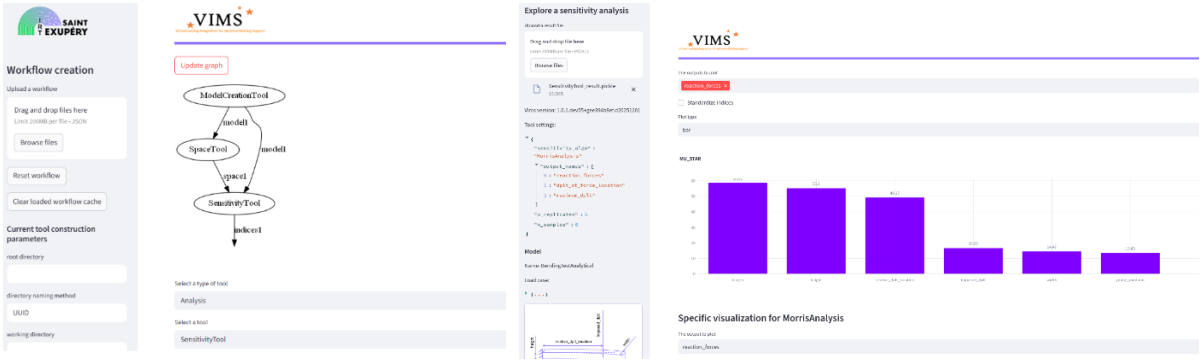

Figure 5 – The dashboard to define VV&UQ workflows (left): example of a sensitivity analysis where the user loads a model, defines a parameter space and the sensitivity analysis. On the right, the dashboard to explore the result.

It should be noted that VIMSEO relies on GEMSEO (a Multi-Disciplinary Optimisation library initiated and developed at IRT Saint Exupery [6]) for model wrapping and some analysis, including:

- Uncertainty Propagation (mostly based on Monte Carlo method),
- sensitivity analysis and associated post-processing and plots,
- surrogates, including state-of-the-art active-learning by batch methods,
- optimisation algorithms to solve the inverse problem relative to model calibration.

VIMSEO and GEMSEO also offer a two-way compatibility between the parametric models wrapped in each library.

4. Use cases

4.1 Finite-Element model of a beam

A Finite-Element model of a straight beam made of an elastic isotropic material is used to illustrate some VIMSEO capabilities. As shown in Figure 6, the runs can be explored through the *Mlflow* graphical interface. Several runs can be compared through pre-defined plots. Runs can be searched with SQL-like syntax, based on datetime, metric or metadata value.

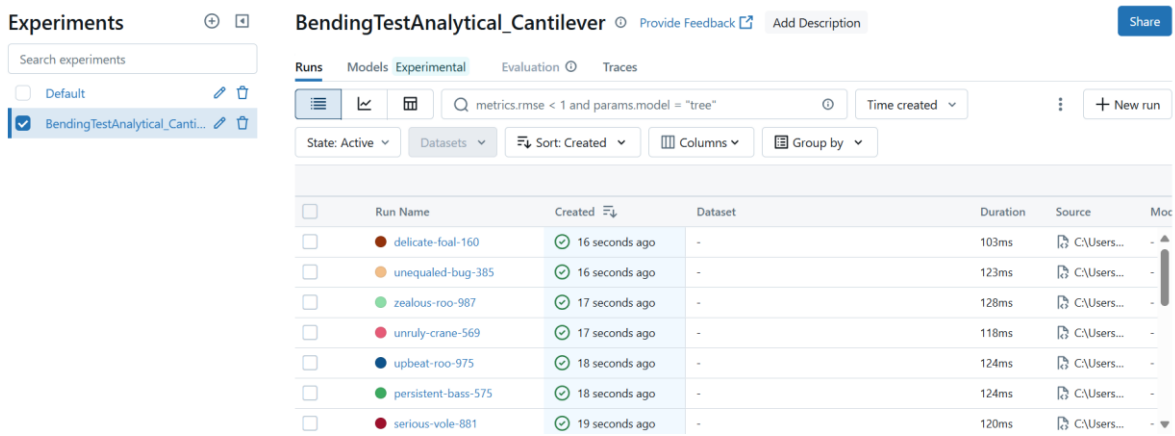

Figure 6 – Tracking of run results.

Then, we illustrate a mesh-convergence solution verification. This analysis is key when for heavy models, since using a fine mesh would lead to prohibitive computational time. For a given mesh refinement, it allows to estimate the numerical error and its uncertainty, and also find a compromise between the CPU time and the numerical error.

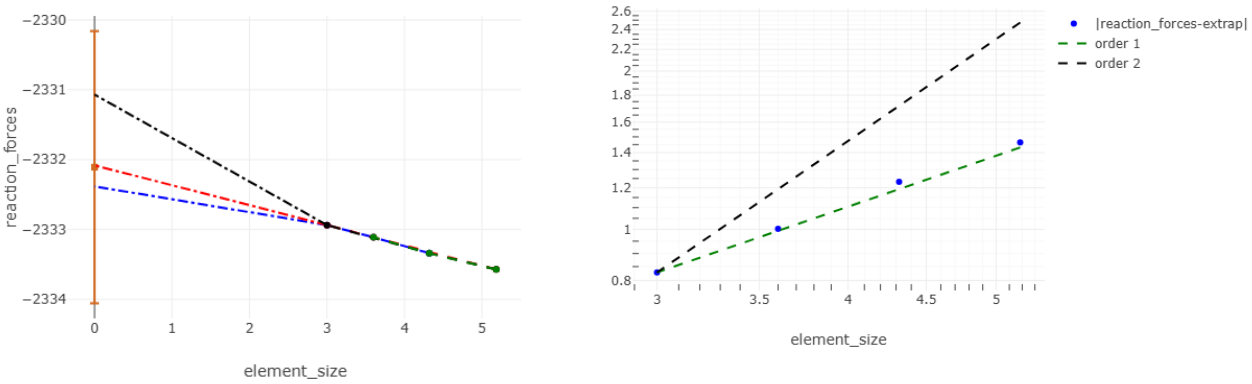

Figure 7 - Mesh-convergence solution verification: convergence trajectories (left) and convergence order (right).

4.2 Finite-Element model of an open hole composites specimen

The second use case illustrates the validation of a composites open-hole-compression model based on a non-linear material law with damage [7]. VIMSEO implements a stochastic validation, which is a good example of analysis workflow. Indeed, stochastic validation propagates on the one hand, the impact of model input uncertainties, and on the other hand that of the measured data. At both levels, these uncertainties include the variability of the material properties among other sources of uncertainties. Crude Monte Carlo method is used for uncertainty propagation, but could be replaced by more advanced methods like active-learning by batch available in GEMSEO-Mlearning [8]. Validation metrics can be further consolidated by accounting for the limited number of test repetitions from which the distributions are calibrated by using for instance Bayesian inference. As shown in Figure 8, once the experimental and simulated uncertainties are propagated, CDF (Cumulated Distribution Function) can be compared for each validation point, which provides a rich comparison between experiments and model.

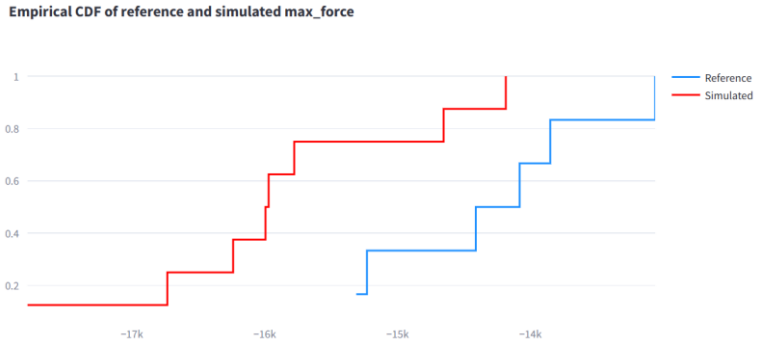

Figure 8 - Comparison of cumulated distribution functions between reference and simulation at a given validation point.

Then, stochastic metrics like the area metric can be computed. A prediction-versus-true plot is shown in Figure 9, which summarises the validation result for all the validation points. Uncertainty bars can be added to each point since the CDF are computed.

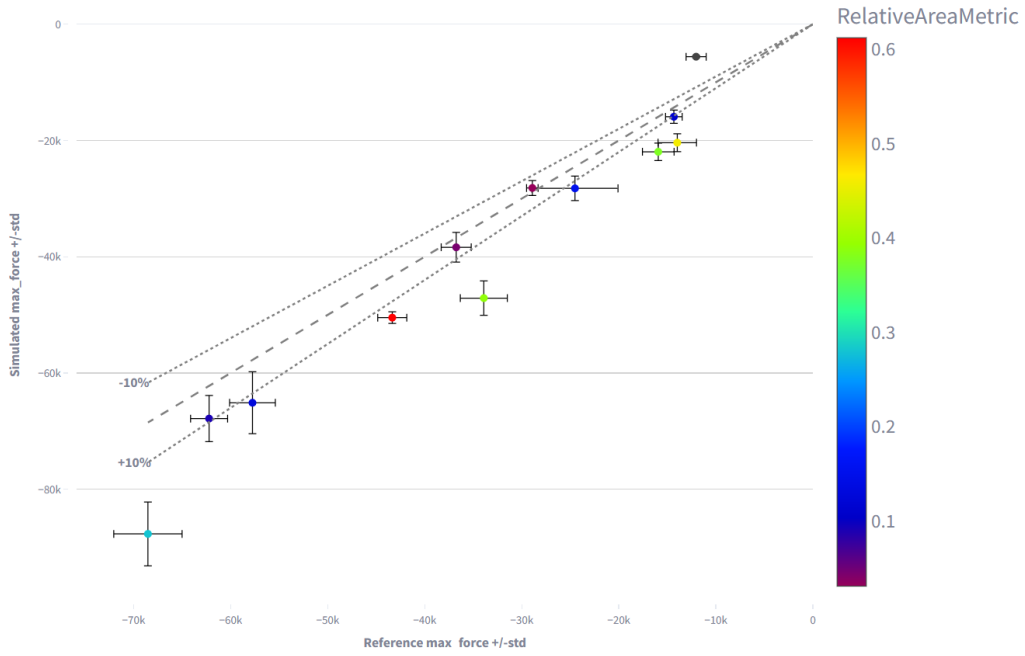

Figure 9 - Prediction versus True plot with uncertainty bars for several validation points.

5. Roadmap

VIMSEO will be released in the coming months under LGPLv3 licence, on IRT Saint Exupéry Github.

Here are some of the next functionalities we are focusing on:

- wrapping of Open Source FEA solvers, extend to other physics (CFD): Code Aster, OpenFoam, SU2 etc…),
- HPC compatibility for other job schedulers, link towards Barcelona Super Computing (BSC) for example,
- handling fields and meshes and associated metrics,
- traceability of dataset to analysis,
- workflow orchestration with definition from graphical user interface and live monitoring, using existing workflow orchestration libraries,
- run complete VV&UQ workflows on use cases representative of industrial applications.

6. References

[1] Certification by Analysis https://ntrs.nasa.gov/api/citations/20210015404/downloads/NASA-CR-20210015404%20updated.pdf

[2] Wim Doeland. Modelling & Simulation – CS-25 Structural Certification Specifications. Technical Report, EASA, 07 2020. URL: https://www.easa.europa.eu/en/document-library/product-certification-consultations/proposed-certification-memorandum-modelling#group-easa-downloads.

[3] VV10 - Standard for Verification and Validation in Computational Solid Mechanics https://www.asme.org/codes-standards/find-codes-standards/standard-for-verification-and-validation-in-computational-solid-mechanics

[4] Christopher J Freitas. Standards and methods for verification, validation, and uncertainty assessments in modeling and simulation. *Journal of Verification, Validation and Uncertainty Quantification*, 5(2):021001, 2020.

[5] Vicente Romero. Comparison of several model validation conceptions against a "real space" end-to-end approach. *SAE International*, 4:396–420, 2011.
URL: https://www.jstor.org/stable/10.2307/26273778.

[6] François Gallard, Charlie Vanaret, Damien Guénot, Vincent Gachelin, Rémi Lafage, Benoît Pauwels, Pierre-Jean Barjhoux, and Anne Gazaix. Gems: a python library for automation of multidisciplinary design optimization process generation. In 01 2018. doi:10.2514/6.2018-0657. https://gemseo.readthedocs.io/en/stable/index.html

[7] Laurin, N. Carrère, and J.-F. Maire. A multiscale progressive failure approach for composite laminates based on thermodynamical viscoelastic and damage models. Composites Part A: Applied Science and Manufacturing, 38(1):198–209, 2007s

[8] https://gitlab.com/gemseo/dev/gemseo-mlearning.git