pyvale: A virtual engineering laboratory for simulating sensor measurement uncertainties

Lloyd Fletcher[1]
TODO[1]
[1]UK Atomic Energy Authority, Fusion Technology Facilities, Rotherham, UK
[2]UK Atomic Energy Authority, Culham Campus, Culham, UK

Abstract

'pyvale' is a simulation package... virtual engineering laboratory
Keywords:

Metadata

Nr.	Code metadata description	Metadata
C1	Current code version	2025.X.X
C2	Permanent link to code/repository used for this code version	`https://github.com/Computer-Aided-Validation-Laboratory/pyvale`
C3	Permanent link to Reproducible Capsule	For example: TODO
C4	Legal Code License	MIT License
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	python
C7	Compilation requirements, operating environments & dependencies	Cross platform distributed on the python package index, pypi: 'pip install pyvale'
C8	If available Link to developer documentation/manual	`https://computer-aided-validation-laboratory.github.io/pyvale/index.html`
C9	Support email for questions	lloyd.fletcher@ukaea.uk

Table 1: Code metadata (mandatory)

1 Motivation and significance

Qualification of fusion technology is reliant on simulations to predict the performance of components in extreme (e.g., thermal and electromagnetic) and untestable (e.g., fusion neutron fluxes) environments. Enabling the use of simulations for risk-informed decision making requires that they are validated over testable domains to reduce uncertainty when extrapolating into irradiated conditions. The cost of performing large-scale validation tests on a complex components such as a breeder blankets will be on the order of £M's. Therefore, significant cost and risk reduction can be achieved by maximising the information obtained from an optimised set of targeted experiments.

A key parameter of validation experiments is the deployment of sensor arrays to measure the components response. There are currently no software tools available that can simulate and optimise

the placement of diverse arrays of sensors for multi-physics conditions with realistic constraints (e.g., cost, reliability, and accuracy). Such a tool would have immediate benefits for reducing costs of the experimental programme required to qualify fusion components such as the breeder blankets and divertors [REFs].

Simulation validation is a fundamental problem of scientific and engineering computing [REFS] and it is particularly challenging for the multi-physics environments components are subjected to in fusion reactors [REFs]. Experimental validation

We are developing this software in parallel to a research programme focusing on .

We envisage multiple applications of 'pyvale': testing validation metrics with known ground truth while modelling systematic and random measurement errors.

Here we describe the core sensor simulation engine of 'pyvale'

TODO [1]

In this section, we want you to introduce the scientific background and the motivation for developing the software.

- *Explain why the software is important and describe the exact (scientific) problem(s) it solves.*
- *Indicate in what way the software has contributed (or will contribute in the future) to the process of scientific discovery; if available, please cite a research paper using the software.*
- *Provide a description of the experimental setting. (How does the user use the software?)*
- *Introduce related work in literature (cite or list algorithms used, other software etc.).*

2 Software description

The pyvale sensor simulation module provides a physics agnostic virtual laboratory allowing the user to deploy synthetic sensors on an existing simulation.

The user workflow is as follows: 1) load multi-physics simulation data, 2) build virtual sensor arrays, 3) run simulated experiments, and 4) visualise and analyse the results.

pyvale supports mesh-based and mesh-free input physics simulations

2.1 Software architecture

- Flow chart figure showing the pyvale sensor simulation module

2.2 Software functionalities

2.3 Sample code snippets analysis (optional)

3 Illustrative examples

Provide at least one illustrative example to demonstrate the major functions of your software/code.

4 Impact

This is the main section of the article and reviewers will weight it appropriately. Please indicate:

- *Any new research questions that can be pursued as a result of your software.*
- *In what way, and to what extent, your software improves the pursuit of existing research questions.*
- *Any ways in which your software has changed the daily practice of its users.*
- *How widespread the use of the software is within and outside the intended user group (downloads, number of users if your software is a service, citable publications, etc.).*

- *How the software is being used in commercial settings and/or how it has led to the creation of spin-off companies.*

Please note that points 1 and 2 are best demonstrated by references to citable publications.

Sensor simulation has the potential to impact a wide range of engineering fields. 'pyvale' is a new
We have specifically designed pyvale
Future modules for pyvale will include

5 Conclusions

Acknowledgements

Dr Lloyd Fletcher acknowledges support from UKRI through the Future Leaders Fellowship scheme (grant).

CRediT Authorship Statement

TODO

References

[1] Derek Gaston, Chris Newman, Glen Hansen, and Damien Lebrun-Grandié. MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, October 2009.