Where Experimental Mechanics and Supercomputing Meet: Uncertainty Quantification for Fusion Validation

L. Fletcher[1a], M. Atkinson[1], A. Marsh[1], A. Tayeb[1] and C. Hamelin[1]

[1]United Kingdom Atomic Energy Authority, Culham Science Centre, Abingdon, OX14 3DB, United Kingdom

[a]lloyd.fletcher@ukaea.uk

Abstract. Qualifying components for service inside the extreme environment of a fusion reactor is a significant engineering challenge. It is possible for us to test large scale components under combined mechanical, thermal and electromagnetic loads, but it will not be possible for us to include fusion spectrum neutron irradiation. This means that our initial component lifetime predictions will be made with simulations. Therefore, our simulations will require experimental validation over the testable mechanical, thermal and electromagnetic domains to reduce uncertainty when extrapolating into irradiated conditions. Large-scale experiments for simulation validation are expensive in terms of the time and resources required so optimising the information gained from these experiments has the potential for significant efficiency gain. In this work we discuss the development of a software toolbox called pyvale (the python validation engine). This toolbox is for simulating sensors (e.g. thermocouples, strain gauges, infra-red thermography and digital image correlation systems) and optimising engineering experiments. We then discuss the future application of this toolbox on supercomputing clusters to run large parallel sweeps of experimental parameters to quantify uncertainties and optimise information gained from fusion component validation experiments.

Possible Sessions

18. Nuclear Applications: Fusion, 14. Model Validation

Introduction & Motivation

Fusion is an attractive source of clean energy but there are significant engineering challenges that must be overcome to make sustainable fusion power a reality. One of these challenges is assessing component structural integrity under combined thermal and electromagnetic loads with neutron irradiation damage. While it is possible for us to experimentally test components under mechanical, thermal and electromagnetic loads before service in a fusion reactor, it will not be possible for us to test large components under fusion spectrum neutron irradiation. For this reason, we will require simulations to qualify components prior to service inside a reactor. Furthermore, our simulations will need to be validated over as much as the operational domain as possible to minimise uncertainty when making predictions under fusion neutron irradiation. Given the cost in time and resources required to undertake large component qualification experiments optimising the information gained from an experiment has the potential for significant savings.

While it is not possible to perform thousands to millions of meter scale component experiments it is possible to perform this many simulations on the latest generation of supercomputing clusters that have thousands to tens of thousands of chips with hundreds of thousands of cores. To bridge this gap between experiments and simulations we are building a software engine that can generate realistic sensor data (with uncertainties) from a physics simulation. Here we take inspiration from the digital image correlation community who have developed and used synthetic image deformation as a method for assessing uncertainty (systematic and random) for image-based deformation measurements [1,2]. We extend this idea to the simulation of a wide array of thermal and mechanical sensors such as thermo-couples, strain gauges, pyrometers, infra-red thermography, and digital image correlation systems. We also specify an abstract interface allowing users to define their own sensors and associated uncertainty models with the idea that the platform could be extended to include thermal-hydraulic measurement systems or neutronics sensors in the future.

The software engine we are developing for creating synthetic experimental data from simulations has a wide variety of applications, including: 1) uncertainty quantification for a given array of sensors, 2) experiment and sensor placement optimisation; 3) providing ground-truth data for benchmarking and developing validation metrics, 4) providing a mechanism for generating large simulated experimental datasets to train machine learning algorithms; and 5) testing the predictive capability of digital shadows/twins. In this work we discuss current progress on pyvale (python validation engine) and show a minimal working demonstration applying simulated thermocouples to a thermal simulation of a fusion heatsink component.

The pyvale Sensor Model

We based the pyvale sensor model on a collection of sensors of the same type called a SensorArray as vectorised operations reduce computation time, and it will make it easier to implement camera sensors in the future. A pyvale SensorArray is an abstract base class that implements the following relationship: measurement = truth + systematic error + random error. The first implementation of a SensorArray in pyvale is a ThermocoupleArray. In general, the truth is taken directly from the simulation by using the element shape functions for spatial interpolation and linearly interpolating between simulation time steps to match the userdefined sensor sampling frequency. The default systematic error function is randomly sampled from a uniform probability distribution where the user specifies the upper and lower bounds. This systematic error function is sampled once and creates an offset for the duration of the simulated sensor trace. The default random error function is a normal probability distribution where the user specifies the standard deviation. The random error function is sampled for every measurement. We also note that the current sensor interface allows users to specify custom systematic and random error functions by passing a callable python object or 'None' which removes the calculation of either error. To create a ThermocoupleArray the user specifies a list of locations and an optional set of time steps to take measurements. If no time steps are specified, then the sensor output frequency is assumed to match the simulation timestep. We have made pyvale open-source and the code is available on GitHub: https://github.com/Applied-Materials-Technology/pyvale.

Prototype Demonstration of pyvale on a Fusion Component Simulation

Here we use the thermal simulation of a divertor armour heatsink that is a multi-material system consisting of an external tungsten armour block with a copper-chromium-zirconium pipe and pure copper interlayer. The top of the tungsten block is subjected to a spatially uniform time varying heat flux with a steady-state peak of 10 $MW.m^{-2}$. The internal surface of the pipe is cooled with water that is assumed to have a heat transfer coefficient based on the Sieder-Tate relationship. The model is solved using the Multi-Physics Object Oriented Simulation Environment (MOOSE) [3]. The full input script for the model can be found in the GitHub repository.

The simulation output is then used with pyvale to create simulated thermocouple traces. Here we simulate four thermocouples equally spaced on the side of the component with a sampling frequency of 2 Hz. A visualisation of the thermocouple locations is shown in Fig. 1. For this case we have assumed the bounds of the uniform systematic error function are [-10,10] K and the standard deviation of the random error function is 10 K. Note that these values are exaggerated compared to true thermocouple uncertainties for testing.

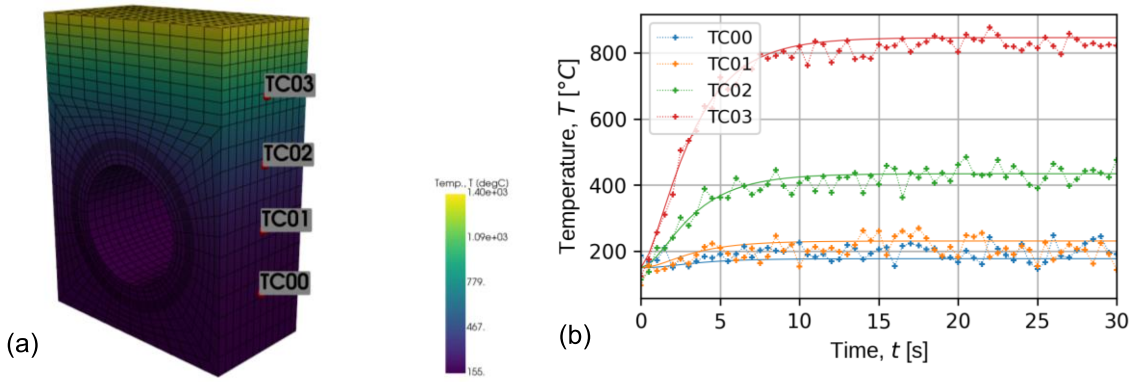

Figure 1: (a) Visualisation of the simulation temperature field for the heatsink component at t=30s showing the locations of the thermocouples (denoted TC). (b) Simulated temperature traces for the thermocouple locations, the solid lines show the simulation output and the markers and dashed lines show the thermocouple traces.

Conclusion & Future Work

In this work we have developed the first prototype of an experiment and sensor simulation engine called pyvale using the example of thermocouples applied to a fusion heatsink component under thermal loading. Our motivation for developing pyvale is to provide us a set of tools that will allow us to design and optimise validation experiments using large parallel simulation sweeps on supercomputing clusters. For future work we will implement a wider range of thermal and mechanical sensor models with increased fidelity of systematic and random error functions. A key focus will be to allow users to build their own custom systematic error function from a library of common sources of error such as spatial/temporal averaging, digitisation error, calibration error and positioning error. We are also developing additional modules for pyvale that will connect the data generated by our sensor models to sensor placement optimisation algorithms, validation metric calculations, and finite element updating procedures for model calibration. At the conference we will demonstrate the latest version of pyvale including an application on the CSD3 cluster at Cambridge.

Acknowledgments: This work has been funded by the EPSRC Energy Programme [grant number EP/W006839/1].

References

[1] M. Rossi, P. Lava, F. Pierron, D. Debruyne, M. Sasso, Effect of DIC Spatial Resolution, Noise and Interpolation Error on Identification Results with the VFM, Strain 51 (2015) 206–222. https://doi.org/10.1111/str.12134.

[2] D. Rohe, E. Jones, Generation of Synthetic Digital Image Correlation Images Using the Open-Source Blender Software, Exp. Tech. 46 (2021). https://doi.org/10.1007/s40799-021-00491-z.

[3] C.J. Permann, D.R. Gaston, D. Andrš, R.W. Carlsen, F. Kong, A.D. Lindsay, J.M. Miller, J.W. Peterson, A.E. Slaughter, R.H. Stogner, R.C. Martineau, MOOSE: Enabling massively parallel multiphysics simulation, SoftwareX 11 (2020) 100430. https://doi.org/10.1016/j.softx.2020.100430.