University College Dublin

School of Mechanical and Materials Engineering

Using the autoprogressive algorithm with OpenFOAM and Python to learn mechanical constitutive models

Simon Rodriguez, Mert Celikin, Pádraig Cunningham, Philip Cardiff

simon.rodriguezluzardo@ucdconnect.ie, mert.celikin@ucd.ie, padraig.cunningham@ucd.ie, philip.cardiff@ucd.ie

WHAT IS THIS PROJECT ABOUT?

This project aims to increase the accuracy of elastoplastic numerical simulations by replacing the typical theoretical material models with machine-learning models trained directly from mechanical tests or microstructural simulation results (see figure 1).

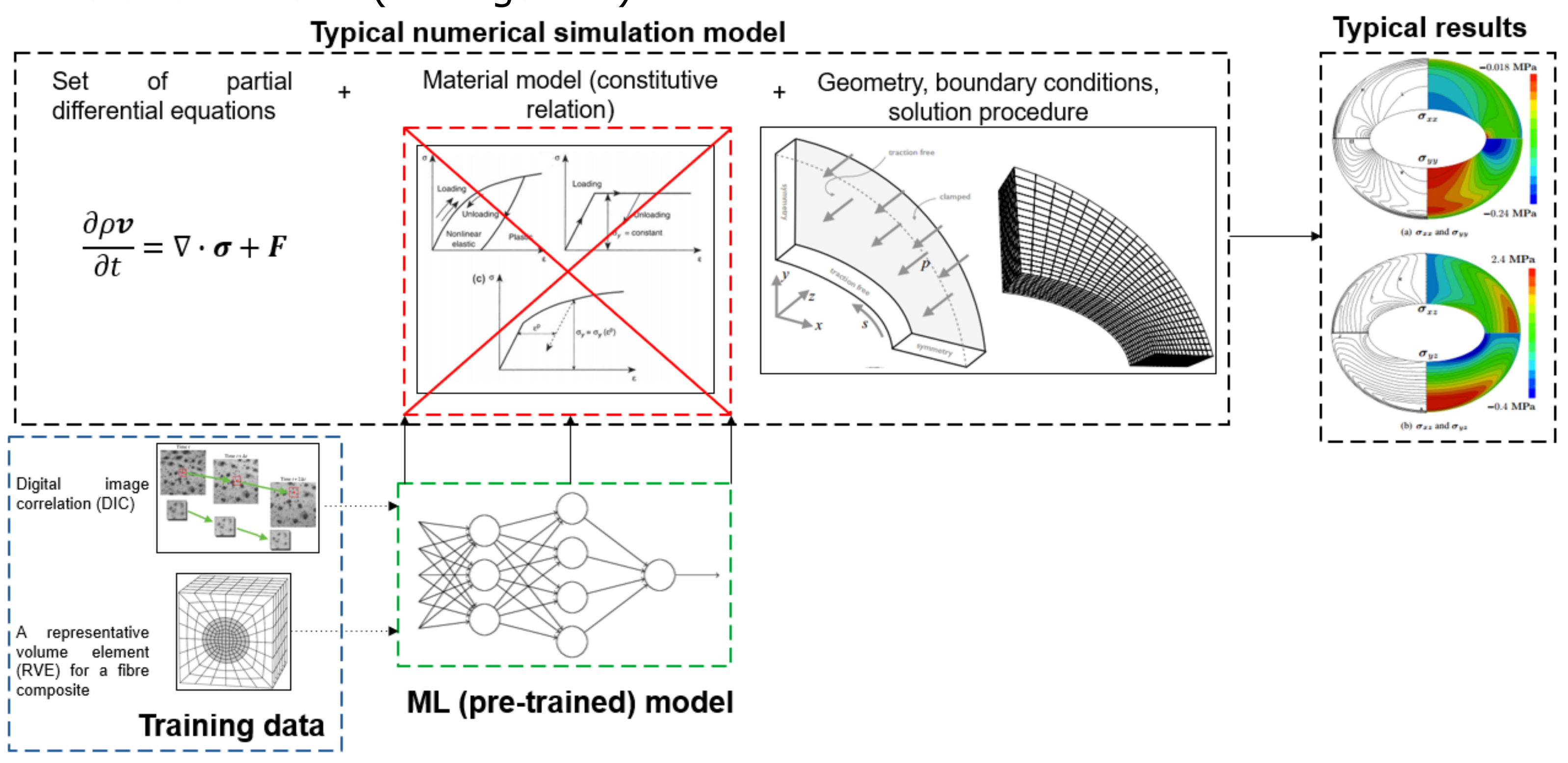
Typical numerical simulation model
Set of partial differential equations
+
Material model (constitutive relation)
+
Geometry, boundary conditions, solution procedure
$\frac{\partial \rho \boldsymbol{v}}{\partial t} = \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{F}$
Loading
Unloading
Nonlinear elastic
Plastic
constant
traction free
clamped
symmetry
Typical results
−0.018 MPa
−0.24 MPa
2.4 MPa
−0.4 MPa
Digital image correlation (DIC)
A representative volume element (RVE) for a fibre composite
Training data
ML (pre-trained) model

Figure 1. The typical solid mechanics simulation framework (black dashed rectangle, left) is modified by replacing the constitutive law (red square) with a machine-learning model (green rectangle) directly trained with mechanical tests o microstructures simulation (blue square). This is expected to enhance the simulation accuracy [1].

The difficulty with the approach is that the supervised algorithm requires a training set composed of known (measured) strain-stress pairs. However, the stresses are not directly measurable in the laboratory, although the global forces are measurable (for example, at the gripping attachments).

THE AUTOPROGRESSIVE ALGORITHM

The autoprogressive algorithm [2] is an inverse method that guesses an initial machine-learning-based material model and uses it to generate new strain-stress pairs whose behaviour approximates the measured one. This new data is added to the training set and the model is retrained, improving its accuracy. The process is repeated until convergence is achieved.

The new strains and stresses are generated by performing two simulations:
a) *Load-based simulation*: The measured loads are applied as boundary conditions.
b) *Displacement-based simulation*: The displacements at the measured locations are included as boundary conditions.

This project develops a framework for implementing the autoprogressive algorithm in OpenFOAM. This is achieved by constructing a Python routine which:

- Sets up and runs the load-based simulation.
- Saves the required OpenFOAM Field (stress) as Python NumPy readable files.
- Sets up and runs the displacement-based simulation.
- Repeats step b, saving the strain Field this time.
- Reads the stress from step b) and the strain from step d), adds them to the training set and retrains the machine learning model.
- Repeats the process a) to e) until convergence is achieved.

PYTHON/OPENFOAM TYPE CONVERSIONS

The machine-learning models are developed in Python and the mechanics solver is OpenFOAM which is a C++ code, however, there is no standard for running Python codes in OpenFOAM/C++. This project introduced a computational framework for this. It is known as pythonPal4foam [3] (see figure 2). More information in https://pythonpal4foam.github.io/

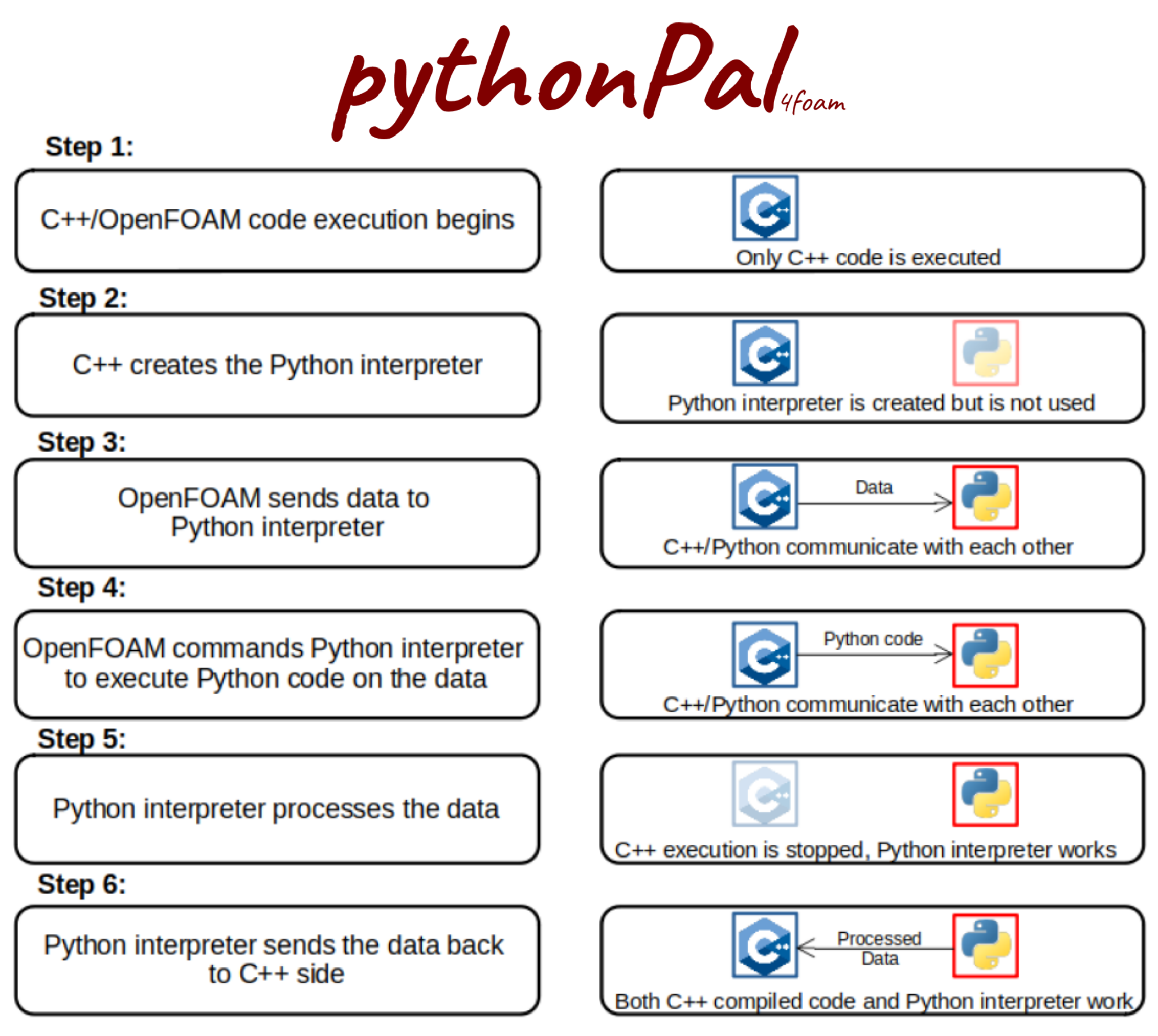
pythonPal4foam
Step 1:
C++/OpenFOAM code execution begins
Only C++ code is executed
Step 2:
C++ creates the Python interpreter
Python interpreter is created but is not used
Step 3:
OpenFOAM sends data to Python interpreter
Data
C++/Python communicate with each other
Step 4:
OpenFOAM commands Python interpreter to execute Python code on the data
Python code
C++/Python communicate with each other
Step 5:
Python interpreter processes the data
C++ execution is stopped, Python interpreter works
Step 6:
Python interpreter sends the data back to C++ side
Processed Data
Both C++ compiled code and Python interpreter work

Figure 2. Schematic depiction of the pythonPal4Foam. In the background, the pybind11 Python library is used to embed a Python interpreter in OpenFOAM where the machine-learning calculations are performed. Python/OpenFOAM communication is provided by pybind11. [3].

INITIAL RESULTS

The initial tests for the autoprogressive algorithm presented in this work solve a reference problem where a linear elastic steel plate with a hole fixed to a wall is subjected to a load at one of the boundaries, as shown in Figure 3.

The initial guess for the material model is given by linear regression, implemented with scikit-learn, which is trained on strain/stress pairs generated with the linear Hookean law using material properties considerably different from the expected ones (see Table 1). The initial results show that the assumed material model learns the experimental behaviour (see Figures 4-5).

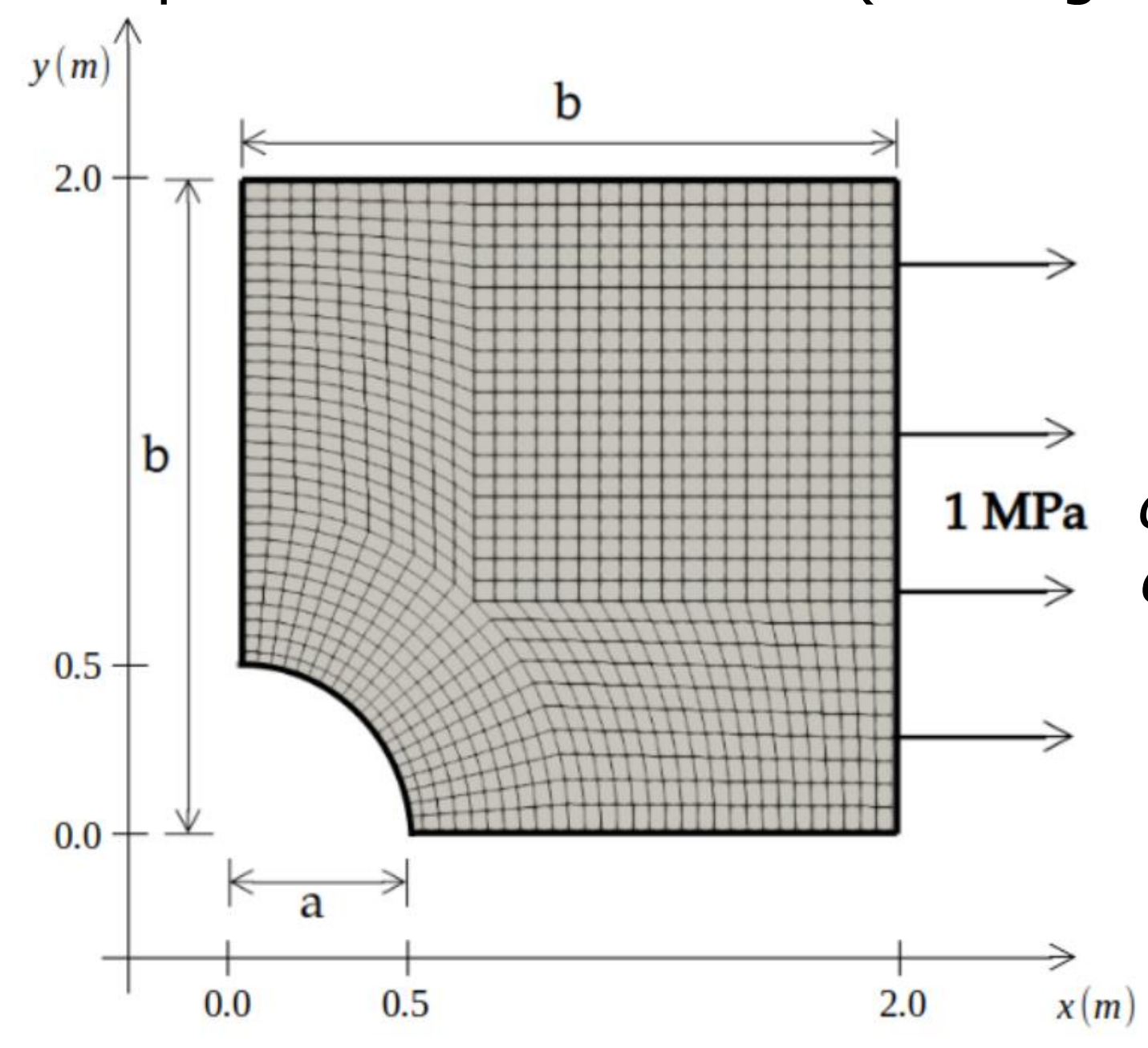

Figure 3. Geometry of the spatial computational domain for the flat plate with a circular hole (a = 0.5 m, b = 2 m, t_x = 1MPa) [3].

Table 1: Material properties.

	Young's modulus (GPa)	Poisson's ratio
Real	200	0.3
Initial guess	100	0.1

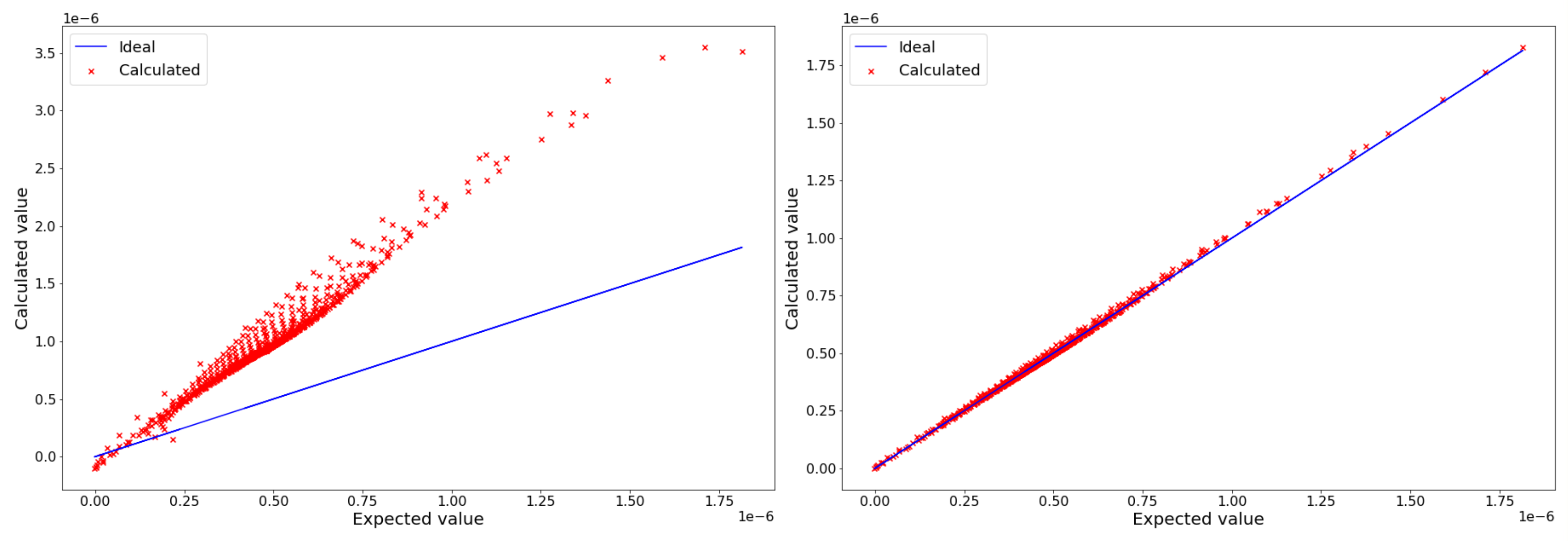

Figure 4: Expected vs calculated ε_{xx} results for the linear regression-based material model. Each red cross corresponds to a material point in the mesh, whereas the blue line shows the ideal behaviour. The final model (right) considerably improves the initial estimates (left).

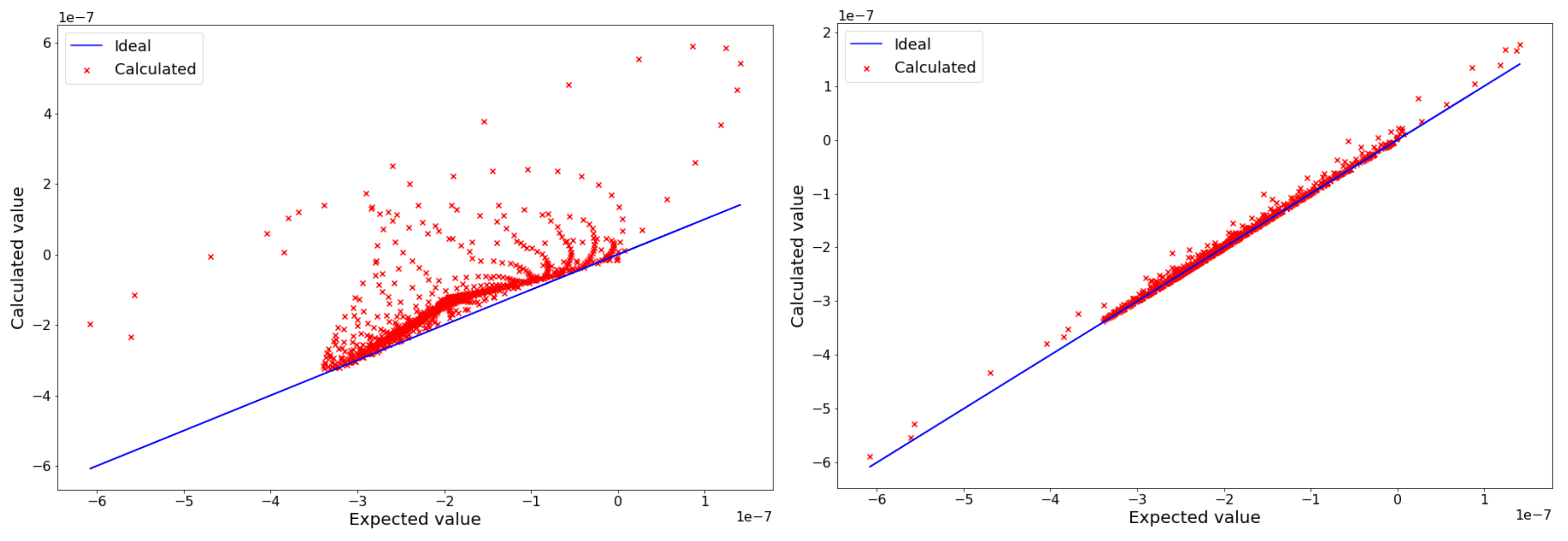

Figure 5: Expected vs calculated ε_{yy} results for the linear regression-based material model. As for ε_{xx}, the final model (right) considerably improves the initial estimates (left).

NEXT STEPS

- Replace the linear regression model with a neural network.
- Evaluate the performance of the framework on an elastoplastic deformation.

CONCLUSION

- This project proposes an alternative solid mechanics simulation framework that takes advantage of machine-learning methods to improve accuracy.
- The mechanical solver is OpenFOAM and the machine-learning methods are Python-based and the interoperability between both languages is provided by an in-house tool called pythonPal4foam.
- The initial tests performed in an elastic case show the promise of the autoprogressive algorithm as a mechanism to learn elastoplastic behaviour from experiments.
- The next steps involve the use of neural network and the inclusion of elastoplastic behaviour.

References:

[1] Simón Rodríguez and P. Cardiff, "INTEGRATING PYTHON CODES INTO OPENFOAM SOLVERS USING AN EMBEDDED PYTHON INTERPRETER VIA PYBIND11," 2022, doi: 10.13140/RG.2.2.31612.62087.
[2] J. Ghaboussi, D. A. W. Pecknold, and X. Wu, Nonlinear computational solid mechanics. Boca Raton: CRC Press, Taylor & Francis Group, 2017.
[3] S. A. Rodriguez Luzardo and P. Cardiff, "A General Approach for Running Python Codes in OpenFOAM Using an Embedded PYBIND11 Python Interpreter," OpenFOAM J, vol. 2, pp. 166–182, Dec. 2022, doi: 10.51560/ofj.v2.79.

IRISH RESEARCH COUNCIL
An Chomhairle um Thaighde in Éirinn

UCD Centre for Mechanics

UCD STEM Early Career Researchers Symposium
13 June 2023

ICHEC Irish Centre for High-end Computing

#LoveIrish Research