The Application of Machine Learning in Micromechanical Modelling

Student Name: William Brennan

ID Number: 17237645

Supervisor: Prof. Noel O'Dowd

Second Reader: Prof. Paul Weaver

Contributor: Dr. Alison O'Connor

Degree: Master of Engineering in Mechanical Engineering

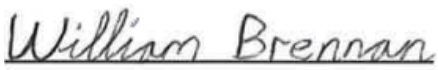
Master of Engineering Thesis submitted to the University of Limerick,

March 2022

School of Engineering

Declaration

I declare that this is my work and that all contributions from other persons have been appropriately identified and acknowledged.



William Brennan

Abstract

In engineering applications, it is often necessary to predict material behaviour by implementing finite element (FE) methods to minimise time, cost and the wasting of resources. In recent years significant interest has developed in accurately modelling the effect of ductile damage to improve simulated results of a tensile test. It is challenging to include ductile damage in a finite element analysis (FEA) due to the complexity of the Gurson-Tvergaard-Needleman (GTN) yield function, which requires a number of material related parameters. Therefore, the purpose of this work is to create a framework that can identify the optimum GTN parameters, thereby reducing the time and computational expense of conducting these simulations. The framework combines Abaqus FEA software with a machine learning (ML) algorithm using the Python programming language to efficiently identify the optimum combination of material parameters.

A python script was created to automatically generate Abaqus models representing an experimental tensile test conducted by Meade *et al.*, (2021). The python script automatically extracts the simulated load-displacement results outputting the data to a comma separated value (csv) file. Simulated results are compared to experimental data using the mean absolute percentage error (MAPE) to identify whether the load-displacement data are similar. A Bayesian Optimisation (BO) ML approach was incorporated to improve the parameter reselection based on previous MAPE values. To determine the accuracy of the BO approach, it was compared to a fractional factorial design (FFD) method in terms of the time taken and number of simulations required to identify the optimum GTN parameters. Additionally, the repeatability of the BO algorithm was investigated by running the algorithm multiple times and assessing the variation in results. The versatility of the algorithm was examined by applying the algorithm to additional experimental data.

The implementation of the BO algorithm successfully identified the optimum GTN parameters for an ex-service (63,000 hrs) P91 steel tested at 500 °C as q_1 =0.90, q_2= 0.90, q3 = 2.56, ε_N = 0.25, s_N = 0.20, f_N = 0.03, f =0.0013 and the extrapolated true stress-plastic strain data slope = 417.2 MPa, resulting in a local minimum MAPE of 2.10 %. The BO minimum was slightly higher than the FFD minimum of 2.05% and significant variation between the identified optimum parameters was evident, indicating that multiple solutions may exist for the problem. Despite the low MAPE achieved by the BO, it did not converge on a final MAPE value however, the algorithm showed no variation in the identified optimum parameters and the minimum MAPE value in the repeated tests, providing the user with confidence in reproducing the results. Furthermore, the BO algorithm was applied to additional experimental data sets of ex-service (23,000 hrs) P91 steel tested at 20 °C, in which it located minimum MAPE values of 2.17 % and 1.19 % respectively.

From this work, it can be concluded that the incorporation of the BO algorithm improved the parameter identification process by reducing the time required by 91.1 % and decreasing the number of simulations by 92.5 % in comparison to the FFD approach, in which both approaches identified equivalent local minimum MAPE values.

Acknowledgements

I would like to thank my supervisor Prof. Noel O'Dowd for his support, guidance, and expertise offered throughout the project. Prof. O'Dowd was always available to discuss any questions or problems that arose and his insight was essential to the completion of the project.

I would also like to thank postdoctoral researcher Dr. Alison O'Connor for sharing with me her expertise regarding python coding and the implementation of python with Abaqus software. Dr. O' Connor's insight was vital to my understanding of python scripting in Abaqus and I have gained valuable knowledge from working alongside her throughout this project. Furthermore, I would like to acknowledge Dr. O' Connor for her work regarding the incorporation of the Bayesian Optimisation machine learning algorithm into the python script.

Additionally, I would like to express my gratitude to PhD researcher Pat Mongan for providing an initial Bayesian Optimisation algoritihm and for taking the time to discuss the code.

Finally, I would like to thank my family and friends for their support and encouragement throughout my college years.

Table of Contents

List of Figures

List of Tables

Nomenclature

Symbol/Abbreviation	Description	Units
AI	Artificial intelligence	-
BO	Bayesian optimisation	-
csv	Comma separated value	-
E	Young's modulus	GPa
f	Total void volume fraction	-
$\dot{f}$	Rate of change of total void volume fraction	s^{-1}
$\dot{f}_g$	Rate of change of void volume fraction due to growth	s^{-1}
$\dot{f}_n$	Rate of change of void volume fraction due to nucleation	s^{-1}
f_N	Volume fraction nucleating voids	-
FEM	Finite element method	-
FFD	Fractional factorial design	-
GTN	Gurson-Tvergaard-Needleman	-
GUI	Graphical user interface	-
i	Simulation number	-
m	Slope of extrapolated true stress-plastic strain data	MPa
MAPE	Mean absolute percentage error	%
ML	Machine learning	-
MSE	Mean squared error	-
q_1,q_2,q_3	Fitting parameters for GTN yield function	-
RF2	Vertical reaction force	(kN)
RT	Room temperature	°C
s_N	Standard deviation of nucleation strain normal distribution	-
SEM	Scanning electron microscope	-
T	Temperature	°C
U2	Vertical displacement	mm
UTS	Ultimate tensile strength	MPa
$\dot{\varepsilon}$	Strain rate	%/s
ε_E	Elastic strain	-
ε_P	Plastic strain	-
ε_N	Mean value of the nucleation strain normal value	-
ε_{UTS}	True strain at ultimate tensile strength point	-
$\dot{\varepsilon}^{pl}$	Plastic strain rate for loading in GTN model	s^{-1}
$\bar{\varepsilon}_m^{pl}$	Matrix equivalent plastic strain GTN model	-
σ_{Eng}	Engineering stress	MPa
σ_H	Hydrostatic stress	MPa
σ_m	Von Mises equivalent stress	MPa
σ_T	True stress	MPa
σ_{UTS}	True stress at ultimate tensile strength point	MPa
σ_y	Yield stress	MPa
ϕ	Yield function	-

1. Introduction

In engineering applications such as solid mechanics, the finite element method (FEM) is often implemented to determine an approximate numerical solution by dividing the model into small finite elements and applying the laws of continuum mechanics to the element groups (Malvern, 1969). The FEM uses an iterative approach, which can require thousands of repetitions to obtain a converged solution, generally computers are used to perform FEM using specialist finite element software such as Abaqus (Hadi and Bodhinayake, 2003). The latest versions of Abaqus 2020/2021 have incorporated the use of python coding, which is a renowned general purpose programming language widely used for scientific analysis (Abaqus, 2021a). Due to this upgrade, Abaqus users can now incorporate python libraries such as numpy and scikit-learn to improve data interpretation and statistical analysis, which provides an opportunity to incorporate machine learning (ML) into finite element analysis.

Artificial intelligence (AI) is a type of computer science which focuses on creating machines that can perform tasks, typically expected to require human intelligence (Attaran and Deb, 2018). Machine learning (ML) is a sub-set of AI focused on the design of general-purpose algorithms capable of automatically extracting information from data (Deisenroth *et al*., 2020). In recent years, AI and ML have been incorporated into engineering environments, particularly manufacturing as they result in cost savings and improved quality control (Monostori, 2002). Based on these benefits, the use of AI has extended into other areas for example the prediction of mechanical behaviour, optimising welding processes, and identifying steel processing defects (Liu *et al*., 2017; Wu *et al.,* 2019; Mongan *et al*., 2021).

The accurate determination of material properties is critical to ensure a material can withstand its designed loading conditions (Liu *et al.,* 2017). At present, material properties are generally determined by experimental testing, however experimental methods can be expensive, time consuming and wasteful of resources as, generally, materials must be tested to failure. The use of computational finite element analysis (FEA) software such as Abaqus is often implemented to analyse mechanical performance, under a range of conditions, based on a few relatively simple experimental results. Despite the practicality of simulating experimental tests on FEA software, it can be very computationally demanding to obtain accurate results, particularly for a non-linear analysis and therefore, minimising the number of simulations is critical.

In recent years, significant interest has been developed in accurately predicting the non-linear material response due to ductile damage (Yildiz and Yilmaz, 2020). Ductile damage refers to the growth, nucleation and coalescence of micro voids due to excessive plastic deformation, which degrades the mechanical performance of the material. Eventually ductile damage results in ductile failure, which has become one of the most frequent failure modes in engineering applications (Bonora, 1997). It is challenging to implement ductile damage into a FEA solution, due to the complexity of the required yield function known as the Gurson-Tvergaard-Needleman (GTN) damage model. Including a damage model in the computational analysis is essential to obtain an accurate

assessment of material behaviour beyond the ultimate tensile strength (UTS) of the material. Figure 1.1 shows the improved correlation between simulated results and experimental data when ductile damage is included.

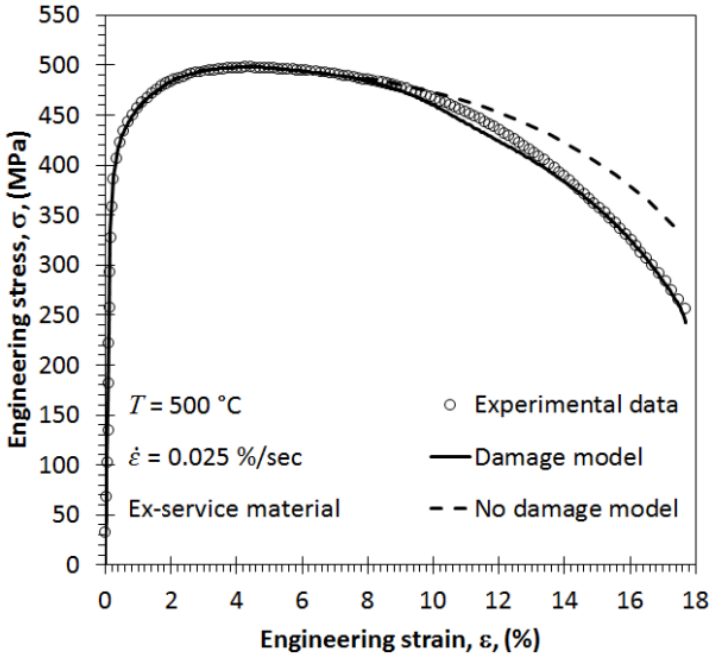


Figure 1.1, Engineering stress-strain data for experimental tensile testing of P91 steel, compared with simulated results with and without the GTN damage model (Meade *et.al*, 2020)

Including ductile damage in a FEA simulation using the GTN model requires the identification of several material parameters. These parameters are generally determined through iterative comparision of numerous FEA simulations to experimental data, which is time consuming and computationally expensive.

2. Objectives

The aim of this work is to create a framework that can identify the optimum GTN yield function parameters, thereby minimising the required time and computational expense. The framework combines Abaqus FEA software with a machine learning (ML) algorithm via the Python programming language to efficiently identify the optimum combination of material parameters.

The key objectives required to accomplish this aim are as follows:

1. To automate the generation and simulation of Abaqus models to represent a known tensile test specimen geometry with modifiable material parameters.
2. To automate the extraction of relevant load-displacement results from each completed simulation and store the data in an appropriate format.
3. To automate the comparison of simulated and experimental results to determine the lowest error and hence, the corresponding optimum GTN parameters.
4. To apply a Bayesian optimisation ML algorithm to autonomously reselect the material parameters based on learnings from past errors.

5. To compare the Bayesian optimisation approach to the fractional factorial design method in terms of the time and computational expense required to identify the optimum parameters.

3. Literature Review

This section outlines the published literature regarding the key areas of this research project, the material under study, the material response due to an applied uniaxial load, the proposed GTN approach to modelling ductile damage and the current ML methods that have been implemented at present to improve the GTN parameter identification process.

3.1. Material

P91 steel is a creep strength enhanced ferritic (CSEF) material containing 9% chrome and 1% molybdenum, that is designed to maintain material strength properties at increasingly high temperatures (Moreno, 2016). P91 steel is commonly used in the energy generation industry and more specifically in high temperature applications such as piping, pressure vessels and heaters. P91 steel maintains superior strength at high temperatures and has improved creep and corrosion resistance compared to other metals.

As the focus on energy generation shifts towards more carbon neutral sources such as nuclear energy, it is expected that P91 steel, an ideal material for the construction of nuclear power plants, will become increasingly more common (Moreno, 2016). As a result, there is growing interest in accurately predicting the material behaviour of P91 under ductile failure mechanisms.

3.2. Behaviour of the Material to an Applied Uniaxial Load

When a uniaxial tensile force is applied to a metallic material such as P91, the specimen experiences both elastic and plastic deformation (Lavocat, 2013). The elastic and plastic regions are represented by the stress-strain curve in Figure 3.1.

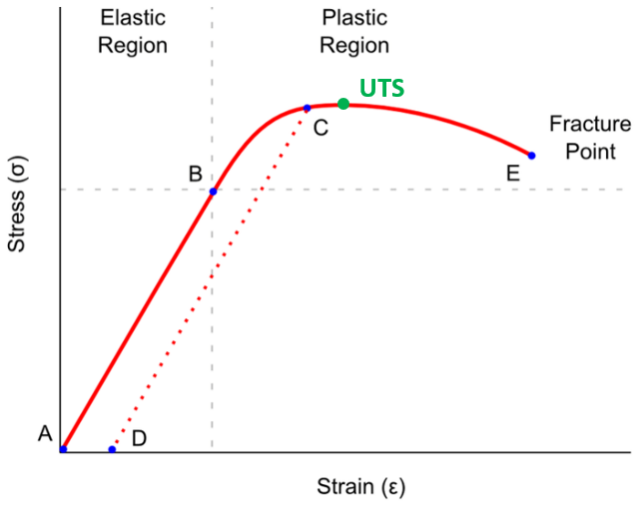


Figure 3.1, Stress strain curve for a metallic material, highlighting elastic and plastic regions (Lavocat, 2013).

Within the elastic region, any strain experienced by the specimen is reversible, therefore if the force applied is unloaded to zero, the initial strain corresponds to zero (Gilmore, 2015). In the elastic region, the Young's modulus refers to the stiffness of the material and affects the slope of the elastic region, in which the stress is linearly proportional to strain, corresponding to Hooke's law. The yield stress point denoted as B in Figure 3.1 refers to the maximum stress a specimen can endure, whilst maintaining the linear proportional stress-strain relationship.

Once the stress exceeds the yield stress of the material, the material enters the plastic region and experiences permanent deformation (Meade *et al.*, 2021). Permanent deformation refers to the strain, that remains within the material after the specimen has been unloaded, as portrayed by the dotted line CD in Figure 3.1. Most metals reach a maximum load known as the ultimate tensile strength (UTS) of the material, highlighted by the green point in Figure 3.1. In the initial plastic region, between the yield point and the UTS, the material experiences strain hardening, where both stress and strain are increasing.

Once the UTS has been surpassed in ductile materials, the rate of strain hardening is exceeded by the decline in load bearing capabilities, due to a reduction in the cross sectional area of the specimen (Meade *et al.*, 2021). This process is known as necking, which is associated with the occurrence of ductile damage at the centre of the specimen.

3.3. Modelling Ductile Damage

It is relatively straightforward to replicate the elastic material response in FE simulations, as the Youngs modulus and Poisson's ratio are the only material properties required and are readily available for most materials. It is more complicated to model the strain hardening response of a material between the yield stress and the UTS point, as true stress and true strain values must be calculated (Tu *et al*., 2020). The true strain and true stress values are determined, by applying Eqn. 3.1 and Eqn.3. 2 to the engineering stress and strain values determined from the experimental load-displacement data.

$$\varepsilon_T = \ln(1 + \varepsilon_{Eng}) \qquad \textbf{Eqn. 3.1}$$

$$\sigma_T = \sigma_{Eng}(1 + \varepsilon_{Eng}) \qquad \textbf{Eqn. 3.2}$$

The conversion to true stress and true strain is required to account for the change in the cross-sectional geometry due to extensive plastic deformation (Tu *et al*., 2020). Figure 3.2 highlights that true stress always increases with increasing true strain, however after the UTS point (ε_u), this is no longer accurate as the stress does not increase with strain. Therefore, the true stress-strain data must be linearly extrapolated to model the behaviour of the material after the onset of necking, as depicted by the red line in Figure 3.2. This method of analysis of material behaviour in the necked region, is a simplification and the true stress-strain data alone does not provide an accurate prediction of the drop in material performance. Therefore, to improve the accuracy of a simulation, ductile damage must be included.

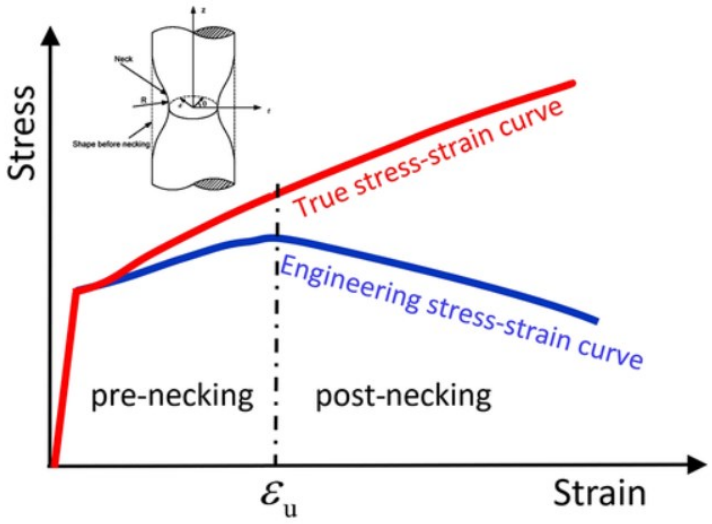


Figure 3.2, Comparison between the engineering stress-strain curve against the extrapolated true stress-strain curve from the onset of diffuse necking highlighted in the graph by the point ε_u (UTS) (Tu *et al*., 2020).

Ductile damage can be modelled using a local approach, which implements constitutive models for damage and plasticity (Chen, 2019). In this approach, ductile damage is represented using continuum-based methods such as the GTN damage model, where the impact of voids and microcracks is denoted by a yield function.

3.3.1. Gurson-Tvergaard-Needleman (GTN) Damage Model

The Gurson model was developed based on previous methods proposed by McClintock (1968) and Rice and Tracey (1969), which originally only analysed the development of a single spherical void in an infinite region and did not account for the interaction between voids. To resolve this issue, Gurson (1977) proposed a solution using a continuum-based method centred on uniformly spaced cylindrical voids within an incompressible plastic matrix.

The Gurson damage model, was improved by Tvergaard, *et al*. (1981) to accommodate different void shapes, by adjusting the original Gurson (1977) fitting parameters (q_1, q_2, q_3). The resultant GTN yield function is shown in Eqn. 3.3 for which Tvergaard, *et al*. (1981) originally set the fitting parameters as q_1 =1.5, q_2 = 1 and $q_3 = q_1^2$, however these parameters can be adjusted further to improve the solution.

$$\phi = \frac{\sigma_m^2}{\sigma_y^2} + 2f q_1 \cosh\left(\frac{3}{2} q_2 \frac{\sigma_H}{\sigma_y}\right) - (1 + q_3(f)^2) = 0 \qquad \textbf{Eqn. 3.3}$$

In Eqn. 3.3, *f* represents the total void volume fraction and if the effect of voids is neglected (*f* = 0), the equation can be simplified to show the effective Von Mises stress σ_m equals the yield stress σ_y indicating classical plasticity behaviour (Tvergaard, *et al*., 1981). The inclusion of void growth complicates the yield function as it now depends on the three fitting parameters and the hydrostatic stress σ_H , for which an increase in *f* results in a sooner onset of yielding in the material.

The model was further extended by Tvergaard and Needleman (1984) to account for the drop in stiffness of the model by incorporating the change in void volume fraction due to the experienced deformation into the yield function. The rate of change of void volume fraction $\dot{f}$, is represented by the sum of the void growth $(\dot{f}_g)$ and the nucleation of new voids $(\dot{f}_n)$ as shown in Eqn. 3.4. (Tvergaard and Needleman, 1984).

$$\dot{f} = \dot{f}_g + \dot{f}_n \qquad \textbf{Eqn. 3.4}$$

The change in void growth is dependent on the conservation of mass determined by calculating the relative density using the void volume fraction for the GTN plastic loading increment ($\dot{\varepsilon}^{pl}$) as highlighted in Eqn. 3.5 (Tvergaard and Needleman, 1984).

$$\dot{f}_g = (1 - f)\dot{\varepsilon}^{pl} : \mathrm{I} \qquad \textbf{Eqn. 3.5}$$

Where I is a second order unit tensor representing a three dimensional material. The rate of void nucleation is dependent on the effective plastic strain as shown in Eqn. 3.6.

$$\dot{f}_n = A\dot{\bar{\varepsilon}}_m^{pl} \qquad \textbf{Eqn. 3.6}$$

For which parameter A ensures A/f_N follows a normal distribution for the matrix equivalent plastic strain ($\bar{\varepsilon}_m^{pl}$) with a mean strain void nucleation ε_N, standard deviation s_N and the impact of void volume fraction due to nucleated voids f_N as portrayed in Eqn. 3.7.

$$A = \frac{f_N}{s_N\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{\bar{\varepsilon}_m^{pl} - \varepsilon_N}{S_N}\right)^2\right] \qquad \textbf{Eqn. 3.7}$$

To implement ductile damage in a standard Abaqus FEA model, the seven GTN parameters q_1, q_2, q_3, f_N, ε_N, s_N, and f as portrayed in Eqns. 3.3 - 3.7 are required. A study conducted by Slimane *et al.*, (2015) investigated the sensitivity of an Abaqus solution to the f_N and q_2 parameters. The study concluded that changing f_N and q_2 had no impact on the elastic part of the solution however beyond the UTS at the point of void initiation, a larger f_N term instigates the onset of yielding sooner and impacts the fracture response of the model. In addition, it was concluded that larger q_2 values resulted in increased softening of the material. The stress-strain curve comparisons for each varied parameter are portrayed by Figure A.1 (a) and (b) respectively in Appendix A.

3.4. Identification of the GTN Parameters Through Experimentation

A recent study conducted by Yildiz and Yilmaz, (2020), investigated the possibility of experimentally determining the GTN parameters for a 6061 Aluminium alloy tensile tested at five different heat treatments. In the study they assumed q_1 =1.5, q_2 =1 and q_3 = 2.25 according to Tvergaard, *et al*. (1981) and focused on the experimental determination of void volume fraction f and the void nucleation terms f_N , ε_N and s_N. Yildiz and Yilmaz, (2020) precisely cut the necked area of the specimen into four regions and individually measured the density of each region using a Precisa XB 220A analytical balance to account for change in density due to the presence of voids (f). The initial void

fraction before the tensile test was determined by using a scanning electron microscope (SEM) micrograph to analyse the surface of the material, which required over 25 micrographs to obtain an accurate representation. Similarly, to determine the nucleation terms f_N, ε_N and s_N , SEM micrographs at x5000 magnification were used to detect second phase particles relating to the effect of void nucleation.

To validate the experimentally determined GTN parameters, a FE analysis was conducted using the experimentally determined parameters and the load-displacement curves for simulation and experimental results were compared. Yildiz and Yilmaz, (2020) concluded that the FE simulations containing the experimentally determined GTN parameters for each heat treatment showed good correlation to the experimental data as shown in Figure 3.3.

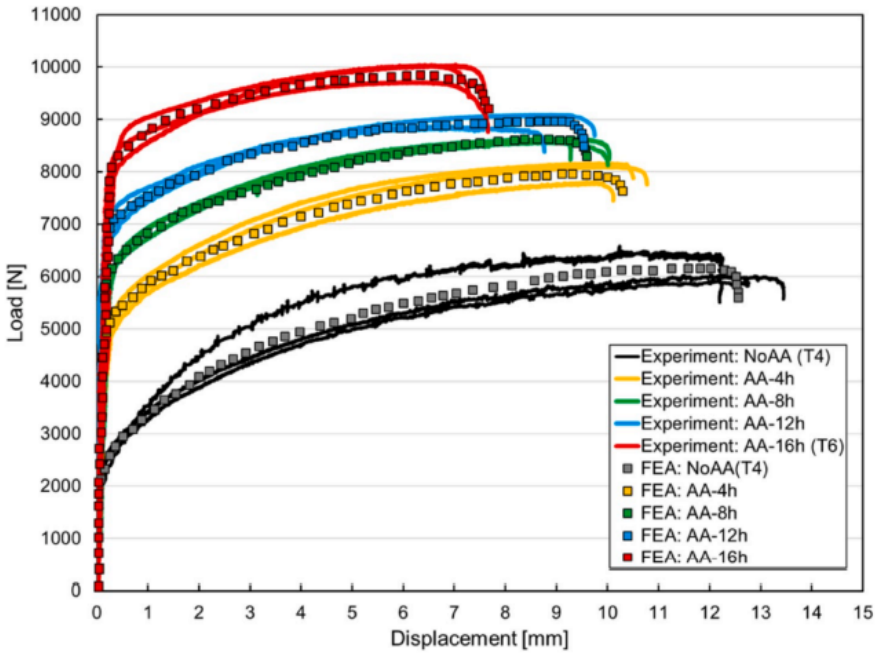


Figure 3.3, Experimental and simulated load-displacement curve comparision for each heat treatment (Yildiz and Yilmaz, 2020)

Despite the good correlation the process required to experimentally determine the GTN parameters is time consuming, expensive, and complex due to the slicing of the necked region and the numerous high-resolution SEM micrographs that had to be interpreted using MATLAB code.

3.5. The Application of Machine Learning Techniques in the Modelling of Ductile Damage

ML approaches have become more common in material science, due to the ability of these algorithms to accurately predict solutions to complex problems, for which preferred analytical or empirical solutions cannot be obtained (Liu, *et al.,* 2020). The advantage of implementing ML techniques is evident in solving nonlinear problems, that contain a complicated relationship between high dimensional physical parameters.

It is clear from Eqns. 3.3 – 3.7 in Section 3.3.1, that the GTN yield function is quite complex due to the seven GTN parameters that must be determined to accurately implement the effect of ductile

damage into a finite element simulation. Therefore, several studies have incorporated ML methods to reduce the time and computational expense required, to numerically identify the optimum GTN parameters and improve the overall process of modelling ductile damage. In the following subsection, studies that applied ML techniques related to predicting the nonlinear behaviour of a material due to ductile damage are discussed in detail.

3.5.1. Application of Artificial Neural Networks to Identify GTN Parameters

Artificial neural networks (ANN) are a subset of machine learning, known as deep learning (Paermentier *et al*., 2021). An ANN is based on a biological computational network consisting of a single input layer, multiple hidden layers, and an output layer. A basic ANN structure for the general identification of the GTN parameters is portrayed by Figure A.2 in Appendix A.

Abbassi *et al.* (2013) applied a detailed ANN structure, to identify the GTN parameters required to model sheet metal forming in Abaqus FEA software. The ANN model proposed by Abbassi *et al.* (2013), used multilayer neural networks with a skilled back propagation supervised learning algorithm, created using MATLAB software, which trained the ANN by adjusting the weights between the layers. To implement the back propagation learning algorithm, Abbassi *et al.* (2013) had to initially define several parameters such as the training rate, moment coefficient and the random initialisation of weights. A specimen was modelled using Abaqus FEA software and through a combination of experiments and numerical FEA simulations, a database to train the ANN model was created, as outlined in Figure 3.4. From the study, an ANN was successfully formed to identify the required GTN parameters, by comparing simulated and experimental results using a mean square error (MSE) analysis, which indicated good correlation between the simulated and experimental outputs.

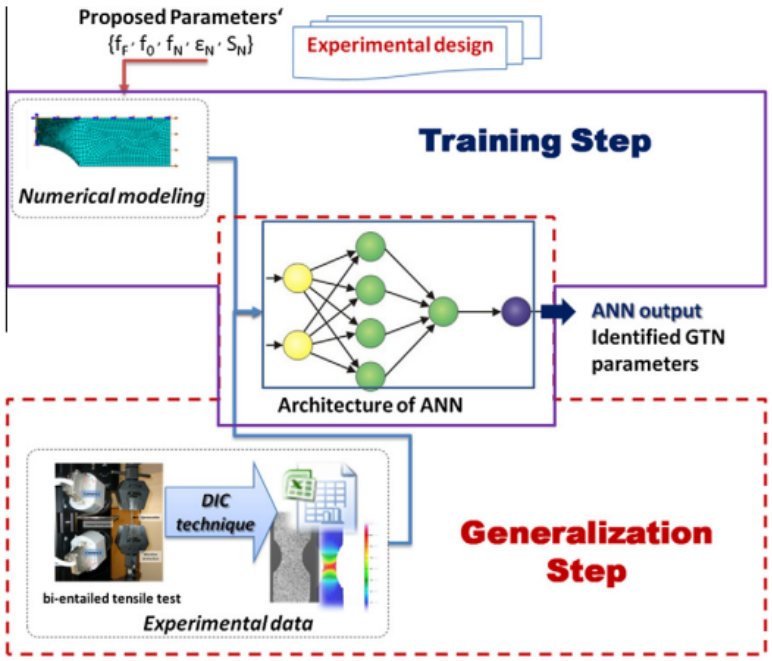


Figure 3.4, Process of combining experimental and simulated results to train the ANN to identify the GTN parameters in sheet metal forming (Abbassi *et al.,* 2013).

Abendroth and Kuna (2006) conducted a similar study to identify the GTN ductile damage and fracture parameters from a small punch test, using an ANN trained from a database provided by FEA simulations. In this study, the input layer contained six units, corresponding to the required GTN parameters, fifty units in the hidden layer and a single output unit. A large ANN was required due to the complexity of the GTN yield function and after 1000 training cycles, the ANN produced a MSE of 1.9×10^{-4} and 2.0×10^{-4} for the training and validation sets. The study concluded that the inclusion of the ANN in the GTN parameter identification process, resulted in improved correlation between simulated and experimental results.

A key challenge encountered by both Abbassi *et al.* (2013) and Abendroth and Kuna (2006), was gathering enough information from FEA simulations to provide a sufficient database for the ANN to be trained. In each study, numerous finite element simulations were required to build the database, for which each simulation had to be completed before extracting the data. This method was very inefficient, as it wasted computational time and expense completing excessive simulations for incorrect parameter configurations. Therefore, a key area in which the identification of the GTN parameters could be improved is the creation of an algorithm that could minimise the number of simulations required to identify the optimum GTN parameters. This modification to the current process would reduce the time and computational expense wasted in completing simulations for incorrect GTN parameter configurations.

3.6. Classification of Optimum GTN Parameters

The purpose of the ANNs created by Abbassi *et al.* (2013) and Abendroth and Kuna (2006), was to determine the optimum GTN parameters by comparing the simulated and experimental load-displacement curves to examine the respective deviation. In order to determine the optimum GTN parameters, 1,024 simulations containing various GTN parameter combinations had to be completed and then compared against the experimental data, for which the simulated curve resulting in the lowest error in comparison to the experimental curve was deemed to contain the optimum GTN parameters.

In assessing the deviation between the respective curves, Abendroth and Kuna (2006) implemented the mean square error (MSE) technique, which is based on the Euclidean distance formula that measures the Pythagorean distance between two co-ordinates. Hence, the difference between the experimental and predicted simulated values at a point on the curve can be determined and then squared, which always results in a positive error. The total MSE between the curves can then be determined by summing the individual squared differences for each point and dividing by the total number of data points (n) as shown in Eqn. 3.8.

$$MSE = \frac{\sum_{i=1}^{n}\left(y_{Exp} - y_{Sim}\right)^2}{n} \qquad \textbf{Eqn. 3.8}$$

The MSE forecasting technique has many advantages for example the error always being positive which prevents positive and negative errors cancelling out (Brassington, 2017). In addition, the MSE

technique increases the weight of larger errors as the difference is squared. However, a key disadvantage of the MSE method is that the error is scale dependent and therefore, understanding the significance of the error depends on the magnitude of the experimental and simulated values. The scale dependency of the MSE increases the difficulty in categorising the error as acceptable in comparison to other methods such as the mean absolute percentage error (MAPE), for which the error can be assessed on a scale of 0 -100 %.

3.7. Summary

In conclusion, P91 steel is an important and relevant material to investigate the effects of ductile damage, due to its role in the construction of nuclear energy power plants. The elastic and initial plastic response of the material up to the UTS point is relatively straightforward to predict using finite element software, however after the UTS point, necking due to ductile damage occurs. Ductile damage results in a complex non-linear material response, that significantly compromises the mechanical performance of the material. It is essential to accurately model this region, to obtain an insight into the material behaviour before ductile fracture occurs. Ductile damage is generally modelled using the GTN damage model, however it is extremely difficult to predict the GTN damage model parameters to input into FEA software, to obtain an accurate solution.

Although recent studies have investigated the possibility of experimentally determining these parameters, generally numerical approaches have been implemented. To improve the parameter identification process, numerous studies have applied machine learning techniques such as artificial neural networks that can be trained to optimise these parameters, by comparing completed FEA simulations against experimental results. Studies have shown excellent correlation between FEA and experimental load-displacement curves, with a MSE value as low as 1.9×10^{-4}. A major disadvantage of applying AI techniques to identify the optimum GTN parameters, is the number of simulations that must be completed to train the ANN, which wastes time and unnecessary computational power. Therefore, the literature identified a clear avenue for future work in this area, in which a ML algorithm could be designed capable of comparing experimental data and simulated results and hence, reselecting the GTN parameters based on learnings from the calculated error. This would be a huge advantage in the GTN parameter identification process, as it would reduce the number of required simulations and minimise the time and computational expense wasted for simulations containing incorrect GTN parameters.

4. Numerical Finite Element Analysis

The numerical finite element analysis replicated an entire experimental tensile test using Abaqus/Standard 2020 for a typical round specimen geometry illustrated in Figure 4.1. The analysis was conducted on the gauge area of length 25 mm and diameter 4 mm.

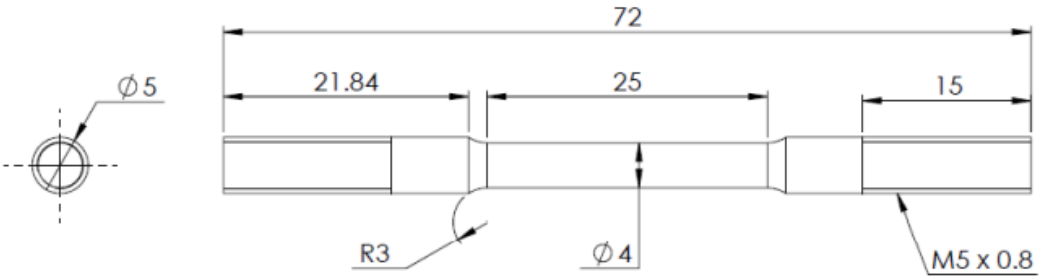


Figure 4.1, Engineering tensile test specimen dimensioned drawing (Meade *et al.*, 2021)

The key features of the analysis and the steps required to convert the experimental tensile test to a standard Abaqus FEA problem are outlined in subsections 4.1 – 4.4.

4.1. Model Geometry and Boundary Conditions

As the test specimen is axisymmetric along the Y-axis and symmetric across the X-Z plane as shown in Figure 4.2 (a), only a 2D quarter section of the gauge area was required to achieve a complete 3D solution by conducting an axisymmetric analysis. As the analysis was axisymmetric, the model was fixed along the axis in the x direction. In addition, a symmetry condition was applied to the bottom surface and a displacement was applied to the top to deform the model as per Figure 4.2 (b)

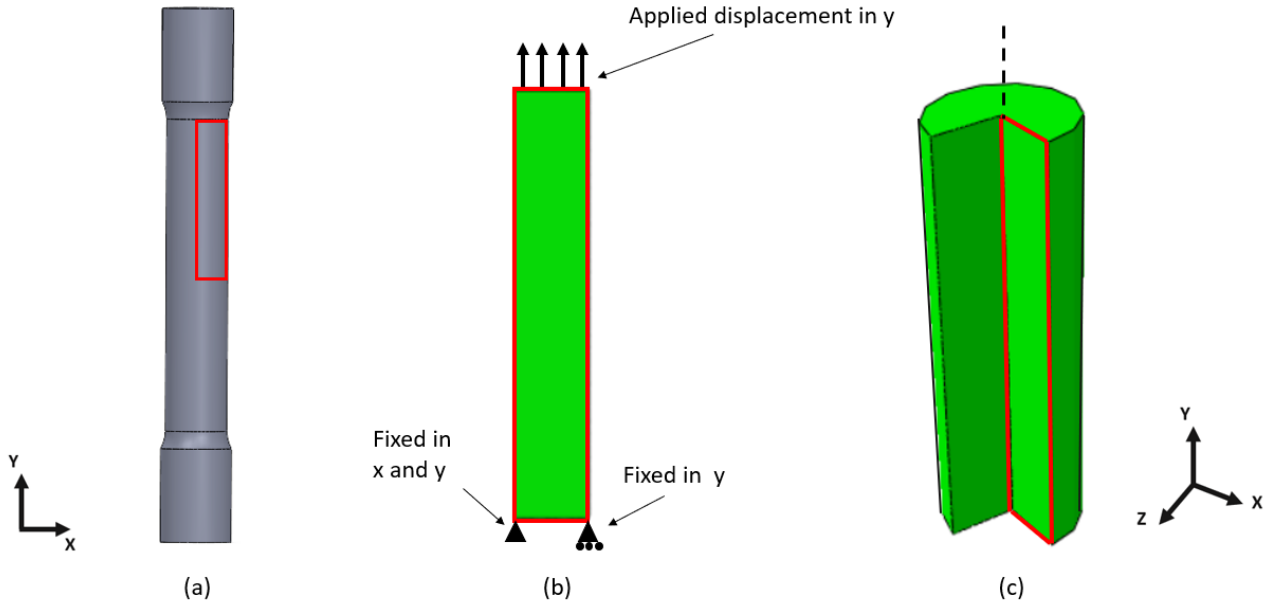


Figure 4.2, (a) Tensile test specimen (b) rectangular quarter section highlighting applied displacement and corresponding boundary conditions (c) full 3D model obtained from axisymmetric analysis

In an experimental test, necking occurs in a model due to local imperfections in the material, therefore a geometrical imperfection must be applied to induce necking at the centre of an Abaqus standard FEA model. A geometric imperfection in the shape of a triangular notch of width 0.015 mm and height 0.1 mm was applied to the bottom right corner of the model in Figure 4.2 (b), to ensure the model necked correctly at the centre of the specimen.

4.2. Material Properties

To numerically model an experimental tensile test, elastic properties, plastic properties, and the porous metal plasticity model were required as outlined in subsections 4.2.1 and 4.2.2.

4.2.1. Elastic and Plastic Material Properties

The elastic and plastic properties were determined from converting experimental load-displacement data to true stress-strain values. In this project, the material data obtained from an experimental tensile test conducted and outlined by Meade *et al.*, (2021) for ex-service (63,000 hours) P91 steel from the Coolkeeragh power plant tested at 500 °C was incorporated into the Abaqus model.

For the elastic part of the analysis the Young's modulus was input in MPa and as the analysis was independent of Poisson's ratio it was arbitrarily set to 0.3. For the plastic analysis, the true stress and plastic strain up to the UTS point had to be input from the yield point onwards. To model necking beyond the UTS point, the true stress and plastic strain data had to be extrapolated beyond the UTS point, for which the slope (m) for the equation of the extrapolated data was an important parameter that had to be considered to accurately model the tensile test. Figure A.3 in Appendix A highlights the impact of changing the extrapolated data slope on the simulated load-displacement results.

The range for the equation of the extrapolated data was defined by determining a maximum and minimum slope for the extrapolated line. The maximum slope was calculated by assuming a linear relationship between each of the last five points before the UTS point and the minimum slope was set to zero as true stress can never decrease with true strain. Each time the model was created a slope within this range was selected as defined by the yellow region in Figure 4.3. The y-intercept of the equation of the line was than determined by assuming the UTS point was on the line and inserting the slope into Eqn. 4.1, ensuring a smooth transition from the experimental to extrapolated data.

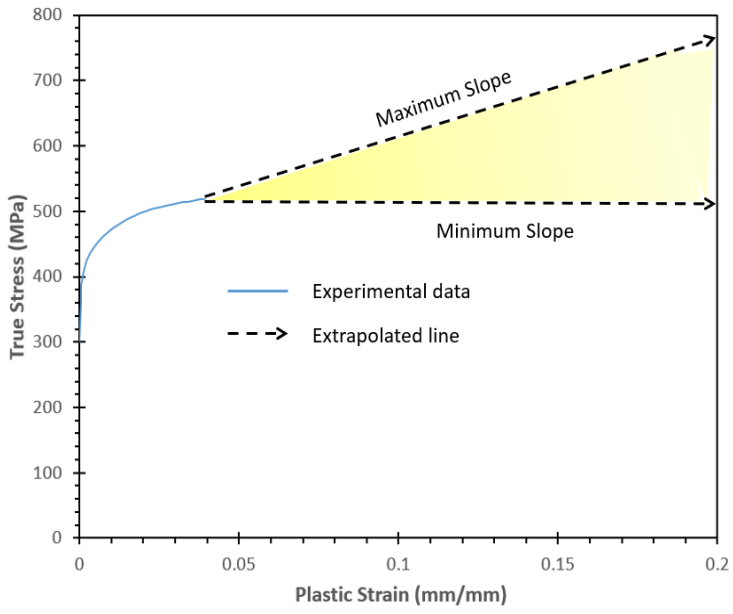


Figure 4.3, True stress-plastic strain data up to UTS point. Highlighted range showing the effect of slope selection on extrapolated data.

$$y = \sigma_{UTS} - (m \times \varepsilon_{UTS}) \quad \textbf{Eqn. 4.1}$$

4.2.2. Porous Metal Plasticity Model

Ductile damage was included in the model using the porous metal plasticity model with the void nucleation sub option (Abaqus, 2021b). These options require seven parameters as highlighted in Table 4.1. The minimum and maximum values for each parameter were determined based on literature studies and the limitations outlined in the Abaqus manual (Meade *et al.*, 2021; Kiran and Khandelwal, 2014; Abaqus, 2021b). The focus of the python script which is outlined in Section 5.1 was to determine the optimum value of each parameter within the specified range in addition to the slope of the extrapolated data as previously discussed.

Table 4.1, Specified range of GTN parameters required for the Abaqus model

GTN Parameter	Minimum	Mean	Maximum
q_1	0.90	1.25	1.60
q_2	0.90	1.00	1.10
q_3	0.81	1.69	2.56
ε_N	0.25	0.33	0.40
S_N	0.10	0.15	0.20
f_N	0.03	0.06	0.09
f	0.0013	0.0014	0.0015

Each parameter listed in Table 4.1 is dimensionless and two significant figures after the decimal place within this range was selected for each parameter based on the results of a parameter sensitivity study outlined in Section 6.2.

4.3. Mesh Generation

A uniform linear quadrilateral mesh of element type CAX4R was applied to the Abaqus model. CAX4R refers to a continuum axisymmetric element with four node reduced integration, which is frequently used in literature to model a tensile test as it reduces the computational time, whilst maintaining the accuracy of the simulated results. (Zhong, *et. al*., 2019). A 5,000-element mesh was deemed sufficiently dense to model the problem based on a conducted mesh study in which the results from the study are highlighted in Section 6.1.

4.4. Steps and Incrementation

The Abaqus model required two steps an initial and a displacement step. The symmetry conditions were applied on the initial step and a second static step was created to apply the displacement boundary condition which adjusted the number of increments based on the rate of convergence. As the Abaqus problem was independent of time, a period of 1 was applied, with the minimum and maximum increments set to 0.01 to achieve enough data points to plot a smooth simulation load-displacement curve.

5. Python Script Development

In this section, the key stages in the development of the python script to accurately and autonomously model the FEA problem outlined previously in Section 4 are described in subsections 5.1 - 5.7. All python files are located in a private repository titled “williamb21/FYP” which is available on a software development cloud-based platform known as GitHub, for access contact Dr. Alison O’Connor. Furthermore, a procedure describing how to apply the code is outlined in Appendix B.

5.1. Overview

The designed python script as outlined by the flowchart in Figure 5.1, minimises the user input required to obtain the optimum GTN parameters. The script can be run from python software known as PyCharm with a single click, which automatically checks out an Abaqus license and runs the Abaqus models without opening the graphical user interface (GUI). The script continuously generates Abaqus models and compares the results to the experimental data until the maximum number of simulations (i) is reached, or until a minimum converged MAPE of less than 5 % is achieved. After each loop, the Bayesian optimisation (BO) algorithm reselects new GTN parameters based on learnings from previously calculated MAPE values.

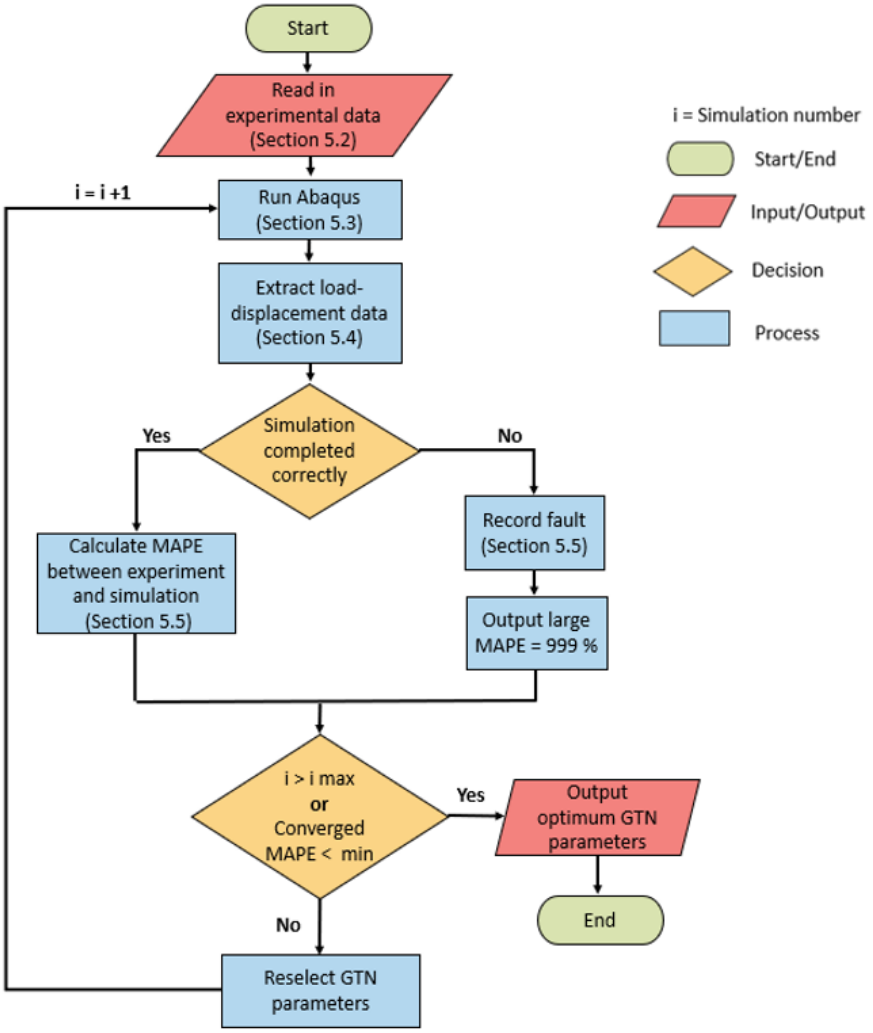


Figure 5.1, Flowchart illustrating the key steps in the python script required to solve the numerical analysis

5.2. Analyse Experimental Data

Before running the code, the user must have the experimental load-displacement data in csv file format located in the working directory of the python scripts. The script locates the csv file and automatically converts the load-displacement data to engineering stress (MPa) and strain up to the UTS point based on the input gauge length and diameter of the specimen. The engineering stress strain values are converted to true stress and strain using Eqns. 3.1 - 3.2 in Section 3.3. The yield stress point is automatically identified by interpolating the experimental data and locating the maximum peak of the second derivative of strain, which represents the end of the linear region. Hence, the Youngs modulus is calculated by identifying the line of best fit of the linear region. The code outputs a plot of the experimental true stress-strain data highlighting the yield point as shown in Figure 5.2 and passes the yield stress and Youngs modulus values to the Abaqus model.

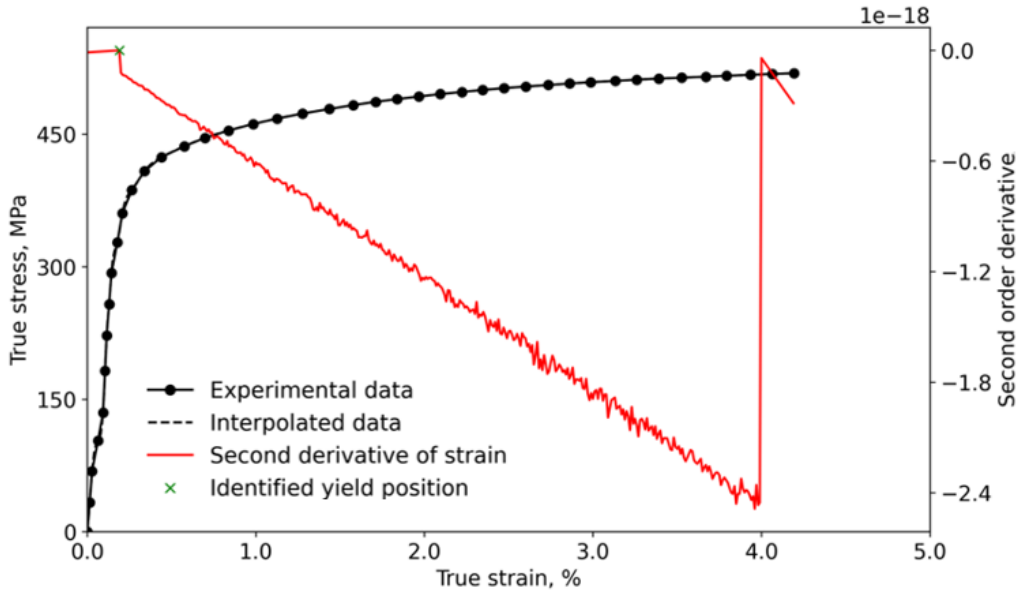


Figure 5.2, Plot of true stress and true strain highlighting the maximum peak corresponding to the second derivate of strain

The script converts the true strain to plastic strain using Eqn. 5.1 and outputs a csv file titled “Plastic Properties” to the working directory and the “FEA Output” folder, which contains the true stress and plastic strain values from the yield point to the UTS point.

$$\varepsilon_p = \varepsilon - \varepsilon_e = \varepsilon - \frac{\sigma}{E} \qquad \textbf{Eqn. 5.1}$$

5.3. Build and Run Abaqus Model

The code automatically creates the Abaqus model corresponding to the steps outlined in Section 4 in which the material parameters are the only modifiable section. The Young’s modulus and true stress-plastic strain up to the UTS are constant for every model, therefore the code reads in the data from the csv file and then adds the extrapolated data based on the selected slope for the simulation and the UTS point. In addition, the GTN parameters as outlined previously in Table 4.1 are modified for each simulation.

The code runs the job name corresponding to the simulation number to the maximum experimental displacement value and outputs all Abaqus files for each respective job to the “FEA Output” folder.

5.4. Extraction of Load-Displacement Data

After each simulation, the simulation results are located and the relevant load-displacement results are extracted as portrayed by the flowchart in Figure 5.3.

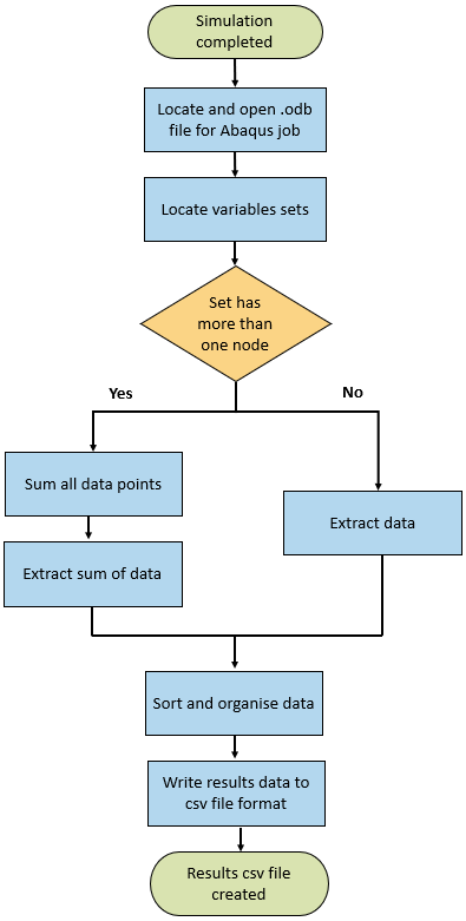


Figure 5.3, Flowchart illustrating the automatic extraction of the load-displacement data

The code extracts the variables RF2 on the bottom edge set and U2 on the top left vertex which refer to the vertical load and displacement respectively. As the bottom edge contains more than one node the RF2 values are summed together, and the summed value is exported for each time increment. As the top left vertex contains only one node the data for each time increment is extracted without any modifications. The data is sorted in a results dictionary for which a dictionary is a data storage method capable of storing multiple values under a single heading known as a key for example ‘RF2 or ‘U2’ (Nelli, 2018). The creation of a results dictionary improves the exportation of the data to csv file format as multiple RF2 values can be listed in rows under a column heading ‘RF2’ and likewise the ‘U2’ values can be located in the next column, which makes the csv file easy to read.

In addition to the vertical reaction force and displacement, the horizontal displacement on the top right corner of the model is recorded using the same process outlined in Figure 5.3 and is used to identify if the model incorrectly necks at the top instead of the bottom containing the inserted notch.

5.5. Comparison Method of Experimental and Simulated Results

As highlighted previously in Figure 5.1, the code will assess the deviation between the experimental and simulated load-displacement curves if the simulation completes correctly. A correct simulation is classified as a simulation that reached the maximum experimental displacement and necked only at the mid-gauge length region, containing the notch. The section of code required to compare the experimental data and simulated results is outlined by the flowchart in Figure 5.4.

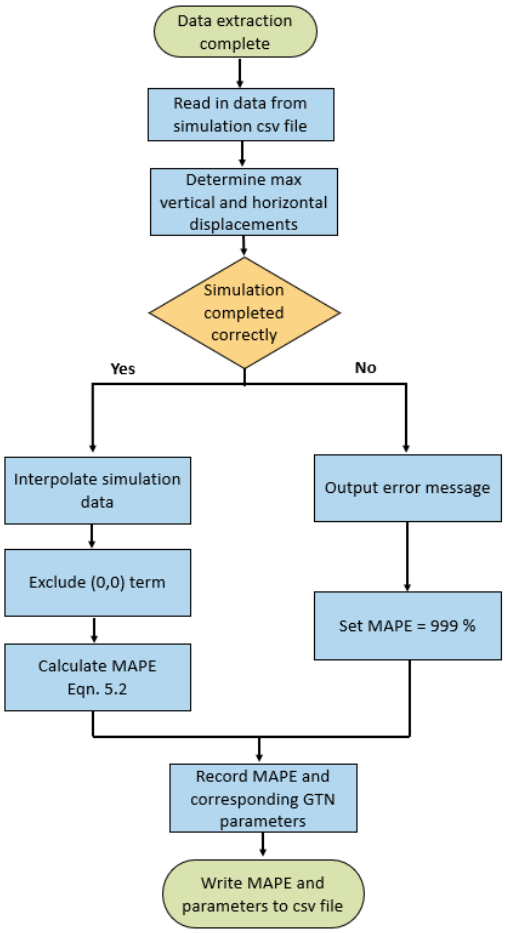


Figure 5.4, Flowchart illustrating the steps required to compare the experimental and simulated results

To accurately compare the experimental and simulated load-displacement curves, the simulated data must be interpolated to ensure experimental and simulated force values are compared at identical displacements. To interpolate the data, the numpy library containing the interp1d function is required which assumes a linear relationship between the simulation data points to obtain the simulated load values at the corresponding experimental displacement values (Nelli, 2018). The interpolation is effective due to the high number of simulation data points output according to the incrementation defined in Section 4.4, which results in a small difference of less than 1.3 % between the points.

The absolute percentage error (APE) calculates the positive difference between the experimental and simulated force values. The APE is calculated for each interpolated data point excluding the (0,0) terms, as zero cannot be the denominator of a fraction. Individual APE values are then summed and

averaged to obtain a single value representing the overall fit between the simulated and experimental curves known as the mean absolute percentage error (MAPE), calculated as shown in Eqn. 5.2. The MAPE represents a percentage value, providing the user with an accessible method of comparing various simulation results quickly (Mongan, *et. al.* 2022).

$$MAPE = \frac{1}{n} \times \Sigma \frac{(|Experimental\ force - Simulated\ force|)}{|Experimental\ force|} \times 100 \qquad \textbf{Eqn. 5.2}$$

If the simulation does not complete correctly, the experimental and simulated data is not compared. Instead, a large MAPE of 999 % is assigned to the simulation. For each simulation either the calculated MAPE or assigned MAPE of 999 % is output to an existing csv file with the corresponding GTN parameters of the simulation and the job number.

5.6. Reselection of Material Parameters

As seen from the overview flowchart in Figure 5.1, a loop is created within the code to continue running simulations until either a maximum simulation number is reached or a converged minimum MAPE is achieved. Each time a new loop runs the material parameters are reselected. In this work, two methods of autonomously reselecting the material parameters are investigated, the fractional factorial design (FFD) method and the Bayesian optimisation (BO) machine learning algorithm, which are outlined in detail in subsections 5.6.1 and 5.6.2 respectively.

5.6.1. Fractional Factorial Design

Initially FFD was applied to the parameter reselection process in which a fraction of all possible outcomes was determined to provide a baseline comparison to the machine learning approach (O'Mahoney *et al.*, 2013). FFD was chosen as a baseline comparison to minimise the total simulations due to the number of parameters required and the wide range of each parameter that must be reselected. As there were eight parameters that must be varied, the seven GTN parameters and the slope of the extrapolated line, increasing the number of values per parameter drastically increased the total number of simulations as shown in Eqn. 5.3.

$$Total\ simulations = (Number\ of\ values)^8 \qquad \textbf{Eqn. 5.3}$$

In this approach, each parameter was limited to three values, the maximum, the minimum and the mean of the ranges outlined previously in Table 4.1. This resulted in a total of 6,561 simulations, therefore the number of parameters was reduced to six by assuming $q_3 = q_1^2$ and that s_N is constant at 0.1, which was frequently seen in literature and reduced the total number of simulations to 729.

The FFD was easily implemented into the code outlined in Section 5.1 by creating a loop for each parameter list containing the minimum, mean and maximum values. In the code each loop was nested in which a nested loop contains an inner loop within an outer loop and iteratively runs through all possible combinations of the listed values (Nelli, 2018). As loops are a common tool in python programming, the reselection of material parameters could be accomplished by the Abaqus python software version 2.7.15.

5.6.2. Bayesian Optimisation

To improve the parameter reselection process, a machine learning based approach known as Bayesian optimisation (BO) was applied to the problem by modifying code designed previously by Mongan *et. al.*, (2021). BO is a sequential model-based optimisation algorithm, which determines global optimums for an unknown function. In this work, the BO algorithm is modified to reselect the next material parameter configuration based on learnings from the calculated or assigned MAPE of the previous simulation.

The BO algorithm requires five initial estimates to train the algorithm, which provides an insight into how the parameters effect the calculated MAPE. For the five initial estimates, the seven GTN parameters are set as constant at a mean value of the range depicted previously in Table 4.1 and the slope is varied at five different increments. After the fifth simulation, the BO algorithm starts to reselect parameters by initially varying a single parameter at a time and analysing the effect of the change on the calculated MAPE.

The incorporation of BO into the code outlined in the Section 5.1 was more challenging in comparison to the FFD method due to the python limitations of Abaqus FEA software. As the BO functions require the scikit-learn python library only available in python version 3.7 or later, they cannot be directly called in Abaqus which only contains python version 2.7.15. Therefore, the code had to be modified to analyse the experimental data in PyCharm software containing python version 3.7 as the automatic determination of the yield point and Young's modulus required the linear regression scikit-learn function. The key material properties such as the maximum slope, yield stress, Young's modulus and the GTN parameters were then transferred to Abaqus using the command line interface. On completion of the Abaqus simulation, the job name is passed back to PyCharm through the command line interface and the postprocessing of the data can be conducted in PyCharm. Hence the calculated MAPE can be directly input into the BO algorithm to reselect the next parameter configuration.

5.7. Script Outputs

After the experimental data analysis is completed, an image of the plot highlighting the identification of the yield stress and Young's modulus is output to the working directory. In addition, the experimental true stress-plastic strain properties determined up to the UTS point is output in csv file format into the working directory.

After each Abaqus simulation is completed all Abaqus files are saved to the "FEA Output" folder in the working directory according to the job name. In addition, the extracted load-displacement data is saved in csv file format within the folder. In the working directory, the "Bayesian Outputs" csv file contains all simulations regardless of completion, highlighting the MAPE for each simulation and the corresponding parameters to ensure ease in identifying the optimum GTN parameters. To obtain a graphical comparison of the experimental and simulated results a separate python script must be applied, which requires the experimental data, the simulation number and the name of the working directory containing the simulated load-displacement files.

6. Results and Discussions

In this section, the results regarding the generation of the numerical analysis and the application of the python script to solve the analysis on the Coolkeeragh material are discussed in detail. The experimental load-displacement curve obtained from the tensile test conducted by Meade *et al.*, (2021) required to build the Abaqus model and assess the simulation results is shown in Figure 6.1.

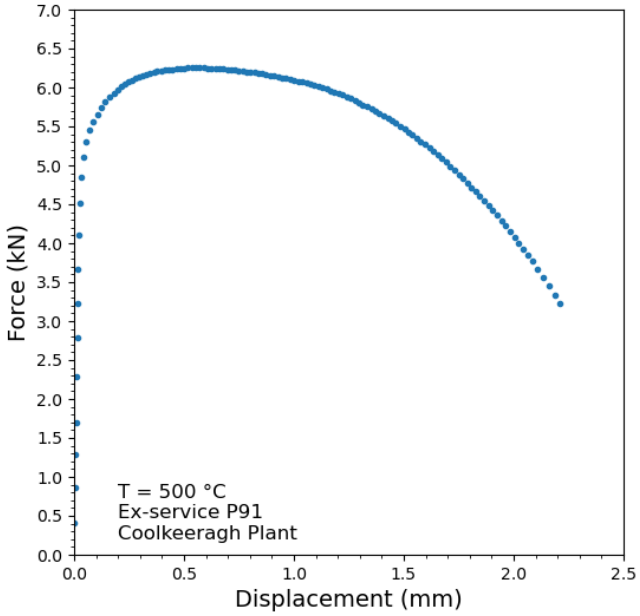


Figure 6.1, Experimental load-displacement curve for Coolkeeragh ex-service P91 steel

6.1. Mesh Sensitivity Analysis

Before applying the python script to the Abaqus problem, a mesh sensitivity study was conducted on the model as highlighted in Figure 6.2.

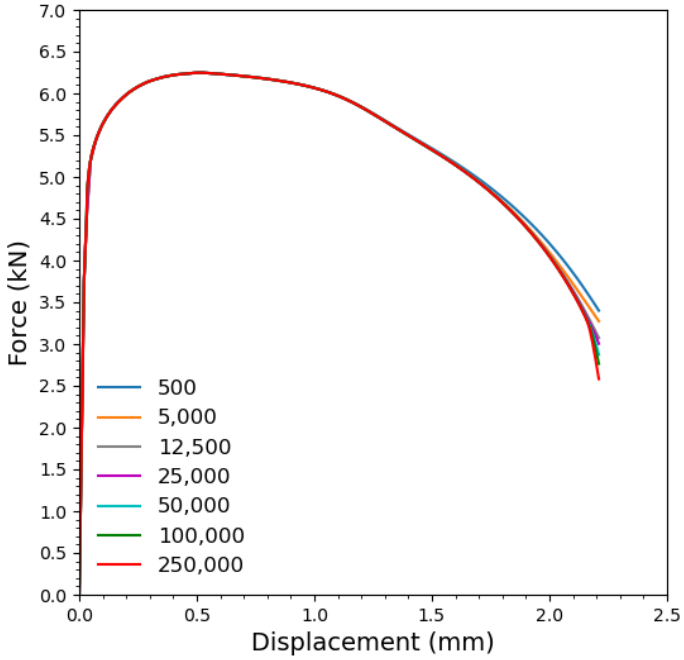


Figure 6.2, Mesh refinement study highlighting the impact of mesh size on the load-displacement results

From Figure 6.2, it is evident the mesh-density has minimal impact on the results up to a displacement of approximately 1.75 mm. However, comparing the various meshes at the maximum displacement of 2.21 mm, a significant variation in force is noted to occur. Therefore, it was decided to assess the meshes against the 250,000-element mesh at a displacement of 2.1 mm as outlined in Table 6.1. The 5,000-element mesh was deemed suitable for the model as it minimised the computational expense, whilst resulting in only a 3.11 % percentage difference between the 250,000-element mesh.

Table 6.1, Comparison of force value recorded at 2.1 mm for each mesh against the 250,000 mesh

Mesh	Force at 2.1 mm (kN)	% Difference 250,000 mesh
250,000	3.578	0
100,000	3.597	0.53
50,000	3.624	1.29
25,000	3.635	1.59
5,000	3.689	3.11
500	3.859	7.85

The low computational expense for a single simulation associated with applying the 5,000-element mesh to the model, partially questions the significance of applying a machine learning approach.

6.2. Parameter Sensitivity to Significant Figures Analysis

Before automating the parameter reselection process, a parameter sensitivity study was conducted on the seven GTN parameters and the slope of the extrapolated line as shown in Table 6.2. To assess the effect of each parameter, the final force value was recorded, all parameters were set as constant and the assessed parameter was varied in increments of 0.5 after the decimal place, for example 0.9 ,0.95, 0.955 and 0.9555. The percentage difference between the final force values for each significant figure configuration were compared against the value at four significant figures to assess the deviation as shown in Table 6.2.

Table 6.2, Parameter sensitivity study highlighting the variation in results due to the significant figures applied for the respective GTN parameters

Parameter	Number of significant figures after decimal point			
	4	3	2	1
	Force (kN)	% Difference between force value at 4 significant figures		
q_1	3.120	0.0641	0.545	4.90
q_2	3.171	0.0315	0.378	3.21
q_3	3.278	0.00	0.00	0.183
ε_N	3.283	0.0305	0.306	3.07
s_N	3.230	0.00	0.0309	1.33
f_N	3.196	0.0313	0.219	2.41
f	3.266	0.0612	0.214	0.398
Slope (MPa)	3.299	0.00	0.00	0.0220

From Table 6.2, the difference between four and three significant figures is negligible with the highest difference 0.0641 % and similarly for two significant figures the highest difference was 0.545 %. For one significant figure after the decimal point, the error was significantly larger particularly for parameters q_1 q_2 and ε_N with differences of 4.9 %, 3.213 % and 3.076 % respectively. Additionally, it was observed that the number of significant figures after the decimal place for the slope parameter had negligible impact on the final force value. Therefore, it was decided that two significant figures after the decimal place would be sufficient for each GTN parameter when inputting the parameters into the Abaqus model.

6.3. Identification of Optimum GTN Parameters

The results of the FFD and BO methods to identify the optimum material parameters are outlined in subsections 6.3.1 and 6.3.2 respectively and then compared in subsection 6.3.3.

6.3.1. Fractional Factorial Design

For the FFD approach described in Section 5.6.1, a total of 729 simulations were run which took 12 hrs 18 mins to complete. Only 438 simulations completed correctly and 291 either failed to reach the maximum experimental displacement or had an incorrect neck due to the selected material parameters. The variation in MAPE for the correctly completed simulations is displayed in Figure 6.3.

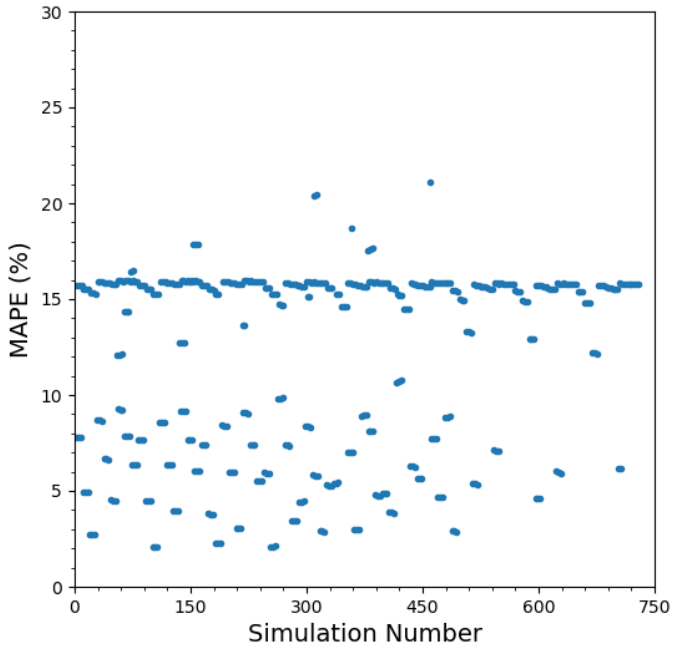


Figure 6.3, Variation in MAPE for correctly completed simulations based on the FFD approach

From the study, the FFD approach identified simulation 107 to have the lowest MAPE of 2.05 % which contained the GTN parameters displayed in Table 6.3. A graphical comparison between the optimum simulation and experimental load-displacement curves is shown in Figure 6.4.

Table 6.3, Optimum GTN parameters identified by FFD

q_1	q_2	q_3	ε_N	s_N	f_N	f	Slope (MPa)
0.90	1.00	0.86	0.25	0.10	0.09	0.0015	503.2

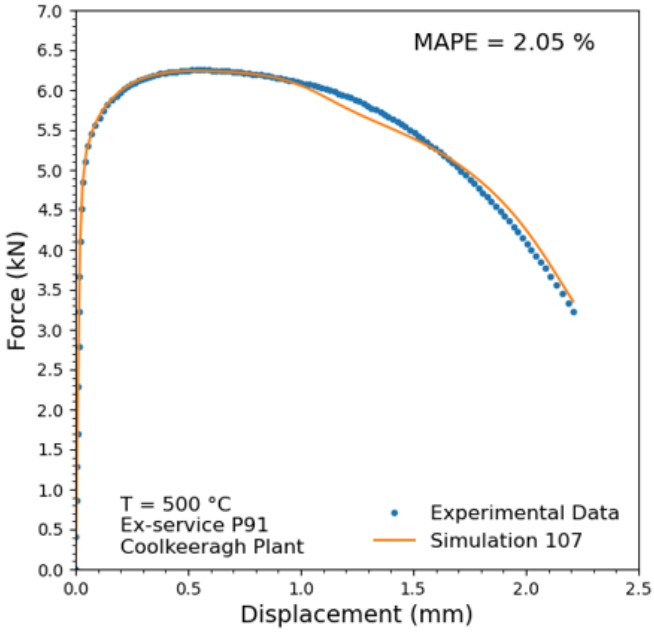


Figure 6.4, Load-displacement curve comparison between FFD optimum simulation and experimental data

From Figure 6.4, it is evident the optimum parameters identified by the FFD approach provide an excellent correlation to the experimental load-displacement curve in which only slight deviation is observed between approximately 1.1 mm and 1.6 mm.

6.3.2. Bayesian Optimisation

The BO algorithm was applied as detailed in Section 5.6.2, in which the total number of simulations was set to 55. Out of the 55 simulations, only 12 completed correctly in which a plot of MAPE against simulation number for the correctly completed simulations is evident in Figure 6.5.

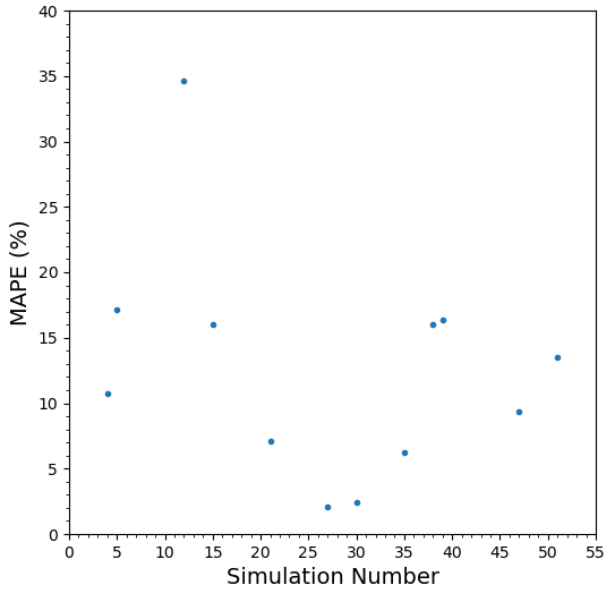


Figure 6.5, Variation in MAPE for correctly completed simulations based on the BO approach

From Figure 6.5, 78.2 % of simulations did not complete correctly in comparision to 39.9 % in the FFD approach shown in Figure 6.3. As a uniform error of 999 % is assigned to the simulation regardless of

the size of the incorrect neck or displacement reached with respect to the maximum experimental displacement, the BO algorithm cannot learn from previous errors and hence, an increase in the magnitude of incorrect simulations is seen. Figure A.4 in Appendix A illustrates the classification of a correct and incorrect neck formations on the Abaqus model.

Additionally, Figure 6.5 highlights that the algorithm does not converge on a final MAPE value, this could be due the high number of parameters that must be adjusted as each time the algorithm must reselect a parameter from the entire parameter space. However, it is also probable the algorithm does not converge due to the assignment of 999 % for simulations that do not complete correctly. The low errors of 2.10 % and 2.42 % occur at simulation 27 and 30 respectively, however simulations 28 and 29 do not complete correctly and therefore the MAPE of 999 % is assigned, which drastically increases the rate of change of MAPE directly impacting the convergence. The issue of non-convergence is discussed further in Section 8 highlighting possible methods to resolve the matter.

From Figure 6.5, at simulation 7 the BO algorithm locates a maximum MAPE of 34.6 % in which it adjusts the parameters and a decreasing MAPE trend is evident, which finds a local minimum MAPE of 2.10 % at simulation 27, however it cannot be said with certainty that this is the global minimum. The parameter reselection process of the algorithm is highlighted in Table C-1 in Appendix C. The optimum parameters corresponding to the lowest MAPE are displayed in Table 6.4. The difference between the maximum and minimum recorded MAPE values is shown in Figure 6.6(a) and a comparison between the experimental data and the optimum simulation is shown in Figure 6.6(b).

Table 6.4, Optimum GTN parameters identified by BO

q_1	q_2	q_3	ε_N	s_N	f_N	f	Slope (MPa)
0.90	0.90	2.56	0.25	0.20	0.03	0.0013	417.2

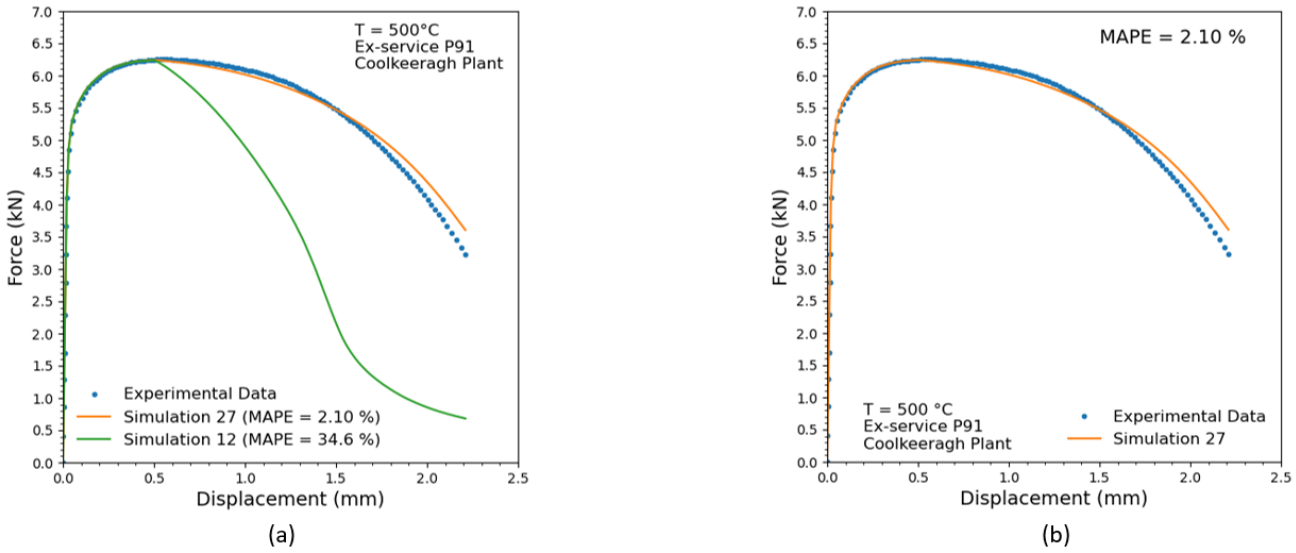


Figure 6.6, Load-displacement curve comparison between (a) maximum and minimum recorded MAPE simulation load-displacement curves (b) BO optimum simulation 27 and experimental data

From Figure 6.6 (a), it is clear the maximum MAPE of 34.6 % at simulation 12 provides a very poor correlation to the experimental curve in which huge deviation is evident from the UTS point onwards,

however the algorithm quickly learns from this error and within 15 additional simulations reaches the local minimum of 2.10 %. Figure 6.6 (b) shows the optimum simulation provides an excellent correlation to the experimental curve, with minimal deviation up to a displacement of 1.7 mm after which only slight deviation is observed, indicating good accuracy in the identified parameters.

6.3.3. Comparison of the Optimum Parameters Identified by FFD and BO

From sections 6.3.1 and 6.3.2, it is clear both the FFD and BO approaches were able to identify a low MAPE of 2.05 % and 2.10 % respectively. However, the main advantage of the BO approach is it only took 1 hr 6 mins to complete and identify the optimum parameters, whilst the FFD approach took 12 hrs 18 mins. Therefore, the BO method reduced the time by 91.1 % and computational expense by 92.5 %. Although the MAPE values of the respective optimum parameters are relatively close, significant variation occurs between the identified optimum parameters as shown in Table 6.5.

Table 6.5, Variation in the optimum parameters identified by the FFD and BO methods

	q_1	q_2	q_3	ε_N	s_N	f_N	f	Slope (MPa)	MAPE (%)
FFD	0.90	1.00	0.86	0.25	0.10	0.09	0.0015	503.2	2.05
BO	0.90	0.90	2.56	0.25	0.20	0.03	0.0013	417.2	2.10
% Diff.	0.0	10.5	99.4	0.0	66.7	100	14.3	18.7	2.41

The low percentage difference between the two MAPE values of 2.41 % highlighted in Table 6.5, despite the large difference in parameters indicates the solution to the analysis might not be unique for which there may be multiple solutions capable of achieving the global minimum MAPE.

6.4. Repeatability Test on BO Algorithm

To investigate the variation in the BO algorithm results, a repeatability test was conducted for which the algorithm was run five times for identical parameter ranges. The variation in MAPE versus simulation number for correctly completed simulations was analysed as shown in Figure 6.7.

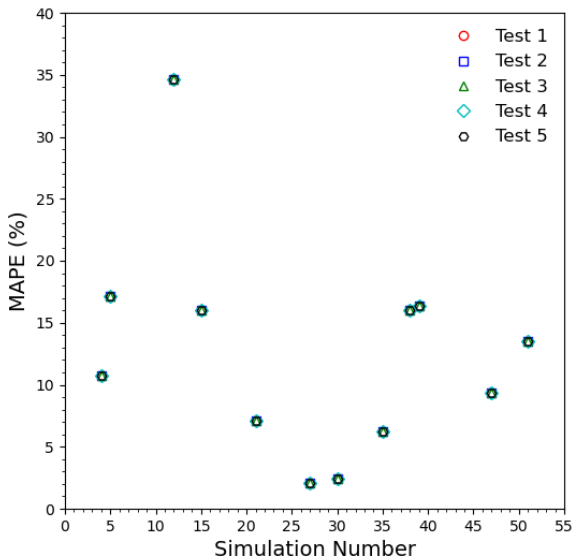


Figure 6.7, Variation in MAPE against simulation number for five identical BO test runs

From Figure 6.7, no variation in MAPE versus simulation number is present, indicating high repeatability in the BO reselection of parameters, for which the lowest MAPE and the corresponding optimum parameters for each test were identified in simulation 26 as highlighted in Table 6.6.

Table 6.6, Optimum parameters identified by each repeatability test

Test	q_1	q_2	q_3	ε_N	s_N	f_N	f	Slope (MPa)	MAPE (%)
1	0.90	0.90	2.56	0.25	0.20	0.03	0.0013	417.2	2.10
2	0.90	0.90	2.56	0.25	0.20	0.03	0.0013	417.2	2.10
3	0.90	0.90	2.56	0.25	0.20	0.03	0.0013	417.2	2.10
4	0.90	0.90	2.56	0.25	0.20	0.03	0.0013	417.2	2.10
5	0.90	0.90	2.56	0.25	0.20	0.03	0.0013	417.2	2.10

From Table 6.6, it is evident there is no variation between the lowest MAPE and corresponding optimum GTN parameters for each test, which indicates the BO algorithm will identify the same local minimum each time despite the issue of non-convergence. Therefore, the user can be confident that the results of the BO algorithm are highly reproducible.

6.5. Versatility of BO Algorithm

To test the versatility of the Bayesian optimisation algorithm, it was applied to additional experimental load-displacement data sets obtained from separate tensile tests on ex-service P91 steel from the Aghada power plant conducted by Kennedy (2018). As this material had a different service history (23,000 hours) and was tested at room temperature, it provides a contrasting load-displacement curve to the Coolkeeragh material (63,000 hrs) tested at 500 °C. The Aghada material was originally tested and then a repeat test was conducted in which significant deviation was observed between the load-displacement curves as highlighted in Figure 6.8.

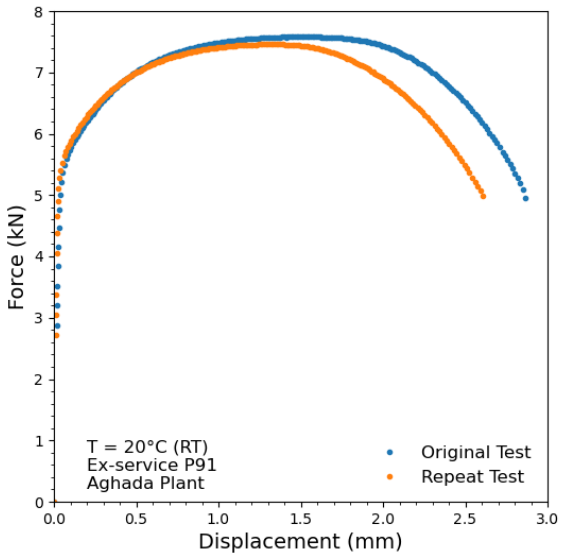


Figure 6.8, Experimental load-displacement curves for Aghada ex-service P91 steel

The scatter in the Aghada data provides an excellent opportunity to assess the effect of the experimental data on the identification of the optimum GTN parameters using BO.

6.5.1. Identification of Optimum GTN Parameters for Aghada Material

The BO algorithm was applied to the original Aghada material test data in which the number of simulations was set to 55 to correspond to the Coolkeeragh test. In the original test 14 out of 55 simulations completed correctly as shown in Figure 6.9 (a) and for the repeat test 15 out of 55 simulations completed correctly as shown in Figure 6.9 (b)

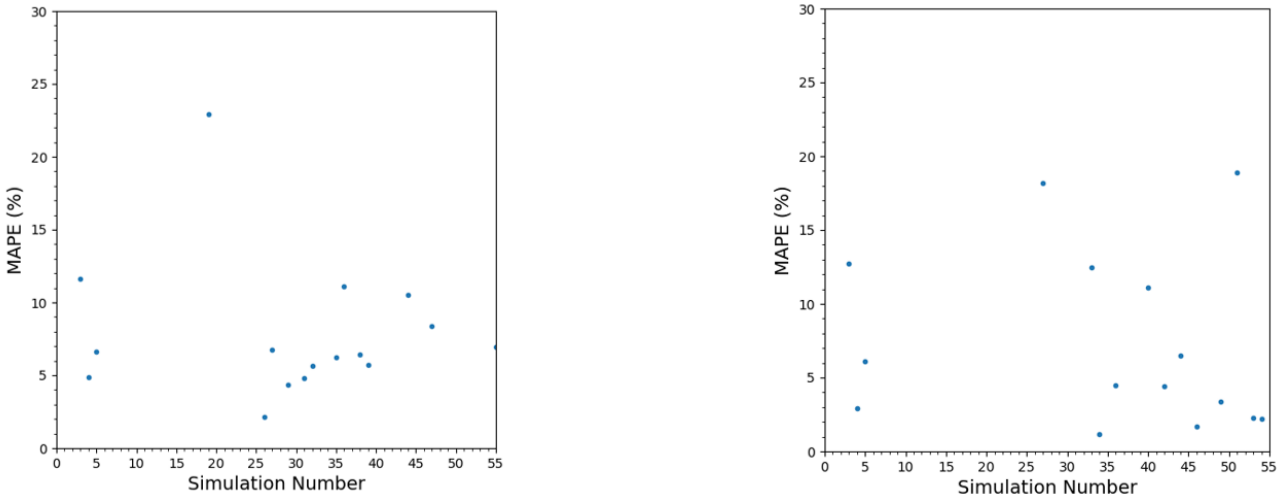


Figure 6.9, Variation in MAPE for completed simulations for Aghada material (a) original test (b) repeat test

Similar to the Coolkeeragh analysis, the BO algorithm did not converge for the Aghada material for either the original or repeat test. Despite non convergence, the algorithm identified a local minimum MAPE of 2.17 % at simulation 26 for the original data and a local minimum MAPE of 1.19 % at simulation 34 for the repeat test. Graphical comparisons of the identified optimum simulation against the experimental data for the original and repeat tests are shown in Figure 6.10 (a) and (b).

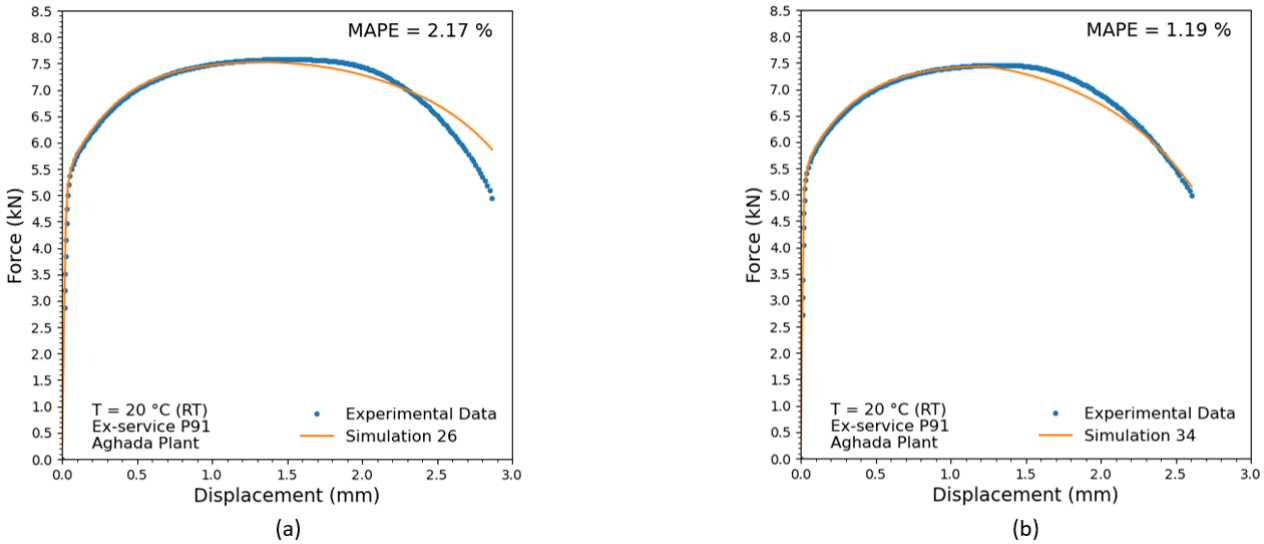


Figure 6.10, Load-displacement curve comparision between BO optimal simulation and the Aghada experimental data (a) original test (b) repeat test

From Figure 6.10, it is evident the BO algorithm was able to obtain a better fit to the repeat experimental data by achieving 0.98 % of a lower MAPE. The repeat test data shows excellent

correlation up to 1.5 mm in which slight deviation occurs due to the curve of the experimental data however, excellent correlation reoccurs from 2.3 mm till failure. Similarly, the original test data provides an excellent comparison up to approximately 1.5 mm, however after which significant deviation is evident particularly as the material tends to failure for which the simulation curve overpredicts the force considerably. A comparison of the optimum parameters applied to achieve the comparisons shown in Figure 6.10 (a) and (b) for each test is displayed in Table 6.7.

Table 6.7, Comparision of optimum parameters identified for the Aghada material based on original and repeat test experimental data

	q_1	q_2	q_3	ε_N	s_N	f_N	f	Slope (MPa)	MAPE (%)
Original	0.90	0.90	2.56	0.25	0.20	0.057	0.0014	662.03	2.17
Repeat	0.90	0.90	0.99	0.25	0.20	0.030	0.0013	499.36	1.19
% Diff.	0.0	0.0	88.5	0.0	0.0	62.1	7.41	28.0	58.3

From Table 6.7, the algorithm identifies the same values for parameters q_1, q_2, ε_N and s_N. Slight variation occurs for f with a percentage difference of 7.41%. In contrast, significant scatter occurs between the GTN parameters q_3, f_N, and the slope of the extrapolated data with percentage differences of 88.5 %, 62.1 % and 28.0 % respectively. The scatter in these GTN parameters is also reflected in the calculated MAPE resulting in a percentage difference of 58.3 %. Furthermore, the variation in parameters and MAPE indicates the scatter in the experimental data has a direct impact on the ability of the BO algorithm to determine a local minimum MAPE value and the identification of the corresponding optimum GTN parameters. The parameter reselection processes for the BO algorithm for the original and repeat tests are located in Table C-2 and Table C-3 in Appendix C.

7. Conclusions

From the completion of this work, the following conclusions can be drawn with respect to both the numerical analysis set-up and the application of machine learning to the analysis.

7.1. Numerical Analysis

- It was determined the load-displacement results obtained from modelling a tensile test are independent of a high-density mesh. A relatively low-density mesh of 5,000 elements is sufficient to match simulated data with experimental results. The low-density mesh reduces the computational time and expense required to run a single simulation, therefore the value of applying a machine learning approach in this situation is somewhat questionable.

- For the seven GTN parameters required to model ductile damage in an Abaqus simulation, it was determined that two significant figures after the decimal was sufficient to accurately model a specimen, with less than 0.55 % difference with respect to four significant figures after the decimal point.

- In addition to the seven GTN parameters, it was determined that the slope of the extrapolated true stress-plastic strain data had a direct impact on the load-displacement results beyond the UTS point and was included as a parameter to be optimised in modelling ductile damage.

7.2. Machine Learning

- The limitations of Abaqus software regarding python version 2.7.15 increases the complexity of incorporating a machine learning algorithm such as BO which requires python version 3.7 or later. To use python version 3.7, it was necessary to integrate the machine learning and Abaqus modules using a third software package PyCharm. Using python version 3.7 Abaqus simulations were submitted using the command line interface to pass simulation variables.

- The BO algorithm identified a local minimum MAPE of 2.10 % for the Coolkeeragh ex-service P91 steel tested at 500 °C by identifying the optimum GTN parameters for the material to be: q_1 =0.90, q_2= 0.90, q3 = 2.56 , ε_N = 0.25, s_N = 0.20, f_N = 0.03 , f =0.0013 and the extrapolated true stress-plastic strain data slope = 417.24 MPa.

- Despite establishing a local minimum MAPE, the algorithm showed a poor successful simulation completion rate in which 78.2 % of simulations did not reach the maximum experimental displacement or necked incorrectly. The high incompletion rate is likely due to the uniform MAPE error assignment of 999 %, which limits the learning of the algorithm and additionally impacts the ability of the algorithm to converge on a final MAPE value.

- The BO algorithm showed excellent repeatability for which zero variation was observed in the plot of MAPE versus simulation number for five tests. In addition, the algorithm successfully identified the same local minimum and corresponding optimum GTN parameters for each test. Therefore, the user can be confident in reproducing the results despite the non-convergence of the algorithm.

- The FFD approach identified a local minimum of 2.05 % after 729 simulations which took 12 hrs 18 mins to complete. In contrast the BO algorithm located a similar minimum MAPE of 2.10 % within 55 simulations which required only 1 hr 6 mins. Therefore, the BO algorithm reduced the required computational time by 91.1 % and the number of simulations by 92.5 %, whilst maintaining the accuracy of the solution.

- The BO and FFD solutions achieved almost identical MAPE values, however significant differences were observed between the identified optimum parameters. The large variation in optimal parameters despite minimal difference in the calculated MAPE may indicate that the global minimum MAPE solution is not unique and that multiple parameter combinations could achieve similar values of MAPE.

- The application of the BO algorithm to additional experimental datasets obtained from testing Aghada ex-service P91 steel at room temperature indicated good versatility in the BO algorithm. The BO algorithm located minimum MAPE values of 2.17 % and 1.19 % for the original and repeat tests respectively, despite similar issues regarding high incorrect simulation completion rate and non-convergence that were previously encountered.

8. Future Work

For future work in this area, to improve the convergence of the BO algorithm, the following recommendations are advised

1. Incorporate sequential domain reduction to the Bayesian optimisation algorithm. Sequential domain reduction reduces the size of the initial GTN parameter range after each simulation based on the calculated MAPE. Including sequential domain reduction, and decreasing the search range on every iteration, may significantly reduce the time to convergence (Mongan, *et. al.* 2022).

2. The impact of assigning a MAPE of 999 % should be addressed more comprehensively. Alternatively, the geometry and meshing of the simulation could be modified to ensure that necking occurs only at the expected location, thereby reducing the frequency of assigning an error MAPE value.

    a. Adjust the assigned error for an incomplete simulation depending on the magnitude of an incorrect neck or the simulated displacement reached with respect to the maximum experimental displacement. This would improve the BO algoritihm by providing an insight into how close the simulation was to completing correctly rather than just assigning a uniform value of 999 % and hence, reduce the number of simulations required for convergence.

    b. Create a 2D axisymmetric model of the full specimen shown in Figure 4.1 to include the grip section rather than just the gauge area. Modelling only the gauge area results in a sharp corner on the top of the model which possibly accounts for incorrect necks. The inclusion of the grip results in a smooth transition from the top of the gauge area and should prevent a neck from forming on the top of the specimen.

    c. Similarly, as a relatively low mesh is required for the 2D approach, the model could be extended to 3D to investigate if the issue of incorrect neck formations is still present. In addition, the 3D and 2D approaches should be compared with regards to the required computational time and the accuracy of the load-displacement results compared to the experimental data.

References

Abaqus (2021a) ‘Abaqus, Scripting, Using the Abaqus Scripting Interface, Introduction to Python’ Dassault Systèmes, Providence, RI, USA.

Abaqus (2021b) ‘Abaqus, Materials, Inelastic Material Properties, Metal plasticity, Porous metal plasticity.’ Dassault Systèmes, Providence, RI, USA.

Abbassi, F., Belhadj, T., Mistou, S. and Zghal, A. (2013) ‘Parameter identification of a mechanical ductile damage using Artificial Neural Networks in sheet metal forming’, *Materials & Design*, 45, pp. 605–615. doi:10.1016/j.matdes.2012.09.032.

Abendroth, M. and Kuna, M. (2006) ‘Identification of ductile damage and fracture parameters from the small punch test using neural networks’, *Engineering Fracture Mechanics*, 73(6), pp. 710–725. doi:10.1016/j.engfracmech.2005.10.007.

Attaran, M. and Deb, P. (2018) ‘Machine learning: the new “big thing” for competitive advantage’, *Machine learning*, International Journal of Knowledge Engineering and Data Mining, 5(4) pp. 277-300. doi: 10.1504/IJKEDM.2018.10015621.

Brassington, G. (2017) 'Mean absolute error and root mean square error: which is the better metric for assessing model performance' *EGU General Assembly,* 19(1), p. 3574. availible at: 2017EGUGA..19.3574B

Bonora , N. (1997) 'A nonlinear CDM model for ductile failure'. *Journal of Engineering Fracture Mechanics,* 58(1), pp. 11-28. doi: https://doi.org/10.1016/S0013-7944(97)00074-X

Chahboub, Y. and Szavai, S. (2019) ‘Determination of GTN parameters for SENT specimen during ductile fracture’, in *Procedia Structural Integrity*. Elsevier B.V., pp. 81–88. doi:10.1016/j.prostr.2019.07.025.

Chen, Y. (2019) ‘Modeling of ductile fracture using local approach: reliable simulation of crack extension’, Mechanics [physics.med-ph]. Université Paris sciences, pp. 5-12.

Deisenroth, M.P., Faisal, A.A. and Ong, C.S. (2020) *Mathematics for Machine Learning*. Cambridge University Press. Available at: https://mml-book.com/ (Accessed: 23 October 2021).

Gilmore, C.M. (2015) *Materials science and engineering properties*. SI edition. Australia ; Stamford, CT: Cengage Learning.

Gurson, A.L. (1977) ‘Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I—Yield Criteria and Flow Rules for Porous Ductile Media’, *Journal of Engineering Materials and Technology*, 99(1), pp. 2–15. doi:10.1115/1.3443401.

Hadi, M.N.S. and Bodhinayake, B.C. (2003) ‘Non-linear finite element analysis of flexible pavements’, *Advances in Engineering Software*, 34(11–12), pp. 657–662. doi:10.1016/S0965-9978(03)00109-1

Kennedy, D. (2018) 'Experimental testing of high chromium martensitic steels for fossil fuel plant applications', unpublished thesis (B.E.), University of Limerick.

Kiran, R. and Khandelwal, K. (2014) ‘Gurson model parameters for ductile fracture simulation in ASTM A992 steels', *Fatigue & Fracture of Engineering Materials & Structures*, 37(2), pp. 171–183. doi:10.1111/ffe.12097.

Lavocat, J.-C. (2015) ‘Active photonic devices based on liquid crystal elastomers’, Ph.D. in Photonic Science Physics Department p9. 18-20. doi:10.13140/RG.2.1.1053.4644

Liu, Y., Zhao, T., Ju, W. and Shi, S. (2017) ‘Materials discovery and design using machine learning’, *Journal of Materiomics*, 3(3), pp. 159–177. doi:10.1016/j.jmat.2017.08.002.

Liu, X., Athanasiou, C.E., Padture, N.P., Sheldon, B.W. and Gao, H (2020) ‘A machine learning approach to fracture mechanics problems’, *Acta Materialia*, 190, pp. 105–112. doi:10.1016/j.actamat.2020.03.016.

Malvern, L. E., 1969. 'Introduction to the mechanics of a continuous medium'. New Jersey: Prentice-Hall, Inc..

McClintock, F.A. (1968) ‘A Criterion for Ductile Fracture by the Growth of Holes’, *Journal of Applied Mechanics*, 35(2), pp. 363–371. doi:10.1115/1.3601204.

Meade, E.D., Sun, F., Tiernan, P., O’Dowd, N.P. (2021) ‘A multiscale experimentally-based finite element model to predict microstructure and damage evolution in martensitic steels’, *International Journal of Plasticity*, 139. doi:10.1016/j.ijplas.2021.102966.

Mongan, P.G., Hinchy, E.P., O’Dowd, N.P. and McCarthy*, C.T* (2021) ‘Quality prediction of ultrasonically welded joints using a hybrid machine learning model’, *Journal of Manufacturing Processes*, 71, pp. 571–579. doi:10.1016/j.jmapro.2021.09.044.

Mongan, P.G., Modi, V., McLaughlin, J.W., Hinchy, E.P., O'Higgins, R.M., O'Dowd, N.P., McCarthy, C.T. (2022) ‘Multi-objective optimisation of ultrasonically welded dissimilar joints through machine learning’, *Journal of Intelligent Manufacturing*, 33(4), pp. 1125–1138. doi:10.1007/s10845-022-01911-6.

Monostori, L. (2002) ‘AI AND MACHINE LEARNING TECHNIQUES FOR MANAGING COMPLEXITY, CHANGES AND UNCERTAINTIES IN MANUFACTURING’, *IFAC Proceedings Volumes*, 35(1), pp. 119–130. doi:10.3182/20020721-6-ES-1901.01644.

Moreno, M.F. (2016) ‘Application of small punch testing on the mechanical and microstructural characterizations of P91 steel at room temperature’, *International Journal of Pressure Vessels and Piping*, 142–143, pp. 1–9. doi:10.1016/j.ijpvp.2016.04.002.

Nelli, F. (2018). '*Python Data Analytics'.* 2nd ed. New York: Apress.

O’Mahoney, D.C., Katnam, K.B., O'Dowd, N.P., McCarthy, C.T., Young, T.M. (2013) ‘Taguchi analysis of bonded composite single-lap joints using a combined interface–adhesive damage model’, *International Journal of Adhesion and Adhesives*, 40, pp. 168–178. doi:10.1016/j.ijadhadh.2012.06.001.

Paermentier, B., Debruyne, D. and Talemi, R. (2021) 'A machine learning based sensitivity analysis of the GTN damage parameters for dynamic fracture propagation in X70 pipeline steel', *International Journal of Fracture*, 227(1), pp. 111–132. doi:10.1007/s10704-020-00499-3.

Rice, J.R. and Tracey, D.M. (1969) 'On the ductile enlargement of voids in triaxial stress fields∗', *Journal of the Mechanics and Physics of Solids*, 17(3), pp. 201–217. doi:https://doi.org/10.1016/0022-5096(69)90033-7.

Slimane, A., Bouchouicha, B., Benguediab, M., (2015) 'Parametric study of the ductile damage by the Gurson–Tvergaard–Needleman model of structures in carbon steel A48-AP', *Journal of Materials Research and Technology*, 4(2), pp. 217–223. doi:10.1016/j.jmrt.2014.12.011.

Tu, S., Ren, X., He, J. and Zhang, Z., (2020) 'Stress–strain curves of metallic materials and post-necking strain hardening characterization: A review', *Fatigue & Fracture of Engineering Materials & Structures*, 43(1), pp. 3–19. doi:10.1111/ffe.13134.

Tvergaard, V. and Needleman, A. (1984) 'Analysis of the cup-cone fracture in a round tensile bar', *Acta Metallurgica*, 32(1), pp. 157–169. doi:10.1016/0001-6160(84)90213-X.

Tvergaard, V., Needleman, A. and Lo, K.K. (1981) 'Flow localization in the plane strain tensile test', *Journal of the Mechanics and Physics of Solids*, 29(2), pp. 115–142. doi:http://dx.doi.org/10.1016/0022-5096(81)90019-3.

Wu, B., Zhou, J., Ji, X., Yin, Y. and Shen, X. (2019) 'Research on Approaches for Computer Aided Detection of Casting Defects in X-ray Images with Feature Engineering and Machine Learning', *Procedia Manufacturing*, 37, pp. 394–401. doi:https://doi.org/10.1016/j.promfg.2019.12.065.

Zhong, J., Xu, T., Guan, K., Szpunar, J. (2019) 'A procedure for predicting strength properties using small punch test and finite element simulation', *International Journal of Mechanical Sciences*, 150, pp. 238 -235. doi: https://doi.org/10.1016/j.ijmecsci.2019.01.006

Appendices

Appendix A. Supplementary Graphs and Diagrams

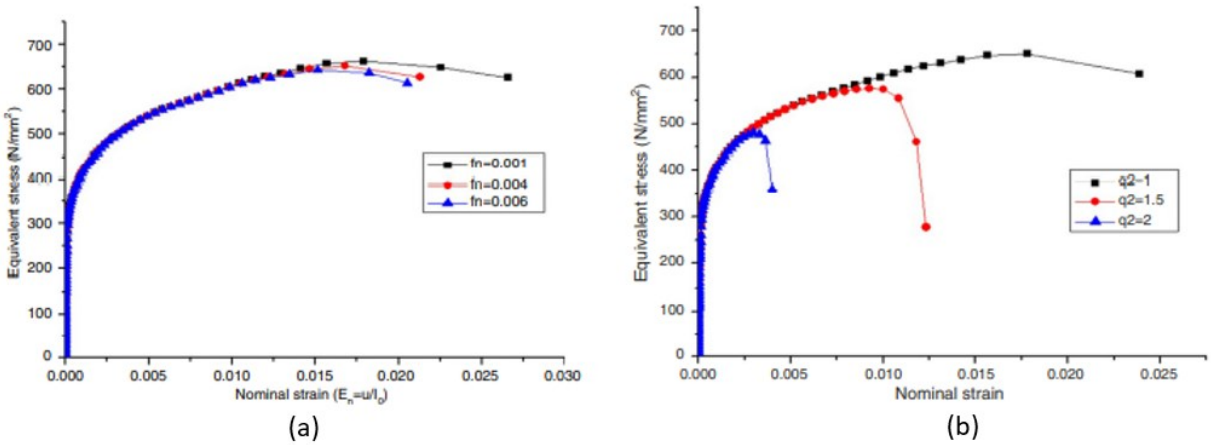


Figure A.1, The impact of adjusting a single parameter on the resulting simulated stress-strain curve (a) f_N (b) q_2 (Slimane *et al.*, 2015).

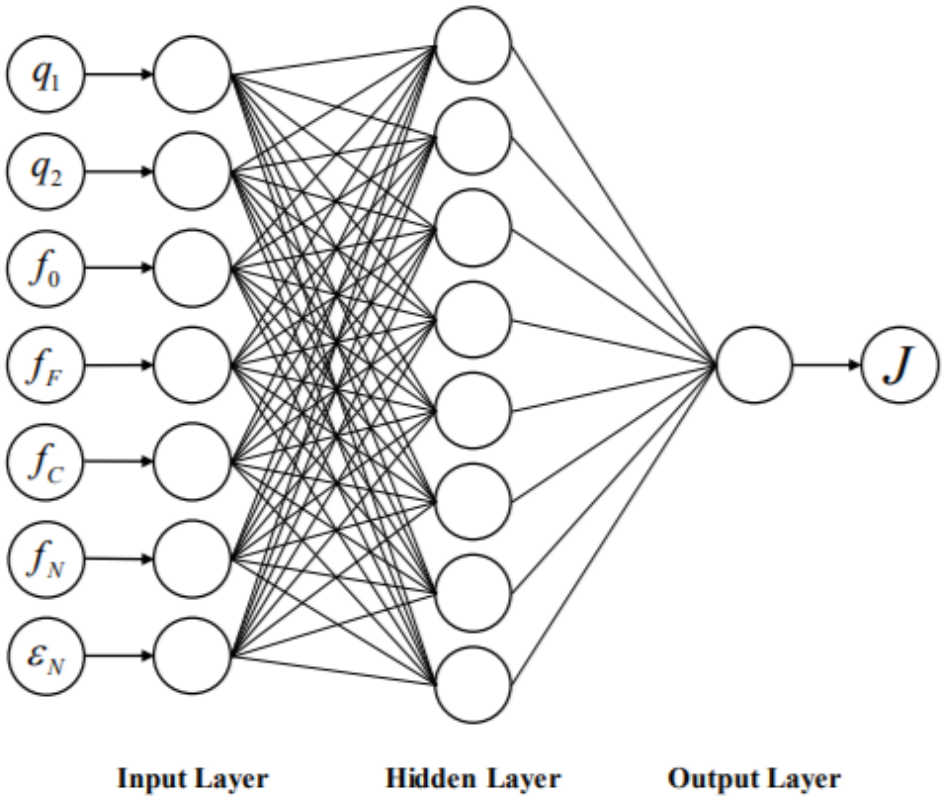
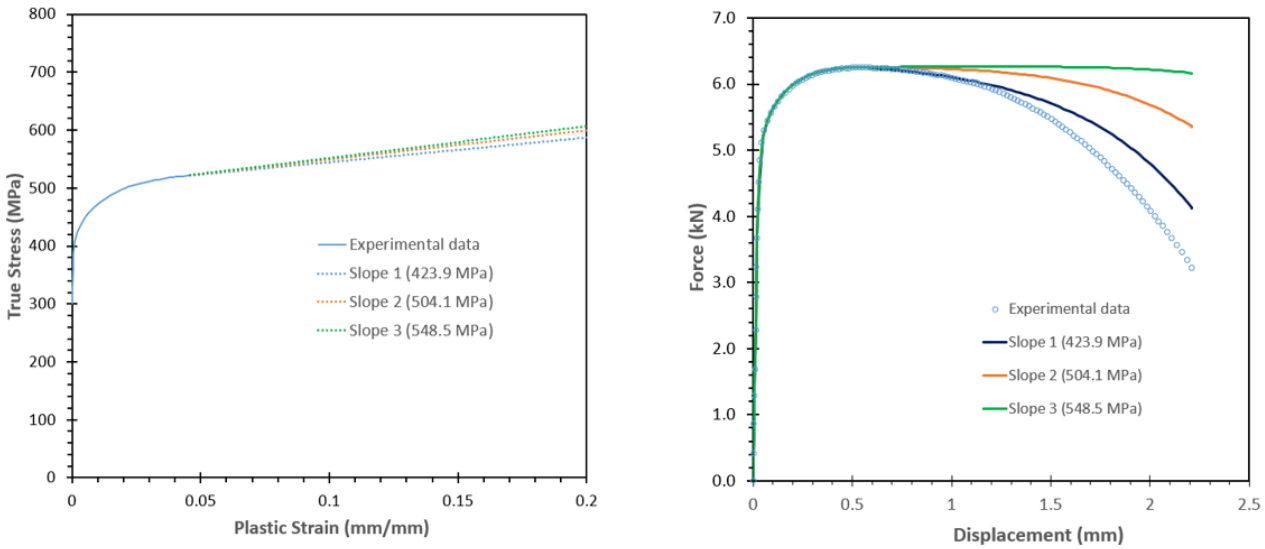
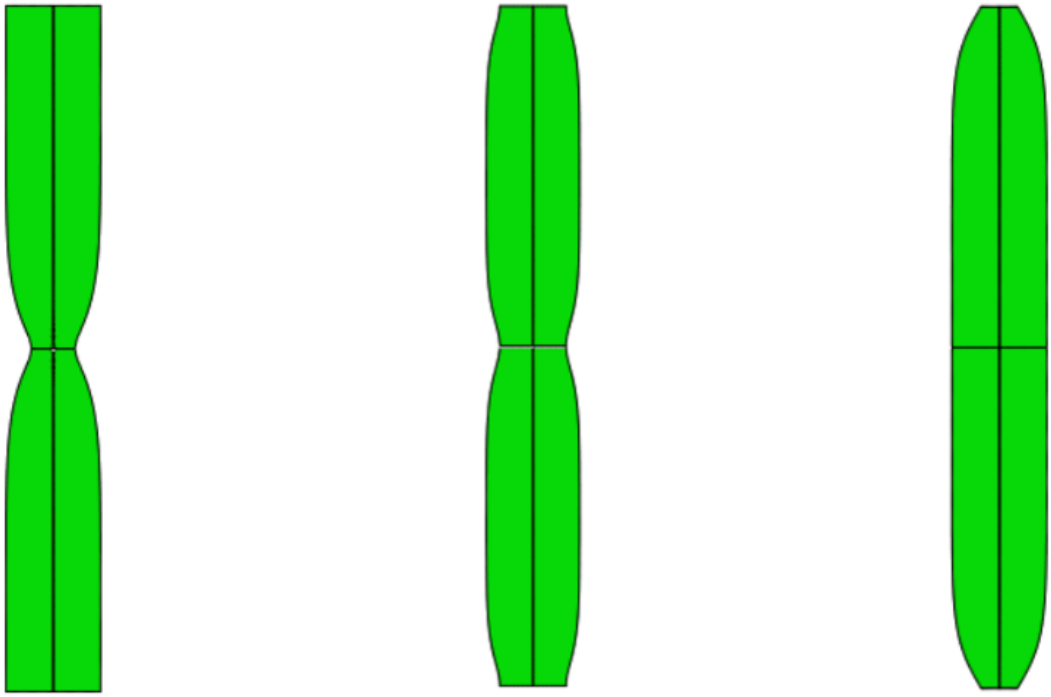


Figure A.2, Sample artificial neural network for GTN parameter identification (Paermentier *et al.*, 2021)



(a) (b)

Figure A.3, (a) The variation in the slope of the extrapolated data (b) the impact of the variation in slope on the load-displacement curve without the inclusion of the GTN damage model.



(a) (b) (c)

Figure A.4: Different neck configuration results for Abaqus model (a) correct neck at location of inserted notch (b) incorrect neck due to formation of neck at top and bottom of model in addition to neck at centre (c) incorrect neck due to formation of neck at top and bottom of model

Appendix B. Operating Manual for Designed Software

1. Open folder “Basic_BO” and ensure all python files are located within the working directory in addition to an empty folder titled “FEA OUTPUT” and an empty csv file titled “Bayesian Outputs” as shown in Figure B.1.

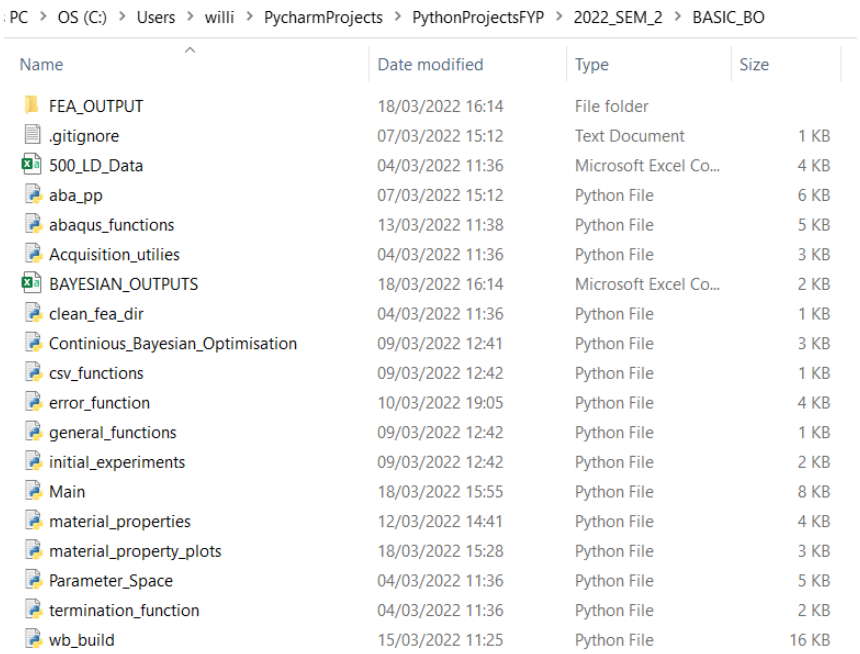
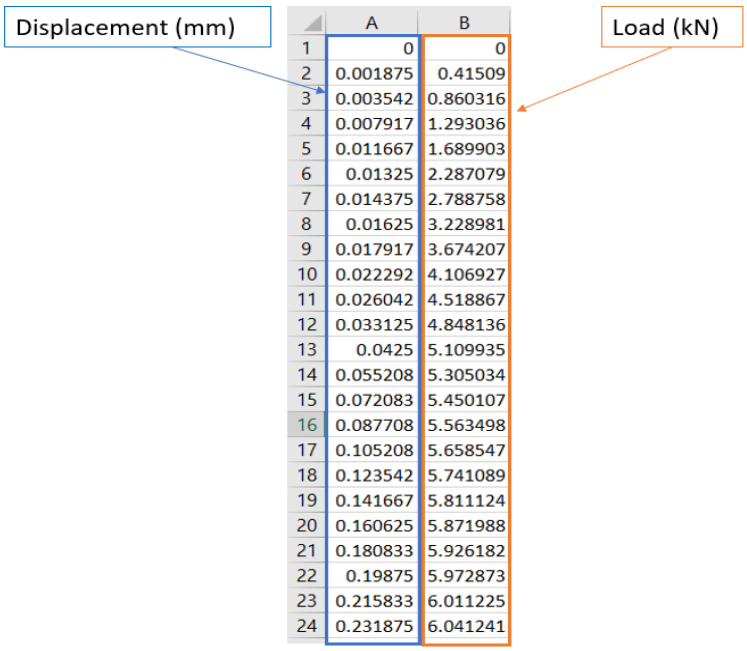


Figure B.1, Working directory highlighting the required python and csv files

2. Ensure the experimental load-displacement data “500_LD_Data” is in the working directory saved in a csv file in which the load values are in kN and displacement values in mm formatted a shown in Figure B.2.



	A	B
1	0	0
2	0.001875	0.41509
3	0.003542	0.860316
4	0.007917	1.293036
5	0.011667	1.689903
6	0.01325	2.287079
7	0.014375	2.788758
8	0.01625	3.228981
9	0.017917	3.674207
10	0.022292	4.106927
11	0.026042	4.518867
12	0.033125	4.848136
13	0.0425	5.109935
14	0.055208	5.305034
15	0.072083	5.450107
16	0.087708	5.563498
17	0.105208	5.658547
18	0.123542	5.741089
19	0.141667	5.811124
20	0.160625	5.871988
21	0.180833	5.926182
22	0.19875	5.972873
23	0.215833	6.011225
24	0.231875	6.041241

Figure B.2, Required layout of experimental data in csv file

3. Open PyCharm and then open the python file titled “Main.py” in the main window by double clicking the file as shown by the red rectangle in Figure B.3. When the file is open right click the Main tab on the top of the screen and select “Run Main”

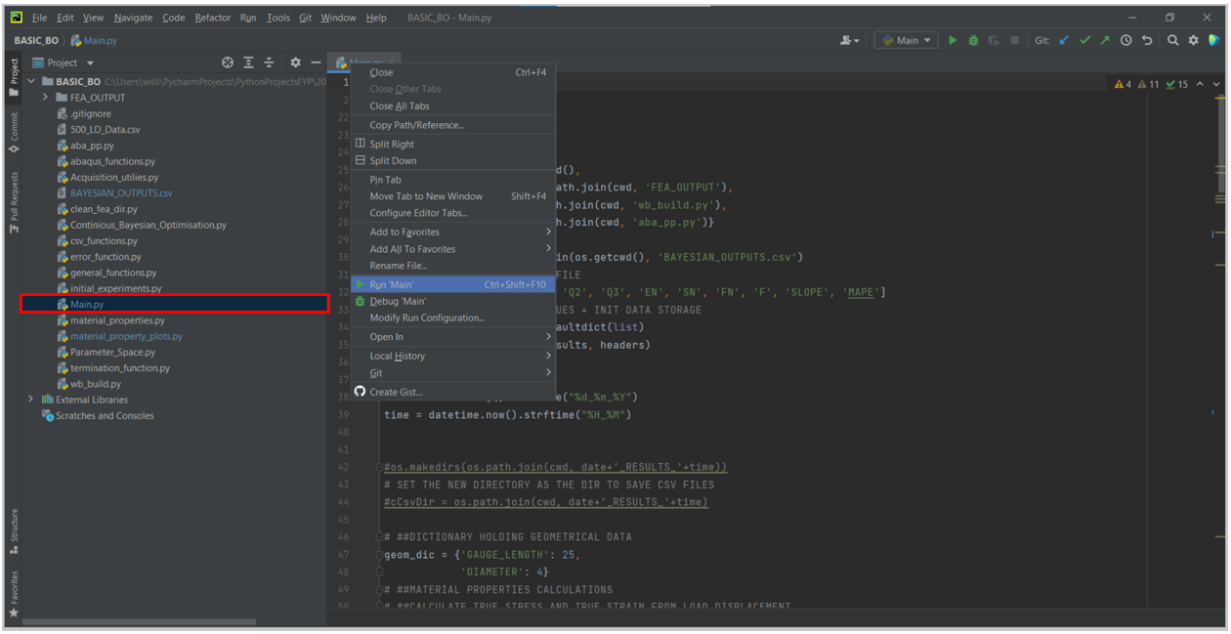


Figure B.3, PyCharm home screen highlighting how to run the Main.py file

4. An image of the converted true stress and strain appears on the screen, if the user is happy with the conversion of the data, click the x in the red box on the top right corner of the image as shown in Figure B.4

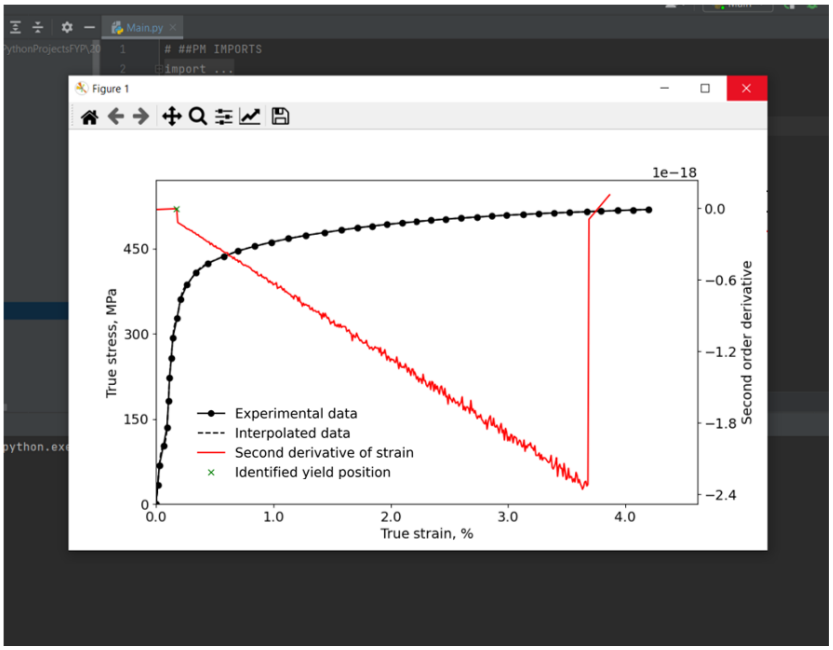


Figure B.4, True stress and strain graph that appears before the code starts to run FEA simulations

5. The code outputs the yield stress and Young’s modulus and verifies the first simulation has started in the PyCharm command window as shown by the red rectangle in Figure B.5.

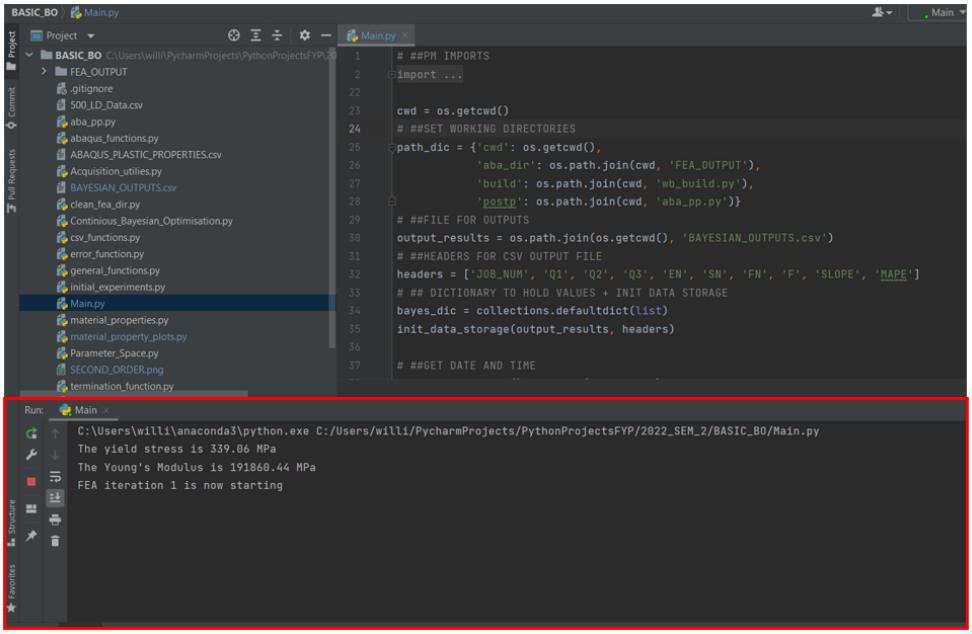


Figure B.5, Message in PyCharm command window indicating the code has started to run FEA simulations

6. As each simulation runs the Abaqus files are outputted into the FEA Output folder titled as JOB plus simulation number for example 'JOB1' as shown in Figure B.6. In addition, on completion of the simulation the load-displacement data is saved to the "FEA Output" folder and titled according to the JOB title as highlighted by the red rectangle in Figure B.6.

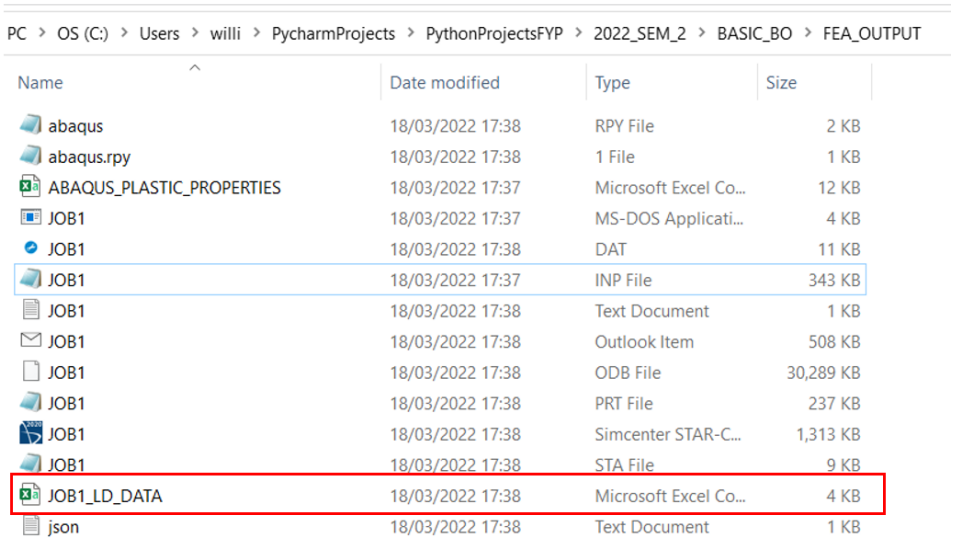


Figure B.6, FEA Output folder highlighting the Abaqus files and results from the simulation

7. In the command window a message appears informing the user the simulation has completed and provides the time taken as shown in Figure B.7.

```
FEA iteration 4 is now starting
JOB4 has completed now post-processing
JOB4 post-processing complete, abaqus analysis took 63.25528907775879 seconds
```

Figure B.7, Message in PyCharm command window indicating simulation completed and the required time

8. If a simulation does not complete correctly a message will appear in the command window informing the user whether the simulation did not reach the maximum displacement (green rectangle) or if it did not neck correctly (yellow rectangle) as shown in Figure B.8

```
FEA iteration 2 is now starting
JOB2 has completed now post-processing
Simulation did not neck sufficiently, diameter was estimated at 2.72347640991 mm
JOB2 post-processing complete, abaqus analysis took 73.29185056686401 seconds
FEA iteration 3 is now starting
JOB3 has completed now post-processing
Simulated displacement was 1.93 mm, compared to the experimental value of 2.21 mm
JOB3 post-processing complete, abaqus analysis took 81.34741449356079 seconds
```

Figure B.8, Error message in PyCharm command window for simulations that do not complete correctly

9. The code repeats steps 6-8 until either the maximum number of simulations is reached or the code has reached a converged solution for which a message saying the code exited appears in the command window as shown in Figure B.9.

```
Code has exited
```

Figure B.9, Message in pycharm command window indicating the code has finished

10. To access the results the user must open the Bayesian outputs csv file which highlights the JOB number, the corresponding GTN parameters and the MAPE for each simulation. The layout of the csv file is shown in Figure B.10 which makes it easy for the user to identify which simulation had the lowest MAPE and the corresponding parameters.

	A	B	C	D	E	F	G	H	I	J
1	JOB_NUM	Q1	Q2	Q3	EN	SN	FN	F	SLOPE	MAPE
2	1	1.25	1.00	1.69	0.33	0.15	0.060	0.0014	0.0	999
3	2	1.25	1.00	1.69	0.33	0.15	0.060	0.0014	186.9	999
4	3	1.25	1.00	1.69	0.33	0.15	0.060	0.0014	373.8	999
5	4	1.25	1.00	1.69	0.33	0.15	0.060	0.0014	560.7	10.74
6	5	1.25	1.00	1.69	0.33	0.15	0.060	0.0014	747.6	17.16
7	6	1.57	1.01	1.19	0.34	0.20	0.044	0.0014	186.5	999
8	7	1.60	1.10	0.81	0.40	0.20	0.030	0.0015	285.5	999
9	8	0.9	1.10	2.56	0.35	0.12	0.090	0.0015	90.8	999
10	9	1.24	1.00	2.41	0.36	0.20	0.064	0.0014	331.6	999
11	10	1.20	0.94	1.21	0.36	0.13	0.043	0.00150	235.6	999
12	11	1.12	0.93	2.49	0.36	0.12	0.038	0.0014	137.5	999
13	12	0.98	1.00	0.96	0.30	0.19	0.042	0.0014	42.7	34.62
14	13	0.9	1.10	2.56	0.40	0.20	0.090	0.0013	430.9	999
15	14	1.6	0.90	0.81	0.25	0.10	0.030	0.0015	473.0	999
16	15	1.6	0.90	2.56	0.25	0.10	0.030	0.0015	654.5	16.03
17	16	1.53	1.08	1.60	0.34	0.17	0.061	0.0014	402.0	999
18	17	1.6	1.10	0.81	0.25	0.10	0.090	0.0015	113.0	999
19	18	1.08	0.97	1.63	0.28	0.11	0.069	0.0014	260.5	999
20	19	0.9	1.10	2.56	0.25	0.10	0.090	0.0015	162.2	999
21	20	1.19	1.10	1.73	0.26	0.17	0.038	0.0014	212.9	999
22	21	1.10	0.94	1.10	0.35	0.13	0.087	0.0015	453.5	7.066
23	22	1.6	1.10	2.56	0.25	0.10	0.030	0.0013	269.0	999
24	23	1.6	0.90	2.56	0.40	0.10	0.090	0.0013	250.9	999
25	24	1.6	0.90	0.81	0.25	0.10	0.030	0.0015	498.5	999
26	25	1.59	1.02	1.45	0.28	0.14	0.080	0.0014	308.7	999
27	26	0.9	1.10	0.81	0.40	0.17	0.030	0.0014	352.8	999
28	27	0.9	0.90	2.56	0.25	0.20	0.030	0.0013	417.2	2.095

Figure B.10, Layout of Bayesian Outputs csv file showing the parameters and MAPE values for each simulation up to the optimum simulation 27.

Appendix C. Bayesian Optimisation Parameter Reselection Process

Table C.1 , Bayesian optimisation results for Coolkeeragh P91 steel tested at 500 °C highlighting identified local minimum MAPE

JOB	q_1	q_2	q_3	ε_N	s_N	f_N	f	SLOPE (MPa)	MAPE (%)
1	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	0.0	999
2	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	186.9	999
3	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	373.8	999
4	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	560.7	10.7
5	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	747.6	17.2
6	1.57	1.01	1.19	0.34	0.20	0.04	0.0014	186.5	999
7	1.60	1.10	0.81	0.40	0.20	0.03	0.0015	285.5	999
8	0.90	1.10	2.56	0.35	0.12	0.09	0.0015	90.8	999
9	1.24	1.00	2.41	0.36	0.20	0.06	0.0014	331.6	999
10	1.20	0.94	1.21	0.36	0.13	0.04	0.0015	235.6	999
11	1.12	0.93	2.49	0.36	0.12	0.04	0.0014	137.5	999
12	0.98	1.00	0.96	0.30	0.19	0.04	0.0014	42.7	34.6
13	0.90	1.10	2.56	0.40	0.20	0.09	0.0013	430.9	999
14	1.60	0.90	0.81	0.25	0.10	0.03	0.0015	473.0	999
15	1.60	0.90	2.56	0.25	0.10	0.03	0.0015	654.5	16.0
16	1.53	1.08	1.60	0.34	0.17	0.06	0.0014	402.0	999
17	1.60	1.10	0.81	0.25	0.10	0.09	0.0015	113.0	999
18	1.08	0.97	1.63	0.28	0.11	0.07	0.0014	260.5	999
19	0.90	1.10	2.56	0.25	0.10	0.09	0.0015	162.2	999
20	1.19	1.10	1.73	0.26	0.17	0.04	0.0014	212.9	999
21	1.10	0.94	1.10	0.35	0.13	0.09	0.0015	453.5	7.07
22	1.60	1.10	2.56	0.25	0.10	0.03	0.0013	269.0	999
23	1.60	0.90	2.56	0.40	0.10	0.09	0.0013	250.9	999
24	1.60	0.90	0.81	0.25	0.10	0.03	0.0015	498.5	999
25	1.59	1.02	1.45	0.28	0.14	0.08	0.0014	308.7	999
26	0.90	1.10	0.81	0.40	0.17	0.03	0.0014	352.8	999
27	0.90	0.90	2.56	0.25	0.20	0.03	0.0013	417.2	2.10
28	1.60	1.10	0.81	0.40	0.10	0.03	0.0015	229.9	999
29	1.60	1.10	2.56	0.40	0.10	0.09	0.0013	231.2	999
30	0.90	1.10	0.81	0.40	0.20	0.03	0.0013	388.8	2.42
31	1.36	0.96	1.94	0.28	0.11	0.08	0.0015	485.1	999
32	0.90	0.90	0.81	0.40	0.10	0.09	0.0013	149.8	999
33	0.95	0.93	1.72	0.38	0.18	0.03	0.0013	199.7	999
34	1.57	0.97	2.34	0.26	0.17	0.03	0.0013	363.5	999
35	0.94	1.04	2.43	0.35	0.11	0.09	0.0013	514.6	6.26
36	1.60	0.90	2.56	0.25	0.10	0.09	0.0013	218.8	999
37	1.60	1.10	2.56	0.25	0.10	0.09	0.0015	219.6	999
38	0.90	1.10	2.56	0.40	0.20	0.03	0.0013	607.6	16.0
39	1.60	0.90	0.81	0.25	0.20	0.03	0.0015	701.0	16.4
40	1.60	0.90	2.56	0.25	0.10	0.09	0.0015	224.7	999
41	1.60	1.10	0.81	0.25	0.10	0.09	0.0013	225.4	999
42	1.60	1.10	0.81	0.25	0.20	0.09	0.0015	74.7	999
43	1.12	1.04	1.21	0.34	0.12	0.09	0.0014	124.5	999
44	1.49	1.03	1.54	0.28	0.18	0.05	0.0014	15.1	999
45	1.60	1.10	2.56	0.25	0.10	0.09	0.0013	201.9	999
46	1.42	1.01	1.65	0.33	0.17	0.05	0.0014	174.8	999
47	0.90	0.90	2.56	0.40	0.10	0.03	0.0013	297.1	9.38
48	0.90	1.10	0.81	0.25	0.20	0.09	0.0015	319.6	999
49	0.99	1.01	1.46	0.27	0.17	0.05	0.0014	101.1	999
50	1.42	1.04	2.40	0.40	0.12	0.08	0.0013	342.3	999
51	0.98	1.04	1.77	0.38	0.19	0.04	0.0013	277.4	13.5
52	1.60	1.10	0.81	0.25	0.10	0.09	0.0015	163.5	999
53	1.60	1.10	0.81	0.25	0.10	0.09	0.0015	163.8	999
54	1.09	1.07	1.31	0.34	0.18	0.08	0.0015	63.0	999
55	0.95	1.09	1.74	0.32	0.14	0.08	0.0013	83.0	999

Table C.2 , Bayesian optimisation results for Aghada P91 steel original test at room temperature highlighting identified local minimum MAPE

JOB	q_1	q_2	q_3	ε_N	s_N	f_N	f	SLOPE (MPa)	MAPE (%)
1	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	0.0	999
2	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	264.8	999
3	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	529.7	11.6
4	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	794.5	4.87
5	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	1059.4	6.62
6	1.20	1.06	1.85	0.33	0.12	0.08	0.0013	265.4	999
7	1.54	1.10	2.56	0.40	0.10	0.09	0.0015	133.6	999
8	1.44	1.01	1.99	0.31	0.16	0.09	0.0013	200.6	999
9	1.20	0.94	1.17	0.26	0.14	0.07	0.0014	64.1	999
10	1.39	0.96	2.55	0.32	0.12	0.04	0.0014	232.6	999
11	1.31	0.98	1.96	0.38	0.19	0.06	0.0014	98.4	999
12	1.25	0.90	0.81	0.25	0.10	0.09	0.0013	167.1	999
13	0.90	1.10	0.81	0.40	0.20	0.09	0.0013	31.0	999
14	1.54	1.01	1.04	0.35	0.16	0.09	0.0013	250.3	999
15	1.41	1.09	0.92	0.29	0.15	0.04	0.0014	116.1	999
16	1.14	0.95	2.36	0.33	0.19	0.05	0.0013	184.9	999
17	1.40	0.98	2.50	0.35	0.17	0.06	0.0014	81.2	999
18	1.37	0.91	2.34	0.39	0.15	0.03	0.0013	46.4	999
19	0.99	1.07	1.09	0.31	0.13	0.06	0.0015	149.6	22.9
20	1.60	0.90	2.56	0.25	0.20	0.09	0.0013	0.0	999
21	1.60	0.90	0.81	0.25	0.20	0.03	0.0015	310.3	999
22	1.60	0.90	2.56	0.40	0.10	0.09	0.0013	0.0	999
23	0.90	1.10	2.56	0.40	0.10	0.09	0.0015	357.2	999
24	1.60	0.90	0.81	0.28	0.20	0.06	0.0015	401.5	999
25	1.56	0.94	0.99	0.38	0.18	0.06	0.0013	440.7	999
26	0.90	0.90	2.56	0.25	0.20	0.06	0.0014	662.0	2.17
27	0.90	0.90	0.81	0.40	0.10	0.09	0.0015	927.0	6.73
28	1.60	0.90	0.81	0.40	0.20	0.03	0.0015	0.0	999
29	0.90	0.90	2.56	0.40	0.10	0.03	0.0014	471.0	4.38
30	1.60	0.90	0.81	0.25	0.20	0.03	0.0013	333.8	999
31	1.15	0.99	2.47	0.33	0.14	0.06	0.0014	595.3	4.80
32	1.60	0.90	0.81	0.40	0.10	0.03	0.0013	727.9	5.63
33	0.90	1.10	2.56	0.40	0.10	0.09	0.0015	421.9	999
34	1.60	1.10	2.56	0.25	0.20	0.03	0.0013	860.3	999
35	1.60	1.10	2.56	0.25	0.20	0.03	0.0013	993.6	6.25
36	0.91	0.98	1.19	0.38	0.16	0.08	0.0013	379.2	11.1
37	1.37	1.02	1.07	0.39	0.15	0.04	0.0014	289.1	999
38	0.90	1.10	2.56	0.40	0.20	0.03	0.0013	882.6	6.40
39	1.60	1.10	2.56	0.25	0.20	0.03	0.0013	839.0	5.75
40	1.14	1.09	0.91	0.33	0.17	0.03	0.0013	217.5	999
41	1.29	0.91	0.93	0.33	0.11	0.03	0.0014	14.9	999
42	1.23	0.91	1.45	0.36	0.12	0.06	0.0014	345.7	999
43	0.96	1.01	2.09	0.27	0.18	0.09	0.0014	321.2	999
44	0.92	1.03	0.86	0.26	0.19	0.06	0.0015	411.5	10.5
45	1.60	0.90	2.56	0.40	0.20	0.09	0.0013	0.0	999
46	1.60	1.10	2.56	0.25	0.10	0.09	0.0013	0.0	999
47	0.99	0.91	1.68	0.36	0.11	0.06	0.0015	431.0	8.39
48	1.60	1.10	2.56	0.40	0.10	0.09	0.0015	0.0	999
49	1.15	0.98	2.04	0.35	0.20	0.06	0.0014	278.5	999
50	1.01	0.92	1.78	0.36	0.17	0.07	0.0014	299.6	999
51	0.91	0.97	2.30	0.36	0.14	0.08	0.0015	450.5	999
52	1.60	1.10	0.81	0.40	0.10	0.09	0.0013	175.9	999
53	1.36	1.06	2.35	0.31	0.18	0.08	0.0013	761.2	999
54	1.60	1.10	0.81	0.40	0.10	0.09	0.0015	628.7	999
55	0.90	0.90	0.81	0.40	0.10	0.03	0.0015	960.3	6.92

Table C.3 , Bayesian optimisation results for Aghada P91 steel repeat test at room temperature highlighting identified local minimum MAPE

JOB	q_1	q_2	q_3	ε_N	s_N	f_N	f	SLOPE (MPa)	MAPE (%)
1	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	0.0	999
2	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	199.8	999
3	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	399.5	12.8
4	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	599.3	2.92
5	1.25	1.00	1.69	0.33	0.15	0.06	0.0014	799.0	6.12
6	1.20	1.06	1.85	0.33	0.12	0.08	0.0013	200.2	999
7	0.90	1.10	2.56	0.25	0.10	0.09	0.0013	100.6	999
8	1.44	1.01	1.99	0.31	0.16	0.09	0.0013	151.3	999
9	1.37	1.08	0.84	0.31	0.19	0.05	0.0013	49.9	999
10	1.39	0.96	2.55	0.32	0.12	0.04	0.0014	175.4	999
11	1.05	0.93	2.37	0.37	0.19	0.04	0.0014	23.7	999
12	1.05	0.91	1.02	0.30	0.11	0.03	0.0015	125.2	999
13	1.49	1.03	0.85	0.33	0.19	0.05	0.0013	75.4	999
14	1.54	1.01	1.04	0.35	0.16	0.09	0.0013	188.8	999
15	0.90	1.10	2.52	0.38	0.18	0.03	0.0015	63.5	999
16	1.14	0.95	2.36	0.33	0.19	0.05	0.0013	139.4	999
17	0.98	1.02	2.13	0.31	0.12	0.03	0.0015	38.3	999
18	1.46	0.99	2.51	0.28	0.14	0.08	0.0015	88.9	999
19	1.20	1.02	0.87	0.27	0.11	0.08	0.0014	11.0	999
20	1.52	1.01	2.41	0.27	0.20	0.07	0.0014	113.7	999
21	1.03	1.08	1.30	0.27	0.18	0.07	0.0015	164.1	999
22	1.19	0.96	1.07	0.34	0.17	0.04	0.0013	213.7	999
23	1.44	1.06	1.30	0.29	0.11	0.06	0.0015	207.2	999
24	1.53	0.91	1.04	0.37	0.17	0.06	0.0014	107.0	999
25	1.46	1.00	0.87	0.32	0.13	0.07	0.0013	28.8	999
26	1.45	0.96	2.54	0.29	0.19	0.06	0.0014	132.0	999
27	0.92	0.94	0.84	0.29	0.14	0.04	0.0013	82.7	18.2
28	1.60	1.10	2.56	0.40	0.20	0.09	0.0015	0.0	999
29	1.60	1.10	2.56	0.40	0.10	0.03	0.0013	0.0	999
30	1.60	1.10	2.56	0.25	0.20	0.09	0.0015	0.0	999
31	1.60	1.10	2.56	0.40	0.10	0.09	0.0015	0.0	999
32	1.60	1.10	2.56	0.25	0.10	0.09	0.0015	236.8	999
33	0.90	0.90	0.81	0.40	0.20	0.08	0.0015	257.9	12.5
34	0.90	0.90	0.99	0.25	0.20	0.03	0.0013	499.4	1.19
35	1.60	1.10	2.56	0.32	0.17	0.09	0.0014	699.2	999
36	0.90	0.90	0.81	0.25	0.10	0.03	0.0013	675.6	4.47
37	1.60	1.10	2.56	0.40	0.20	0.09	0.0014	720.7	999
38	1.46	1.10	0.81	0.25	0.10	0.09	0.0015	743.5	999
39	1.60	1.10	2.56	0.25	0.10	0.03	0.0013	328.9	999
40	0.90	0.90	0.81	0.40	0.20	0.09	0.0014	308.3	11.1
41	1.33	0.91	2.27	0.29	0.15	0.06	0.0014	347.3	999
42	0.90	1.10	2.56	0.25	0.20	0.05	0.0014	449.4	4.42
43	1.60	0.90	2.56	0.40	0.10	0.09	0.0015	549.3	999
44	0.90	0.90	2.56	0.40	0.20	0.03	0.0015	761.5	6.48
45	1.60	1.10	0.81	0.40	0.20	0.04	0.0013	364.7	999
46	0.90	1.10	0.81	0.25	0.20	0.03	0.0013	565.5	1.71
47	1.60	1.07	2.56	0.40	0.10	0.09	0.0015	533.8	999
48	1.60	1.10	2.56	0.40	0.20	0.09	0.0013	637.1	999
49	1.37	1.05	1.18	0.37	0.15	0.08	0.0014	733.1	3.40
50	1.60	1.10	2.56	0.40	0.20	0.09	0.0015	710.3	999
51	1.06	1.07	1.25	0.31	0.19	0.07	0.0014	226.2	18.9
52	1.54	1.01	1.02	0.40	0.11	0.07	0.0014	337.9	999
53	1.16	1.10	1.92	0.29	0.12	0.07	0.0013	649.3	2.24
54	1.18	1.08	1.57	0.32	0.13	0.03	0.0013	626.2	2.23
55	1.13	1.07	2.29	0.30	0.12	0.06	0.0015	356.3	999