Faculty of Science and Engineering

Neural Network-Based Constitutive Modelling of Inelastic Materials

REEM ALHAYKI

A thesis submitted to Swansea University in fulfilment of the requirements for the Degree of Doctor of Philosophy.

September 2023

Declarations

Statement 1

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed:.....Reem Alhayki.....

Date:.....26 Sep 2023.....

Statement 2

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed:.....Reem Alhayki.....

Date:.....26 Sep 2023.....

Statement 3

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed:.....Reem Alhayki.....

Date:.....26 Sep 2023.....

Statement 4

The University's ethical procedures have been followed and, where appropriate, that ethical approval has been granted.

Signed:.....Reem Alhayki.....

Date:.....26 Sep 2023.....

Abstract

New materials that are associated with novel micro-structures are becoming available at a rapid speed. The mechanical characteristics of solid materials are studied using constitutive models. The formulation of constitutive relations is challenging and may be related with the limiting hypothesis. Machine learning approaches have been widely used to simulate material behaviour in recent years. This study presents a neural network-based approach to reproduce the complex constitutive relations of solid materials including the modelling of rate-independent, elastic or inelastic, isotropic or anisotropic material behaviours. The approach can follow a history-based or an internal variables-based strategy. The network is developed using a generic internal variable formalism, with the number of internal variables determined by the nature of the problem and the degree of accuracy desired. It is demonstrated that the history-based and internal variable-based techniques accurately describe the von Mises elastoplastic material model in uniaxial stress condition. However, a thorough examination indicates that the internal variable-based strategy is most suited.

In this work, the network is trained on data sequences of strains and corresponding stresses that can be generated with physical experiments or numerical simulations based on Representative Volume Element (RVE). The used training strategy is based on gradient-free optimisation. The model performance is evaluated on different numerical examples including uniaxial damage mechanics and multiaxial elastoplasticity under plane strain conditions. The trained model is compared against the corresponding constitutive model to evaluate the accuracy. An efficient criterion for the verification of thermodynamic consistency is proposed and applied to the trained stress update models.

The proposed strategy is tested with numerical examples that represent relations of porous solid materials. The training data are generated with numerical multi-scale homogenisation based on an RVE composed of the von Mises elastoplastic matrix with an arbitrary void volume fraction. The RVE simulation was performed by enforcing appropriate RVE boundary conditions, and the stress responses are computed for various macro strain data sequences to generate different loading paths. The neural network based stress update algorithm is trained and validated. The obtained results show the ability of the proposed procedure in describing the material stress/strain relationship with high accuracy. In addition, plots of stress paths in the hydrostatic-deviatoric space show accurate results when compared against the corresponding Gurson model for porous elastoplastic material.

Acknowledgements

I would like to express my deepest gratitude to my supervisors Prof. Wulf Dettmer and Prof. Djordje Perić for their invaluable guidance, insightful comments, and unwavering support throughout my PhD journey.

I am also grateful to my colleague Eugenio Muttio for his collaboration and contribution to the research presented in this thesis. His knowledge and expertise have been invaluable to the completion of this work.

I would like to dedicate this work to my parents, sister and friends for their unconditional love, patience, and support throughout my academic journey. Their sacrifices and encouragement were the driving force that kept me going during the toughest times.

I also dedicate this work to my husband for being a source of motivation to me since the start.

Contents

List of Tables

List of Figures

List of Abbreviations

BVP Boundary Value Problem

FEM Finite Element Method

FFNN FeedForward Neural Network

GRU Gated Recurrent Unit

LSTM Long Short-Term Memory

MSE Mean Squared Error

MOPSO Multi-Objective Particle Swarm Optimization

NARX Nonlinear Autoregressive Network with Exogenous Input

PSO Particle Swarm Optimization

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

RVE Representative Volume Element

VEGP Vanishing/Exploding Gradients Problem

List of Publications

Some of the research leading to this thesis has appeared previously in the following publications.

Alhayki R, Muttio EJ, Dettmer WG and Perić D. *On the Performance of Different Architectures in Modelling Elastoplasticity with Neural Network.* World Congress on Computational Mechanics 2022, WCCM-APCOM2022.(2022). https://www.scipedia.com/public/Alhayki_et_al_2022a.

Dettmer WG, Muttio EJ, Alhayki R, Perić D. *A framework for neural network based constitutive modelling of inelastic materials.* Computer Methods in Applied Mechanics and Engineering. 2024 Feb 15;420:116672.

Alhayki R, Muttio EJ, Dettmer WG and Perić D. 2023 'A Neural Network Based Strategy for the Modelling of Constitutive Behaviour of Solids Applied to Porous Elastoplastic Material', *COMPLAS 2023: XVII International Conference on Computational Plasticity*, Barcelona, 5-7 September.

Muttio EJ, Alhayki R, Dettmer WG, Perić D and Fletcher L. *A Neural Network-Based Surrogate Model for Inelastic Solid Materials Simulations* [in preparation] " " 2023.

Chapter 1

Introduction

With the advancements in manufacturing and material engineering fields, new types of materials and micro-structures are becoming available at a fast rate. Constitutive models are used to understand material characteristics. The formulation of constitutive models is difficult and it is challenging to describe the complex nonlinear behaviors sufficiently [45, 44].

The recent developments in computer architecture and computational power made it possible to leverage novel machine learning techniques to model the behavior of materials and understand the associated physical properties [114, 2, 9]. Within the field of continuum materials mechanics numerous machine learning approaches are already applied successfully to enable, accelerate and simplify the discovery and development of novel materials for future deployment. Referring to [9], in the context of materials mechanics the type of machine learning can be classified as

- *Descriptive:* patterns within data can be recognised based on correlations, trends or anomalies to address questions like "*What factors contribute to the formation of microstructure Y with properties Z under the influence of process parameter X, and how do these elements impact material performance aspects like fatigue and failure?*". For example, in [63] a deep network was used for modeling mechanical material properties on multiple scales, based on homogenisation of two-dimensional RVE.
- *Predictive:* approximations based on available knowledge to foresee specific results generated by certain factors for answering questions like "*How does changing process parameter X affect the microstructure*

Y and subsequently influence the properties Z?". For example, in [110] the effective stiffness of high contrast elastic composites is predicted using a deep learning approach.

- *Prescriptive:* to not only identify and predict but also to implement optimised results with respect to improved actions answering questions like "*How can we change process parameters X to achieve microstructure Y with desired properties Z?*". A framework was proposed by [12] for capturing and communicating critical information related to the material structure evolution in multi-scale simulations to reduce the number of required experiments.

In particular, Artificial Neural Networks (ANNs) are able to process large amounts of data and estimate complex nonlinear relationships. The ANN can be implemented with the help of advanced open-source libraries such as TensorFlow [1] and PyTorch [79].

In the context of constitutive modeling, ANNs have the capability to learn the stress-strain behavior of materials with given examples of stress-strain increments [38, 53]. These increments can be obtained by performing physical experiments or numerical simulations. The data-set used by the model to learn is referred to as the training data-set, with each individual example being called a training sequence or sample [9]. The generation of data with random loading paths in the strain space similar to the stochastic process have been suggested by [70, 40, 109]. As it can be advantageous to include multiple stress-strain histories which help covering a wider training space. After training, the model should be validated to ensure it generalises well on unseen data (*i.e.* assess that the model is not underfitted or overfitted).

Acknowledging this potential, an expanding body of literature demonstrates the successful utilisation of ANNs in constitutive modeling for nonlinear materials. Examples include: hyper-elasticity [90, 57, 75], visco-elasticity [15], elastoplasticity [72, 28, 49, 42, 111, 105, 35], visco-plasticity [50, 32], cyclic plasticity [112, 101, 31], traction-separation [29], continuum damage [61], interface mechanics [107, 29], tensile damage in flexure [54] and rate-dependent materials [53, 51]. All these examples involve training and tuning a neural network with experimental or simulated data then using it to predict the desired material behavior.

Although proved useful, standard feed-forward neural networks struggled to model materials that involve history-dependent state variables. To overcome this challenge, a Recurrent Neural Network (RNN) can be used which includes memory elements to store previous material state information. However, a common difficulty associated with training RNNs are the Vanishing and Exploding Gradient Problem (VEGP) when the network weights become very small or very large respectively during back-propagation [78, 82, 5]. In order

to avoid this issue, special networks known as Long/Short Term Memory (LSTMs) [46] and Gated Recurrent Unit (GRUs) [17] can be used. LSTM-based networks have been used to model the homogenised mechanical response of polycrystals [30], fiber-reinforced composites [16] and visco-elastic behaviour [15]. On the other hand, GRU-based networks have demonstrated their ability to capture path-dependent plasticity, including intricate phenomena like homogeneous anisotropic hardening [40, 72]. Moreover, Mozaffar et al. [72] and Gorji et al. [40] extended the application of GRUs to plastic composites and metamaterials. Alternatively, several works in the literature addressed VEGP by using gradient free optimisation techniques [84, 87, 20, 106, 76, 108, 95].

For more complex materials, multiscale homogenisation is used to describe the behavior of the overall system based on information derived from small scale computational models. In various fields, extensive research has been devoted to homogenization-based multi-scale analysis and the selection of suitable Representative Volume Elements (RVEs) [34, 55, 113, 80, 81]. The work in [9, 114] shown that a neural network model framework have a potential to be used for creating homogenised elastoplastic constitutive model based on the results from RVE analysis (e.g. polycrystalline microstructures, composites or mechanical metamaterials).

In this work the RVE is subjected to random loading to generate numerical training data. The objective of this is to evaluate that the proposed methodology captures the yield criteria for porous materials with the elastoplastic von Mises matrix. The obtained results by the network are then compared to the corresponding classical Gurson model, which proposes an analytical yield criteria for porous media [41].

The main objective of this study is to demonstrate the challenges and opportunities in applying ANN models to the constitutive modeling of composite materials. The proposed strategy for network architecture follows internal variables formalism. Recent research have used RNN, LSTM, and GRU architectures for constitutive modelling, and the findings have been accurate (e.g. [11]). Both this study and the work conducted by Bonatti C and Mohr D [11] acknowledge the importance of using randomly generated datasets to create a more representative learning space for the network, yet there is a notable difference between the nature of the data. The datasets generated by [11] are mostly composed of elastic behaviour, which is comparatively easier for the network to capture. In addition, the used architecture in [10] requires great number of trainable parameters, ranging from 5684 to 12,000 parameters, in contrast to the method proposed in this work, which uses much less parameters. Moreover, when testing gated network with long sequences it was observed that the models suffer from memory loss. The proposed architecture overcomes this issue, enabling accurate

processing of extended sequences.

Additionally, since the model does not include the basic laws of thermodynamics, the predictions of trained network should be checked to make sure the solution does not violate these laws. A possible solution is to encode the laws of thermodynamics in the architecture of the model [66]. In this study while the laws are not encoded in the network, the proposed strategy is checked to ensure that it is thermodynamically consistent by evaluating a closed path in the strain space consisting of two finite steps (*i.e.* advancing step and backward step). Moreover, to verify the model compliance with physical constraints in multi-axial case the mean pressure p and the deviatoric equivalent stress q is included during the training process, which gives an additional measurable accuracy coefficient to check with physical constraints.

The use of suitable strategies based on the capabilities of neural networks can bridge the gap between experimental data and finite element analysis. Several successfully integrated ANNs as material characterisation routines within FE codes are found in the literature such as [59, 58, 88]. The integration of the model in FE code is beyond the scope of this work.

1.1 Problem and Motivation

Advances in added layer manufacturing and materials engineering are enabling creation of new material types and micro-structures at rapid pace. New classes of materials often require the formulation of new classes of models, each with a suitable state-space representation. Examples of different engineering materials and the corresponding conventional physics-based modeling approach are presented in Table 1.1.1.

Table 1.1.1: Examples of classes of engineering materials, each requiring a different model and implementation when taking the conventional physics-based modeling approach [10].

Material	**Model**
Low-carbon steel	Von Mises yields isotropic hardening
Dual-phase steel	Quadratic yields kinematic hardening
Metal foam	Deshhpande-Fleck yields non-associated hardening
Filled rubber	Hyper-elasticity Mullins damage
Glassy polymer	Boyce-Parks-Argon model

The motivation of this work is to develop a generalised strategy to model stress updates using neural networks. The proposed approach can be applied in various situations, including the following two scenarios:

- *Numerical stress update procedure based on physical experiments:* Data is generated in a series of physical experiments in which a sample material is subjected to different load paths. The generated data is then utilised to train a neural network model based on the stress update procedure. Following that, the trained model can be integrated into finite element simulations directly. Hence, allowing efficient and accurate analysis of real-world applications.

- *Numerical homogenisation:* A suitable numerical strategy, for example RVE [80], is used to represent the complex micro structure of a material. Then a series of numerical experiments are performed to compute the stress responses on a macro level for several strain data input sequences, *i.e.* load paths. The stress-strain data obtained based on these numerical experiments is utilised to train the proposed stress update model, which can be applied in finite element analysis to analyse real-world applications.

In both given scenarios, the proposed methodology bridges the gap between experiments and finite element analysis, eliminating the complicated and tedious formulation of an accurate constitutive model and its algorithmic implementation.

1.2 Objectives

The main highlights and key aspects of this work can be summarised as

- Developing a novel neural network based algorithmic constitutive model which can be applied to generic rate-independent inelastic solid material behaviour. Two training approaches have been proposed which are history based and internal variable strategies.
- The algorithm can be trained on stress and strain data sequences generated from physical experiments or numerical homogenisation based on RVEs.
- Recovering the piecewise linear constitutive behaviour including uniaxial elastoplasticity accurately by the strategy.
- Presenting and testing a novel training framework for the recurrent network architecture based on gradient-free optimisation by employing the multi-objective particle swarm optimisation strategy.
- The training data are generated carefully with random loading paths to cover a wider training space.
- Formulating a numerical testing method based on two finite steps to ensure thermodynamic consistency of the trained constitutive models.

- Testing the proposed methodology on numerical examples for uniaxial damage mechanics and multi-axial elastoplasticity under plane strain conditions.
- Assuming that the material behaviour can be described fully in p-q space, the performance of the network in terms of hydrostatic pressure and the norm of the deviatoric stress (*i.e.* $p - q$ diagram) is evaluated to define the yield surface.

1.3 Methodology

The study explores rate-independent materials, with elastic and elastoplastic properties. The primary challenge associated with inelastic materials is their path-dependent nature, where stresses depend on both current and past deformation. To address this, neural network training should be in terms of data sequences rather than simple input-output pairs. Two potential solutions have been proposed:

- A history-based model: takes stress and strain histories as inputs.
- An internal variable-based model: considers the new material state in terms of the previous state.

In both approaches, the network output at time step n is fed into the input of the subsequent time step $n+1$. A case study on the performance of different architectures in modelling elastoplasticity is performed with the two proposed strategies of neural network [3].

The described internal variable based methodology is applied to several numerical examples including uniaxial perfect von Mises elastoplasticity, uniaxial elastoplasticity with hardening, uniaxial elastoplastic damage and plane strain perfect elastoplasticity. The methodology for network training, as shown in Figure 1.3.1, begins with data collection to represent the problem, either through physical experiments or numerical computation (*i.e.* constitutive model or RVE). The data from RVE is generated using an in-house Multi-Physics Analysis Program (MPAP), developed using C++ and incorporating Finite Element Method (FEM) and optimisation libraries. Following this, the data is normalised to a range of -1 to 1 within MPAP.

Subsequently, for training purposes, a gradient-free optimisation technique is employed, specifically the Multi-Objective Particle Swarm Optimisation (MOPSO), integrated into the MPAP optimisation library. The trained model is then validated using unseen data. If further improvement is required, a fine-tuning procedure using a gradient-based technique is applied, implemented in Python with the PyTorch library.

Finally, the data is post processed back to its original scale, and thermodynamic consistency is evaluated, to

ensure that the trained model does not violate thermodynamic laws.

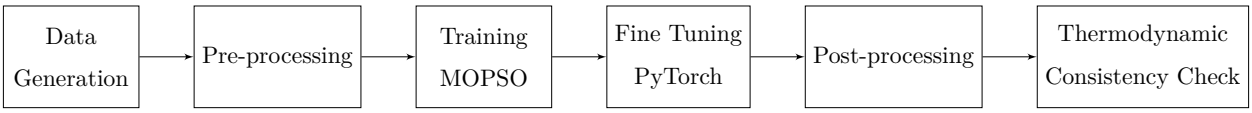


Figure 1.3.1: Flowchart illustrating the training procedure.

1.4 Thesis Layout

The thesis is organised as follows : Chapters 2, 3, 4 and 5 provide a review of fundamental concepts and principles essential for the subsequent research conducted in Chapters 6, 7, 8 and 9. In particular:

Chapter 2: Reviews basic principles related to continuum mechanics to describe materials mechanical behaviour. Additionally, the constitutive relationships that correspond to von Mises elastoplastic material model and the numerical implementation with elastic predictor-plastic corrector are discussed.

Chapter 3: An introduction to the discrete boundary value problem and FEM application in small strain.

Chapter 4: Computer implementation of the multi-scale constitutive framework for material models is reviewed. In addition, the analytical solution known as the rule of mixtures method to evaluate the elastic moduli of matrix-fibres laminate is explained. Moreover, analytical methods for porous elastic materials have been discussed and compared to the results determined with numerical simulation based on RVE. Furthermore, an assessment of the Gurson model is undertaken with regard to two different specified boundary conditions namely linear and periodic boundary conditions.

Chapter 5: A brief introduction to machine learning and neural networks.

Chapter 6: Two proposed strategies for stress update which are history-based and internal variable-based strategies. The strategies are capable of representing exactly the standard stress update algorithm for elastoplasticity in the uniaxial case.

Chapter 7: Presenting the generalised architecture of the proposed two network formulation neural network, the recurrent nature of the algorithm and their implications on the training strategy. Moreover, a computationally efficient criterion for the verification of thermodynamic consistency is proposed.

Chapter 8: The employed training methodology with MOPSO.

Chapter 9: Several numerical examples are provided to demonstrate that:

- The training process recovers the exact stress data for uniaxial elastoplasticity.

- The applications of the strategy can be extended to other nonlinear problems namely uniaxial elastoplastic damage and plane strain elastoplasticity.
- The strategy can be applied to numerical examples of material computational homogenisation based on plane strain elastoplasticity RVE for porous solid material.

Chapter 10: The achievements of this work are summarised and concluding remarks are presented. In addition, future work suggestions are provided.

Chapter 2

Constitutive Modelling

Constitutive modelling has been utilised to describe materials mechanical behaviour and physical characteristics mathematically. The formulation of these constitutive relationships is important as it is widely used in industrial applications and practical engineering problems to link the stress state to the strain state.

This chapter will describe the constitutive theory with internal variables in Section 2.1. The principles of thermodynamics, including energy conservation and the dissipation inequality, will be covered in Section 2.2. Following that, the types of constitutive material models are explained: linear elasticity in Section 2.3, anisotropic elasticity in Section 2.4 and the constitutive relationships for the von Mises elastoplastic material model is given in Section 2.5. In addition, the numerical implementation with elastic predictor-plastic corrector is covered in Section 2.6. All of this information is crucial for understanding the subsequent chapters, particularly Chapters 6 through 9.

2.1 Constitutive Theory with Internal Variables

In any physical system, two types of state variables can be identified which are *observable* and *internal non-observable*. The observable variables are measurable and can be determined directly by physical measurements such as strains and temperature. Whereas, the internal variables refer to quantities that describe the state of

a material system, but are not directly observable and requires an integration process that works with other observable magnitudes and are dependant on the systems specific structure. In an engineering context, these variables are often used to model the behavior of materials, such as their deformation and damage, at the micro-scale.

The constitutive theory that incorporates internal variables is formulated based on the idea that the thermodynamic state at a given point can be fully determined by a finite number of state variables at any instant. The thermodynamic state is dependent only on the current value of these state variables and not on their past history. The assumption is that the thermodynamic state at a given point can be determined at any time t by the following set of state variables [98]

$$\{\boldsymbol{\varepsilon}, \boldsymbol{\alpha}\}, \tag{2.1.1}$$

where $\boldsymbol{\varepsilon}$ is the strain tensor and $\boldsymbol{\alpha}$ represent the set of internal variables that may include scalar, vectorial and tensorial variables associated with dissipation mechanism

$$\boldsymbol{\alpha} = \{\alpha_k\}, \tag{2.1.2}$$

hence, the specific free energy can be expressed in the following form [98]

$$\psi = \psi(\boldsymbol{\varepsilon}, \boldsymbol{\alpha}). \tag{2.1.3}$$

The material constitutive behaviour is described by the following equations

$$\begin{aligned} \boldsymbol{\sigma} &= \frac{\partial \psi}{\partial \varepsilon}, \\ \dot{\boldsymbol{\alpha}} &= f(\boldsymbol{\varepsilon}, \boldsymbol{\alpha}). \end{aligned} \tag{2.1.4}$$

Effectively, the functions ψ and f are used to define any model of the present type.

2.2 Thermodynamic Principles

The conservation of energy is the first principle of thermodynamics which states that the system total energy should remains constant which means that it cannot be created or destroyed. Although it may be transferred

from one form to another, stored or dissipated. The mathematical expression of the conservation of energy can be expressed as [23]

$$\rho \dot{\boldsymbol{e}} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} - \boldsymbol{\nabla} \cdot q + \rho h, \tag{2.2.1}$$

where ρ is the material mass density, $\boldsymbol{e}$ the specific internal energy. The $\boldsymbol{\sigma}$ represent the Cauchy stress tensor, $\boldsymbol{\varepsilon}$ is the strain tensor, q the rate of heat flux and h the specific energy source.

The second principle of thermodynamics is related to heat and energy interconversions and it is commonly used in continuum mechanics problems to indicate that the material constitutive relation is thermodynamically allowable. The mathematical term of the second principle of thermodynamics can be expressed in terms of the local Clausius–Duhem inequality which also known as dissipation inequality [23]

$$\rho \dot{\boldsymbol{s}} \quad \geq \quad \frac{\rho h}{\theta} - \boldsymbol{\nabla} \cdot (\frac{q \cdot n}{\theta}), \tag{2.2.2}$$

where the entropy is $\boldsymbol{s}$, the rate of the supplied entropy is $-\frac{h}{\theta}$ and the rate of entropy flux is $\frac{q \cdot n}{\theta}$. When eliminating the heat supply h in Equations (2.2.1) and (2.2.2) the following expression is obtained [23]

$$\rho(\theta \dot{\boldsymbol{s}} - \dot{\boldsymbol{e}}) + \boldsymbol{\sigma} : \boldsymbol{\varepsilon} - \frac{q \cdot \nabla \theta}{\theta} \quad \geq \quad 0. \tag{2.2.3}$$

where the rate of mechanical dissipation is $D = \rho(\theta \dot{\boldsymbol{s}} - \dot{\boldsymbol{e}}) + \boldsymbol{\sigma} : \boldsymbol{\varepsilon}$ and the thermal dissipation is $D^{th} = -\frac{q \cdot \nabla \theta}{\theta}$. The thermal dissipation is a non-negative value due to heat energy transfer from higher to lower temperatures. Based on that, the mechanical dissipation rate should be a positive value in order to ensure that the dissipation is always greater than or equal to zero (*i.e.* $D \geq 0$).

2.3 Linear Elastic Material Model

The linear elastic model is comprehensively described in [19] and it is used to estimate the materials mechanical characteristics under small strain conditions where stress and strain values are proportional. For a given value of the strain ε, the strain energy density in an index form is expressed as following

$$\psi = \frac{1}{2} C_{ijkl} \varepsilon_{kl} \varepsilon_{ij}, \tag{2.3.1}$$

where C is the fourth order elasticity tensor. The linear stress-strain relation can be defined as

$$\sigma_{ij} = \frac{\partial \psi}{\partial \varepsilon_{ij}} = C_{ijkl}\varepsilon_{kl}. \tag{2.3.2}$$

In general, the fourth order elasticity tensor include 81 independent components. However due to the stress and strain symmetry the number of independent components is reduced to 36. In addition, because of the symmetry in stress and strain tensors, due to the strain energy relation, the elasticity tensor should therefore be a symmetric tensor [23]

$$\boldsymbol{C}_{ijkl} = \boldsymbol{C}_{klij}. \tag{2.3.3}$$

The symmetry in equation 2.3.3 reduces the number of independent components of the elasticity tensor to 21. Its worth mentioning that when the material structure has symmetries the number of material constants is reduced further. The stress-strain relations with 21 independent components for $\boldsymbol{C}_{ijkl}$, no symmetry planes and fully anisotropic, can be written as

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ & & C_{33} & C_{34} & C_{35} & C_{36} \\ & & & C_{44} & C_{45} & C_{46} \\ & Sym. & & & C_{55} & C_{56} \\ & & & & & C_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{bmatrix} \tag{2.3.4}$$

In the case of symmetry with respect to one plane, the structure of the elasticity tensor $\boldsymbol{C}_{ij}$ of such material includes 13 independent coefficients expressed as

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & C_{16} \\ & C_{22} & C_{23} & 0 & 0 & C_{26} \\ & & C_{33} & 0 & 0 & C_{36} \\ & & & C_{44} & C_{45} & 0 \\ & Sym. & & & C_{55} & 0 \\ & & & & & C_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{bmatrix} \tag{2.3.5}$$

For orthotropic materials that are characterised by three mutually orthogonal planes of reflection symmetry. The number of independent coefficients is 9 and the structure of the elasticity tensor $\boldsymbol{C}_{ij}$ is written as

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\\ \sigma_5\\ \sigma_6\end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0\\ & C_{22} & C_{23} & 0 & 0 & 0\\ & & C_{33} & 0 & 0 & 0\\ & & & C_{44} & 0 & 0\\ & Sym. & & & C_{55} & 0\\ & & & & & C_{66}\end{bmatrix} \begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \varepsilon_4\\ \varepsilon_5\\ \varepsilon_6\end{bmatrix} \tag{2.3.6}$$

For a transversely isotropic material, the material physical properties are characterised to be symmetric about an axis that is normal to a plane of isotropy (xy-plane). Therefore, the structure of the elasticity tensor $\boldsymbol{C}_{ij}$ would reduce to five independent components as

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\\ \sigma_5\\ \sigma_6\end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{21} & 0 & 0 & 0\\ & C_{22} & C_{23} & 0 & 0 & 0\\ & & C_{22} & 0 & 0 & 0\\ & & & \frac{1}{2}(C_{22}-C_{23}) & 0 & 0\\ & Sym. & & & C_{66} & 0\\ & & & & & C_{66}\end{bmatrix} \begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \varepsilon_4\\ \varepsilon_5\\ \varepsilon_6\end{bmatrix} \tag{2.3.7}$$

For an isotropic material, the structure of the tensor $\boldsymbol{C}_{ij}$ reduce to two independent components as

$$\begin{bmatrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\\ \sigma_5\\ \sigma_6\end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0\\ & C_{11} & C_{12} & 0 & 0 & 0\\ & & C_{11} & 0 & 0 & 0\\ & & & \frac{1}{2}(C_{11}-C_{12}) & 0 & 0\\ & Sym. & & & \frac{1}{2}(C_{11}-C_{12}) & 0\\ & & & & & \frac{1}{2}(C_{11}-C_{12})\end{bmatrix} \begin{bmatrix}\varepsilon_1\\ \varepsilon_2\\ \varepsilon_3\\ \varepsilon_4\\ \varepsilon_5\\ \varepsilon_6\end{bmatrix} \tag{2.3.8}$$

Corresponding to equation (2.3.8) the constitutive model for an isotropic, linear elastic and homogeneous

material can be expressed as

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij} \tag{2.3.9}$$

where δ_{ij} is the Kronecker-delta, λ and μ are the Lame coefficients. The constitutive law is formulated with two parameters which means that the three parameters of Young's modulus E, Possion's ratio ν and shear modulus G are not independent of one another. The Lame coefficients can be related to these three parameters with the following relationship

$$G = \mu, \qquad E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu}, \qquad \nu = \frac{\lambda}{2(\lambda + \mu)}, \qquad \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}. \tag{2.3.10}$$

where μ is the shear modulus.

For plane stress problems, the stress components out of the plane equal 0 (*i.e.* $\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0$). Therefore, the stress and strain relationship is as follows

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} \tag{2.3.11}$$

For plane strain problems, the strain components out of the plane equal 0 (*i.e.* $\varepsilon_{zz} = \varepsilon_{xz} = \varepsilon_{yz} = 0$). Therefore, the stress and strain relationship is as follows

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} \tag{2.3.12}$$

2.4 Anisotropic Plane Elasticity Models

The linear elastic law for isotropic material model is expressed as [23]

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl}, \qquad \varepsilon_{ij} = S_{ijkl} \varepsilon_{kl} \tag{2.4.1}$$

where $\boldsymbol{C}$ is stiffness tensor and $\boldsymbol{S}$ is compliance tensor. When considering an orthotropic lamina containing unidirectional fibres at an orientation angle of $\theta = 0$ or $\theta = 90$, the components of the stiffness and compliance matrices can be expressed as

$$\begin{aligned} C_{11} &= \frac{E_{11}}{1-\nu_{12}\nu_{21}} & S_{11} &= \frac{1}{E_{11}} \\ C_{22} &= \frac{E_{22}}{1-\nu_{12}\nu_{21}} & S_{22} &= \frac{1}{E_{22}} \\ C_{12} &= \frac{\nu_{12}E_{22}}{1-\nu_{12}\nu_{21}} & S_{12} &= \frac{-\nu_{12}}{E_{11}} \\ C_{21} &= \frac{\nu_{21}E_{11}}{1-\nu_{21}\nu_{12}} & S_{12} &= \frac{-\nu_{21}}{E_{22}} \\ C_{66} &= G_{12} & S_{66} &= \frac{1}{G_{12}} \end{aligned} \tag{2.4.2}$$

The form of elasticity matrix $\boldsymbol{C}$ and compliance matrix $\boldsymbol{S}$ for a general orthotropic lamina $\theta \neq 0$ or $\theta \neq 90$ could be expressed as

$$\boldsymbol{C} = \begin{bmatrix} \bar{C}_{11} & \bar{C}_{12} & \bar{C}_{16} \\ \bar{C}_{12} & \bar{C}_{22} & \bar{C}_{26} \\ \bar{C}_{16} & \bar{C}_{26} & \bar{C}_{66} \end{bmatrix}, \qquad \boldsymbol{S} = \begin{bmatrix} \bar{S}_{11} & \bar{S}_{12} & \bar{S}_{16} \\ \bar{S}_{12} & \bar{S}_{22} & \bar{S}_{26} \\ \bar{S}_{16} & \bar{S}_{26} & \bar{S}_{66} \end{bmatrix} \tag{2.4.3}$$

The elements in the elasticity matrix could be expressed in the principal material direction by introducing the transformation matrix $\boldsymbol{T}$

$$\boldsymbol{T} = \begin{bmatrix} m^2 & n^2 & 2mn \\ n^2 & m^2 & -2mn \\ -mm & mn & m^2-n^2 \end{bmatrix} \tag{2.4.4}$$

where $m = \cos\theta$ and $n = \sin\theta$

$$\begin{aligned}
\bar{C}_{11} &= m^4 C_{11} + 2m^2 n^2 (c_{12} + 2C_{66}) + n^4 C_{22}, \\
\bar{C}_{12} &= n^2 m^4 (C_{11} + C_{22} - 4C_{66}) + (n^4 + m^4) C_{12}, \\
\bar{C}_{16} &= nm[m^2 (C_{11} - C_{12} - 2C_{66}) + n^2 (C_{12} - C_{22} + 2C_{66})], \\
\bar{C}_{22} &= n^4 C_{11} + 2m^2 n^2 (C_{12} + 2C_{66}) + m^4 C_{22}, \\
\bar{C}_{26} &= nm[n^2 (C_{11} - C_{12} - 2C_{66}) + m^2 (C_{12} - C_{22} + 2C_{66})], \\
\bar{C}_{66} &= n^2 m^2 (C_{11} - 2C_{12} + C_{22}) + (n^2 - m^2)^2 C_{66}.
\end{aligned} \tag{2.4.5}$$

2.5 Elastoplastic Material Model

The elastoplasticity constitutive model is used to describe solid material that exhibit loading and unloading behaviour, resulting in a plastic or irreversible deformation. The model is restricted to rate-independent plasticity which means that the material permanent deformations is not influenced by the loading rate. There are different types of hardening including:

- **Perfect plasticity:** The material is considered perfectly plastic when no hardening is applied. In this case, as the material approaches the uniaxial yield stress σ_y while undergoing continuous monolithic straining, the yield surface remains unchanged in shape and size throughout the development of plastic deformations.
- **Isotropic hardening:** When cyclic loading is applied to the plastically deformed material, unloading and then reloading beyond the previously reached maximum stress can lead to additional plastic deformation. This increase in material resistance to plastic flow causes the yield surface to expand proportionally in all directions around the hydrostatic axis when the yield stress is exceeded.

2.5.1 Uniaxial Elastoplastic Model

The von Mises criteria states that plastic deformation is achieved when the maximum octahedral shear stress reaches the critical value. The mathematical formulation of solid materials deformations at small strains elastoplastic are composed of elastic strain ε^e and plastic strain ε^p components:

$$\varepsilon = \varepsilon^e + \varepsilon^p. \tag{2.5.1}$$

The corresponding stress σ can be expressed as

$$\sigma = E\varepsilon^e = E(\varepsilon - \varepsilon^p). \tag{2.5.2}$$

The standard von Mises yield criterion is given as

$$f(\sigma, \alpha) = |\sigma| - (\sigma_y + H\alpha) \leq 0, \tag{2.5.3}$$

where E is the Young's Modulus, H is the isotropic hardening and α is the internal hardening variable. The flow rule equation and hardening law can be expressed respectively as

$$\dot{\varepsilon^p} = \dot{\lambda}\text{sign}(\sigma), \qquad \dot{\alpha} = |\dot{\varepsilon^p}| \tag{2.5.4}$$

where α is internal variable while sign() denotes the classical sign function. The standard Kuhn-Tucker loading/unloading conditions can be expressed as

$$\dot{\lambda} \geq 0, \qquad f(\sigma, \alpha) \leq 0, \qquad \dot{\lambda} f(\sigma, \alpha) = 0. \tag{2.5.5}$$

2.5.2 Multi-dimensional Elastoplastic Model

The 3D tensor representation of the equations mentioned in Section 2.5.1 could be formulated as follows [23]

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p. \tag{2.5.6}$$

The elastic part of the constitutive relation is used to define the corresponding stress $\boldsymbol{\sigma}$ as

$$\boldsymbol{\sigma} = \mathbf{C}[\boldsymbol{\varepsilon}^e] = \mathbf{C}[\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p], \tag{2.5.7}$$

where $\mathbf{C}$ is the elastic modulus. The material follow the standard von Mises yield criterion that can expressed as

$$f(\boldsymbol{\sigma}, q) = \|\mathbf{s}\| - \sqrt{\frac{2}{3}}(\sigma_y + q) \leq 0, \tag{2.5.8}$$

where $\|\cdot\|$ is the norm of the deviatoric stress tensors $\mathbf{s}$, σ_y is the uniaxial yield stress and q is the hardening parameter. The evolution of plastic flow is governed by the flow rule given as

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda}\mathbf{n}, \qquad \dot{q} = \dot{\lambda}\sqrt{\frac{2}{3}}H, \qquad \dot{\alpha} = \dot{\lambda}\sqrt{\frac{2}{3}}. \tag{2.5.9}$$

where $\mathbf{n}= \frac{\mathbf{s}}{\|\mathbf{s}\|}$ is the unit normal on the yield surface, $\dot{\lambda}$ is the plastic multiplier and H is the hardening modulus. The standard Kuhn-Tucker loading/unloading conditions are given as follows

$$\dot{\lambda} \geq 0, \qquad f(\boldsymbol{\sigma}, \boldsymbol{q}) \leq 0, \qquad \dot{\lambda} f(\boldsymbol{\sigma}, \boldsymbol{q}) = 0. \tag{2.5.10}$$

2.5.3 Uniaxial Elastoplastic Damage Model

By assuming the strain equivalence principle introduced by Lemaitre [60], the damage value in a given plane is obtained by computing the intersection of the effective area with that plane. For a 1D case, material damage D can be expressed as

$$D = \frac{S_D}{S}, \tag{2.5.11}$$

where S_D is the damaged section and S is the effective section. The scalar D value vary between $D = 0$ for undamaged material and $D = 1$ for a totally damaged material. The effective stress can be defined as

$$\tilde{\sigma} = \frac{F}{(S - S_D)} = \frac{\sigma}{1 - D}, \tag{2.5.12}$$

The free energy function follows the form

$$\Psi(\varepsilon^e, \varepsilon^p, D) = (1 - D) \cdot \left(\frac{1}{2}E{\varepsilon^e}^2 + \frac{1}{2}H{\varepsilon^p}^2\right), \tag{2.5.13}$$

while the total potential energy function can be written as follows

$$\omega_t = \Psi^0(\varepsilon^e, \varepsilon^p) = \frac{1}{2}E{\varepsilon^e}^2 + \frac{1}{2}H{\varepsilon^p}^2. \tag{2.5.14}$$

The Clausius-Duhem inequality is given by $\sigma \cdot \dot{\varepsilon} - \dot{\Psi} \geq 0$, whereas the stress and the effective stress can be

defined as

$$\sigma = \frac{\partial \Psi}{\partial \varepsilon^e} = (1 - D) \cdot \frac{\partial \Psi^0}{\partial \varepsilon^e}, \qquad \tilde{\sigma} = \frac{\sigma}{1 - D} = \frac{\partial \Psi^0}{\partial \varepsilon^e}. \tag{2.5.15}$$

The damage criterion can be expressed as following:

$$g(\omega_t, r_t) = \omega_t - r_t \leq 0, \tag{2.5.16}$$

where r_t corresponds to the current damage threshold and r_0 refers to the initial damage threshold. The damage evolution law can be expressed as

$$\dot{D}_t = \dot{\mu}, \qquad \dot{r_t} = \dot{\mu}, \tag{2.5.17}$$

where $\dot{\mu} \geq 0$ is the damage consistency parameter. The classic damage Kuhn-Tucker conditions are

$$\dot{\mu} \geq 0, \qquad g(\omega_t, r_t) \leq 0, \qquad \dot{\mu} \cdot g(\omega_t, r_t) = 0. \tag{2.5.18}$$

The value of $\dot{\mu}$ can be obtained using the damage consistency condition and is given as

$$g(\omega_t, r_t) = \dot{g}(\omega_t, r_t) = 0 \Rightarrow \dot{\mu} = \dot{\omega_t}, \tag{2.5.19}$$

where r_t can be expressed as

$$r_t = \max\{r_0; \max_{s \in [0,t]} \omega_s\}. \tag{2.5.20}$$

2.6 Elastic Predictor - Plastic Corrector

The numerical computation applied in this study uses the standard time integration scheme of elastic predictor-plastic corrector procedure because it is easy to implement and has good numerical performance.

The algorithm is initiated with the values of elastic strain ε_n^e and the internal variables α_n at the beginning of the pesudo-time interval $[t_n, t_{n+1}]$ and a prescribed incremental strain $\Delta\varepsilon$ for this interval. The current stress update procedure can be described in two steps: First a trial elastic stress predictor is calculated, and if yielding is detected, a plastic corrector is then introduced. The corrector is calculated by return mapping of

the trial stress σ_{n+1} to the current yield surface $f_{n+1} = f(\sigma_{n+1}, \alpha_{n+1})$ as it is presented graphically in Figure 2.6.1. In other words, this means that $f > 0$ indicates inelastic behavior and $f \leq 0$ indicates elastic behavior. The incremental change in f caused by the changes in the internal variables, results in more inelastic state. Finally, an approximated solution can be obtained for the plastic corrector at t_{n+1}

$$\{\boldsymbol{\sigma_{n+1}}, \boldsymbol{\varepsilon^e_{n+1}}, \boldsymbol{\varepsilon^p_{n+1}}, \boldsymbol{\alpha_{n+1}}\}. \tag{2.6.1}$$

Algorithms 1 and 2 represent the 1D and 3D known fully implicit Backward-Euler elastic predictor/ plastic corrector return mapping respectively. The implementation is comprehensively explained in [98, 100].

The numerical computation of damage in the standard rate independent plasticity Newton-Raphson approach requires introducing a damage corrector step after the classic elastic predictor-plastic corrector steps. The detailed procedure is shown in Algorithm 3.

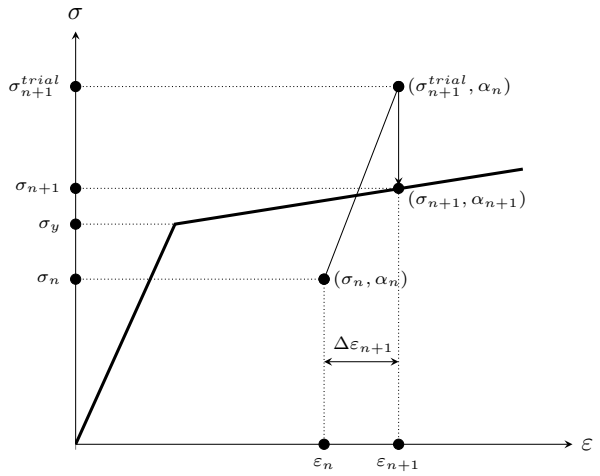


Figure 2.6.1: Ghraphical representation of the elastic predictor - plastic corrector algorithm.

Algorithm 1 1D Elastic Predictor - Plastic Corrector.

1: $\sigma_{n+1}^{trial} = E(\varepsilon_{n+1} - \varepsilon_n^p)$ ▷ Evaluate elastic predictor

2: $f_{n+1}^{trial} = \left|\sigma_{n+1}^{trial}\right| - (\sigma_y + H\alpha_n)$ ▷ Check yield condition

3: **if** $f_{n+1}^{trial} \leq 0$ **then** ▷ The load step is elastic

4: $\quad \sigma_{n+1} = \sigma_{n+1}^{trial}, \quad \varepsilon_{n+1}^p = \varepsilon_n^p, \quad \alpha_{n+1} = \alpha_n$

5: **else** ▷ The load step is plastic

6: $\quad \Delta\lambda = \dfrac{f_{n+1}^{trial}}{E + H}$

7: $\quad \sigma_{n+1} = \left[1 - \frac{\Delta\lambda E}{|\sigma_{n+1}^{trial}|}\right]\sigma_{n+1}^{trial}, \quad \varepsilon_{n+1}^p = \varepsilon_n^P + \Delta\lambda \text{sign}[\sigma_{n+1}^{trial}], \quad \alpha_{n+1} = \alpha_n + \Delta\lambda$

Algorithm 2 3D Elastic Predictor - Plastic Corrector von Mises elastoplasticity without Hardening.

1: $\mathbf{s}_{n+1}^{trial} = 2\mu(\mathbf{e}_{n+1} - \mathbf{e}_n^p)$ ▷ Evaluate elastic predictor

2: $f_{n+1}^{trail} = \left\|\mathbf{s}_{n+1}^{trial}\right\| - \sqrt{\frac{2}{3}}(\sigma_y)$ ▷ Check yield condition

3: **if** $f_{n+1}^{trial} \leq 0$ **then** ▷ The load step is elastic

4: $\quad \sigma_{n+1} = \mathbf{s}_{n+1}^{trial} + K\text{tr}[\varepsilon_{n+1}]\mathbf{I}, \quad \mathbf{e}_{n+1}^p = \mathbf{e}_n^p$

5: **else** ▷ The load step is plastic

6: $\quad \Delta\lambda = \dfrac{f_{n+1}^{trial}}{2\mu}$ ▷ Evaluate consistency parameter

7: $\quad \mathbf{n}_{n+1} = \frac{\mathbf{s}_{n+1}^{trial}}{\left\|\mathbf{s}_{n+1}^{trial}\right\|}, \quad \boldsymbol{\sigma}_{n+1} = K\text{tr}[\varepsilon_{n+1}]\mathbf{I} + \mathbf{s}_{n+1}^{trial} - 2\mu\lambda_{n+1}\mathbf{n}_{n+1}, \quad \mathbf{e}_{n+1}^p = \mathbf{e}_n^p + \lambda_{n+1}\mathbf{n}_{n+1}$

Algorithm 3 Elastic Predictor - Plastic Corrector and Damage Corrector

Evaluate elastic predictor $\tilde{\sigma}_{n+1}^{trial} = E(\varepsilon_{n+1} - \varepsilon_n^p)$

Check yield condition $f_{n+1}^{trial} = \left|\tilde{\sigma}_{n+1}^{trial}\right| - (\sigma_y + H\alpha_n)$

if $f_{n+1}^{trial} \leq 0$ **then** ▷ The load step is elastic

  $\tilde{\sigma}_{n+1} = \tilde{\sigma}_{n+1}^{trial}, \quad \varepsilon_{n+1}^p = \varepsilon_n^p, \quad \alpha_{n+1} = \alpha_n$ ▷ proceed to step (8)

else ▷ The load step is plastic

  $\Delta\lambda = \dfrac{f_{n+1}^{trial}}{E+H}, \qquad \tilde{\sigma}_{n+1} = \left[1 - \frac{\Delta\lambda E}{|\tilde{\sigma}_{n+1}^{trial}|}\right]\tilde{\sigma}_{n+1}^{trial}$

  $\varepsilon_{n+1}^p = \varepsilon_n^P + \Delta\lambda \text{sign}[\tilde{\sigma}_{n+1}^{trial}], \quad \alpha_{n+1} = \alpha_n + \Delta\lambda$

Damage corrector $\quad \omega_{n+1} = \frac{1}{2}E\varepsilon_{n+1}^{e^2} + \frac{1}{2}H\varepsilon_{n+1}^{p^2}$

if $\omega_{n+1} - r_n \leq 0$ **then**

  $D_{n+1} = D_n$

else

  $D_{n+1} = D_n + (\omega_{n+1} - \omega_n)$

$r_{n+1} = \max\{r_n; \omega_{n+1}\}$

$\sigma_{n+1} = (1 - D_{n+1})\tilde{\sigma}_{n+1}$

Chapter 3

Finite Element Method (FEM)

The Finite Element Method (FEM) is used to solve complex domain geometries due to its efficient performance on both structured and unstructured meshes. This section discusses the basics of FEM implementation to the Boundary Value Problem (BVP) for solid mechanics under small strains. A comprehensive explanation of FEM theory can be found in [23].

3.1 Stress and Equilibrium

The equilibrium equations, which describe the problem *strong form*, are used to determine the reactions and can be expressed as

$$div[\boldsymbol{\sigma}] + \boldsymbol{b} = \mathbf{0}. \tag{3.1.1}$$

To begin finite element analysis, the strong form in Equation (3.1.1) must be converted into the integral form known as *weak form* or *the principle of virtual work* (PVW). The virtual work $\delta \boldsymbol{W}$ is obtained by applying an arbitrary virtual displacement $\boldsymbol{\eta}$ in the current configuration to Equation (3.1.1) and given as

$$\delta \boldsymbol{W} = \int_{\Omega} \boldsymbol{\sigma} : \nabla \boldsymbol{\eta} \, dv - \int_{\partial\Omega} \boldsymbol{t} \cdot \boldsymbol{\eta} \, da = 0. \tag{3.1.2}$$

3.2 Incremental Boundary Value Problem

When applying a quasi-static initial boundary value problem under infinitesimal deformation, reference and deformed configurations coincide and the virtual work equation is given as [23]

$$\int_{\Omega} [\boldsymbol{\sigma}(t) : \nabla \boldsymbol{\eta} - \boldsymbol{b}(t) \cdot \boldsymbol{\eta}] \, dv - \int_{\partial \Omega_t} [\boldsymbol{t}(t) \cdot \boldsymbol{\eta}] \, da = 0, \;\; \forall \, \boldsymbol{\eta} \in \mathscr{V}, \tag{3.2.1}$$

where $\boldsymbol{b}$ and $\boldsymbol{t}$ are the body force per unit deformed volume and boundary traction per unit deformed area, respectively. $\boldsymbol{\eta}$ are virtual displacements, and $\mathscr{V}$ is the space of virtual displacements expressed as [23]

$$\mathscr{V} = \{\boldsymbol{\eta} : \Omega \to \mathscr{U} \mid \boldsymbol{\eta} = \boldsymbol{0} \;\; on \;\; \partial \Omega_u\}, \tag{3.2.2}$$

In finite element solution, time and domain discretisation are necessary. This process is essential for converting the continuous problem into a discrete one, involving sets of incremental algebraic finite element equations. These equations can be solved at each time step within a specified time interval.

In the framework of mechanical theory, the stress tensor $\boldsymbol{\sigma}_{n+1}$ is approximated through the integration procedure, which identifies an incremental constitutive function $\hat{\boldsymbol{\sigma}}$ for the stress tensor that depends on the strain $\boldsymbol{\varepsilon}_{n+1}$ at time t_{n+1} and a set of internal variables $\boldsymbol{\alpha}_n$.

$$\boldsymbol{\sigma}_{n+1} = \hat{\boldsymbol{\sigma}}(\boldsymbol{\alpha}_n, \boldsymbol{\varepsilon}_{n+1}), \tag{3.2.3}$$

where $\boldsymbol{\sigma}_{n+1}$ is expected to converge towards the exact solution of the actual evolution problem as the strain increments are reduced. In addition, a similar incremental constitutive function $\hat{\boldsymbol{\alpha}}$ is also defined through the integration algorithm for the internal variables of the model.

$$\boldsymbol{\alpha}_{n+1} = \hat{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_n, \boldsymbol{\varepsilon}_{n+1}), \tag{3.2.4}$$

Hence, the incremental form of Equation (3.2.1) is defined as follows [23] given the set $\boldsymbol{\alpha}_n$ of internal variables at t_n , find a displacement field $\boldsymbol{u}_{n+1} \in \mathscr{K}_{n+1}$, such that

$$\int_{\Omega} [\hat{\boldsymbol{\sigma}}(\boldsymbol{\alpha}_n, \nabla^s \boldsymbol{u}_{n+1}) : \nabla^s \boldsymbol{\eta} - \boldsymbol{b}_{n+1} \cdot \boldsymbol{\eta}] \, dv - \int_{\partial \Omega_t} [\boldsymbol{t}_{n+1} \cdot \boldsymbol{\eta}] \, da = 0, \tag{3.2.5}$$

for any $\boldsymbol{\eta} \in \mathscr{V}$, where $\boldsymbol{b}_{n+1}$ and $\boldsymbol{t}_{n+1}$ are the body force and surface traction fields prescribed at time station t_{n+1}. The set $\mathscr{K}_{n+1}$ is defined as

$$\mathscr{K}_{n+1} = \{\boldsymbol{u} : \Omega \to \mathscr{U} \mid \boldsymbol{u} = \overline{\boldsymbol{u}}_{n+1} on\ \partial\Omega_u\}, \tag{3.2.6}$$

where $\overline{\boldsymbol{u}}_{n+1}$ is the prescribed boundary displacement at t_{n+1}. The finite element discretisation of Equation (3.2.5) is [23]

$$\Big\{ \int_{{}^h\Omega} (\boldsymbol{B}^g)^T \boldsymbol{D} \boldsymbol{B}^g \, dv \Big\} \delta\boldsymbol{u} \cdot \boldsymbol{\eta} = -\Big\{ \int_{{}^h\Omega} [(\boldsymbol{B}^g)^T \boldsymbol{\sigma} - (\boldsymbol{N}^g)^T \boldsymbol{b}] \, dv - \int_{\partial^h\Omega_t} (\boldsymbol{N}^g)^T \boldsymbol{t} \, da \Big\} \cdot \boldsymbol{\eta}, \quad \forall\, \boldsymbol{\eta} \in {}^h\mathscr{V} \tag{3.2.7}$$

where $\boldsymbol{B}^g$ is the global strain-displacement matrix, $\boldsymbol{D}$ is the consistent tangent modulus and $\boldsymbol{N}^g$ is the global finite element interpolation matrix. Since Equation (3.2.7) is satisfied for all vectors of virtual displacements $\boldsymbol{\eta}$, the expression inside the brackets can be eliminated.

Based on that, the problem is reduced to finding the nodal displacement vector $\boldsymbol{u}_{n+1}$ at time t_{n+1}, such that the incremental equilibrium equation

$$\boldsymbol{r}(\boldsymbol{u}_{n+1}) \equiv \boldsymbol{f}^{int}(\boldsymbol{u}_{n+1}) - \boldsymbol{f}^{ext}_{n+1} = \boldsymbol{0} \tag{3.2.8}$$

is satisfied, where $\boldsymbol{f}^{int}(\boldsymbol{u}_{n+1})$ and $\boldsymbol{f}^{ext}_{n+1}$ are assembled from the element vectors [23]

$$\boldsymbol{f}^{int}_{(e)} = \int_{\Omega^{(e)}} \boldsymbol{B}^T \hat{\boldsymbol{\sigma}}\big(\boldsymbol{\alpha}_n, \boldsymbol{\varepsilon}(\boldsymbol{u}_{n+1})\big) \, dv \tag{3.2.9}$$

$$\boldsymbol{f}^{ext}_{(e)} = \int_{\Omega^{(e)}} \boldsymbol{N}^T \boldsymbol{b}_{n+1} \, dv + \int_{\partial\Omega_t{}^{(e)}} \boldsymbol{N}^T \boldsymbol{t}_{n+1} \, da \tag{3.2.10}$$

where $\boldsymbol{B}$ is the elemental strain-displacement matrix and $\boldsymbol{N}$ is the finite element interpolation matrix.

In order to determine the displacement field, an iterative approach is necessary. The Newton-Raphson scheme is an algorithm known for its efficient quadratic rates of asymptotic convergence, enabling it to reach the minimum effectively. In each iteration (k) of the Newton-Raphson scheme, the linearised version of the incremental equilibrium Equation (3.2.8) needs to be solved, involving the solution of a linear system of equations:

$$\boldsymbol{K}_T \, \delta\boldsymbol{u}^{(k)} = -\boldsymbol{r}^{(k-1)} \tag{3.2.11}$$

where $-\boldsymbol{r}^{(k-1)}$ represents the residual vector and the global tangent stiffness matrix [23]

$$\boldsymbol{K}_T \equiv \int_{{}^h\Omega} (\boldsymbol{B}^g)^T \boldsymbol{D} \boldsymbol{B}^g \, dv = \left. \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}_{n+1}} \right|_{\boldsymbol{u}_{n+1}^{(k-1)}} \tag{3.2.12}$$

The solution $\delta \boldsymbol{u}^{(k)}$ of the linear system is used to apply the Newton-Raphson correction to update the global displacement vector

$$\boldsymbol{u}_{n+1}^{(k)} = \boldsymbol{u}_{n+1}^{(k-1)} + \delta \boldsymbol{u}^{(k)} \tag{3.2.13}$$

The Newton-Raphson iterations are maintained until the maximum number of iterations is reached or a convergence condition is met.

In this work, FEM is utilised for homogenisation and data generation purposes within the context of RVE. The integration of a neural network-based constitutive model into the FEM framework is a potential area for future research. However, this aspect is beyond the scope of this project.

Chapter 4

Multi-Scale Homogenisation

Given the diverse and rapid growth of new composite formulations, it becomes crucial to numerically estimate the mechanical properties of a composite with specifically defined structural elements. The composites characteristics at different scales is studied to determine the material behavior including the mechanical properties and stress-strain relations using multi-scale modeling tools. A comprehensive implementation of transition concept from macro-to-micro is found in [80].

4.1 Homogenisation-Based Multi-Scale Constitutive Theory

Based on the assumption that any material point $\boldsymbol{x}$ of the (macroscopic) continuum corresponds to a local Representative Volume Element (RVE) with domain Ω_μ, boundary $\partial\Omega_\mu$ and a length l_μ that is smaller than the characteristic length l, of the macro-continuum as illustrated in Figure 4.1.1 [80]. This ensures distinction of scales, which is essential in multi-scale methods. The domain Ω_μ of the RVE is assumed to consist of a solid part Ω_μ^s, and a void part Ω_μ^v

$$\Omega_\mu = \Omega_\mu^s \cup \Omega_\mu^v \tag{4.1.1}$$

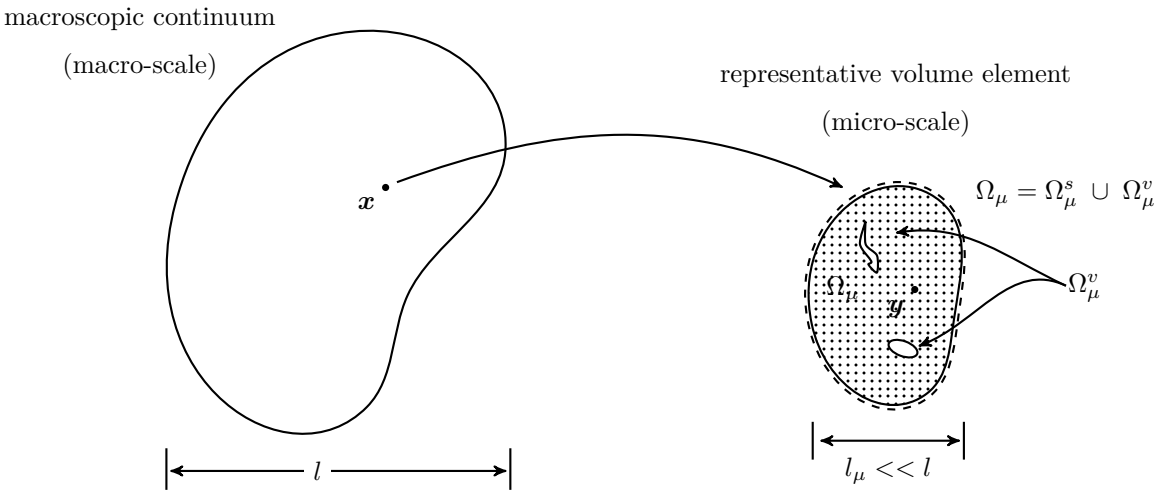


Figure 4.1.1: Macro-continuum with a locally attached micro-structure.

For simplicity, the study considers only RVEs whose void part does not intersect with the RVE boundary.

The transition concept form macro-to-micro is based on the assumption of multi-scale theory [80] which states that at any time instant t, the strain tensor at an arbitrary point $\boldsymbol{x}$ of the macro-continuum can be represented by averaging the volume of the microscopic strain tensor field $\boldsymbol{\varepsilon}_\mu$, defined over Ω_μ

$$\boldsymbol{\varepsilon}(t) = \frac{1}{V_\mu} \int_{\Omega_\mu} \varepsilon_\mu(\boldsymbol{y}, t)\, dv, \tag{4.1.2}$$

$$\boldsymbol{\varepsilon}_\mu = \nabla^s \boldsymbol{u}_\mu. \tag{4.1.3}$$

where y is the RVE local coordinate, V_μ is the RVE volume and $\nabla^s \boldsymbol{u}_\mu$ represent the symmetric gradient of the microscopic displacement field $\boldsymbol{u}_\mu$ of the RVE.

4.1.1 Kinematically Admissible RVE Displacement Fields

By substituting the Equations (4.1.3) into (4.1.2) and using the Green's theorem, it can be shown that the averaging relation (4.1.2) is equivalent to constraint on the displacement field of the RVE [24]

$$\int_{\partial\Omega_\mu} \boldsymbol{u}_\mu \otimes_s \boldsymbol{n}\, da = \int_{\partial\Omega_\mu} \frac{1}{2}(\boldsymbol{u}_\mu \otimes \boldsymbol{n} + \boldsymbol{n} \otimes \boldsymbol{u}_\mu)\, da = V_\mu \boldsymbol{\varepsilon}, \tag{4.1.4}$$

where $\boldsymbol{n}$ denotes the outward unit normal field on $\partial\Omega_\mu$. The displacement $\boldsymbol{u}_\mu$ can be factored into sum of a homogeneous strain–displacement $\boldsymbol{\varepsilon}(t)\boldsymbol{y}$ and a displacement fluctuation field $\tilde{\boldsymbol{u}}_\mu$

$$\boldsymbol{u}_\mu(\boldsymbol{y}, t) = \boldsymbol{\varepsilon}(t)\boldsymbol{y} + \tilde{\boldsymbol{u}}_\mu(\boldsymbol{y}, t), \tag{4.1.5}$$

The constraint (4.1.4) requires that the space $\tilde{\mathscr{K}}_\mu$ of kinematically admissible displacement fluctuations of the RVE be a subspace of the *minimally constrained space of kinematically admissible displacement fluctuations* $\tilde{\mathscr{K}}_\mu^*$

$$\tilde{\mathscr{K}}_\mu \subset \tilde{\mathscr{K}}_\mu^* \equiv \Big\{ \boldsymbol{v},\ \text{sufficiently regular} \quad | \int_{\partial\Omega_\mu} \boldsymbol{v} \otimes_s \boldsymbol{n}\, da = \boldsymbol{0} \Big\}. \tag{4.1.6}$$

From Equations (4.1.5) and (4.1.3) the microscopic strain can be expressed as the sum of a homogeneous strain field (coinciding with the macroscopic average strain) and a field $\nabla^s \boldsymbol{u}_\mu$ that represents a fluctuation about the average.

$$\boldsymbol{\varepsilon}_\mu(\boldsymbol{y}, t) = \boldsymbol{\varepsilon}(t) + \nabla^s \tilde{\boldsymbol{u}}_\mu(\boldsymbol{y}, t) \tag{4.1.7}$$

4.1.2 Macroscopic Stress, Hill–Mandel Principle and RVE Equilibrium

Similar to the macroscopic strain Equation (4.1.2), the macroscopic stress tensor $\boldsymbol{\sigma}$, is taken as the volume average of the microscopic stress field $\boldsymbol{\sigma}_\mu$, over the RVE [80]

$$\boldsymbol{\sigma}(t) = \frac{1}{V_\mu} \int_{\Omega_\mu} \boldsymbol{\sigma}_\mu(\boldsymbol{y}, t)\, dv, \tag{4.1.8}$$

The Hill–Mandel Principle of macro-homogeneity, requires the macroscopic stress power to be equal the volume average of the microscopic stress powers for any kinematically admissible motion of the RVE. This

is expressed by the equation [80]

$$\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} = \frac{1}{V_\mu} \int_{\Omega_\mu} \boldsymbol{\sigma}_\mu : \dot{\boldsymbol{\varepsilon}}_\mu \, dv, \tag{4.1.9}$$

which must be satisfied for any kinematically admissible microscopic strain rate field, $\dot{\boldsymbol{\varepsilon}}_\mu$. The above is equivalent to the following variational equation in terms of the RVE boundary traction and body force fields, denoted by $\boldsymbol{t}$ and $\boldsymbol{b}$, respectively.

$$\int_{\partial\Omega_\mu} \boldsymbol{t} \cdot \boldsymbol{\eta} \, da = 0, \quad \int_{\Omega_\mu} \boldsymbol{b} \cdot \boldsymbol{\eta} \, dv = 0 \quad \forall \, \boldsymbol{\eta} \in \tilde{\mathscr{K}_\mu} \tag{4.1.10}$$

The variational equilibrium statement for the RVE [80] is then given by

$$\int_{\Omega_\mu} \boldsymbol{\sigma}_\mu : \nabla^s \boldsymbol{\eta} \, dv = 0 \in \tilde{\mathscr{K}_\mu} \tag{4.1.11}$$

Furthermore, at any time t the stress at each point $\boldsymbol{y}$ of the RVE is determined by a generic constitutive functional $\mathscr{O}_y$ of the strain history $\varepsilon_\mu^t(y)$ up to time t

$$\boldsymbol{\sigma}_\mu(\boldsymbol{y}, t) = \mathscr{O}_y(\boldsymbol{\varepsilon}_\mu^t(\boldsymbol{y})) \tag{4.1.12}$$

The constitutive assumption and the equilibrium Equation (4.1.11) defines the RVE equilibrium problem which can be found for a given macroscopic strain ε (a function of time) and displacement fluctuation function $\tilde{\boldsymbol{u}}_\mu \in \tilde{\mathscr{K}_\mu}$

$$\int_{\Omega_\mu^s} \mathscr{O}_y \Big\{ \Big[\boldsymbol{\varepsilon}(t) + \nabla^s \tilde{\boldsymbol{u}}_\mu(\boldsymbol{y}, t) \Big]^t \Big\} : \nabla^s \boldsymbol{\eta} \, dv = 0 \quad \forall \, \boldsymbol{\eta} \in \tilde{\mathscr{K}_\mu} \, . \tag{4.1.13}$$

4.1.3 Characterisation of the Multi-scale Constitutive Model

For a given macroscopic strain history, the general multi-scale constitutive model can be defined by solving the RVE equilibrium problem given in Equation (4.1.13). Once the solution $\tilde{\boldsymbol{u}}_\mu$ is determined, the macroscopic stress tensor can be obtained using the averaging relation [80]

$$\boldsymbol{\sigma}(t) = \mathscr{O}(\boldsymbol{\varepsilon}^t) \equiv \frac{1}{V_\mu} \int_{\Omega_\mu} \mathscr{O}_y \Big\{ [\boldsymbol{\varepsilon} + \nabla^s \tilde{\boldsymbol{u}}_\mu]^t \Big\} \, dv, \tag{4.1.14}$$

where $\mathscr{O}$ represent the resulting (homogenised) macroscopic constitutive function.

4.2 Kinematic Constraints

The selection of an appropriate space of kinematically admissible displacement fluctuations $\tilde{\mathscr{K}_\mu} \subset \tilde{\mathscr{K}_\mu^*}$ is a crucial aspect in multiscale methods to achieve high accuracy. Two common types of boundary conditions are linearly prescribed displacement and periodic boundary conditions. The deformed configurations corresponding to these two boundary conditions are illustrated in Figure 4.2.1.

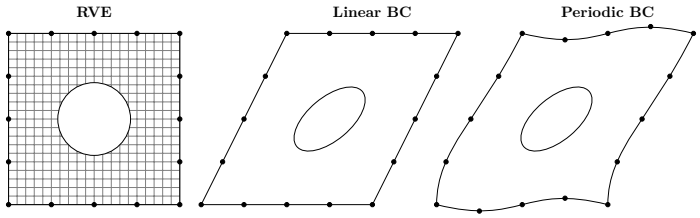


Figure 4.2.1: Schematic representation of the deformed two-dimensional RVE configuration with (a) linear and (b) periodic displacement boundary condition.

4.2.1 Linear Boundary Conditions

The linear boundary condition assumes that the degrees of freedom of boundary fluctuations are fully prescribed as zero, with only the interior nodes having free degrees of freedom. The linear boundary condition can be expressed as

$$\tilde{\mathscr{K}_\mu} = \tilde{\mathscr{K}_{lin}} \equiv \Big\{ \boldsymbol{v},\ \text{sufficiently regular} \quad | \ \boldsymbol{v(y)} = 0 \quad \forall y \in \partial\Omega_\mu \Big\}, \tag{4.2.1}$$

$$\boldsymbol{u}_\mu(y) = \varepsilon y \quad \forall y \in \partial\Omega_\mu. \tag{4.2.2}$$

4.2.2 Periodic Boundary Conditions

The expression of two-dimensional problems will be considered with the notation adopted by Michel et al. [69]. Periodic boundary conditions are imposed by dividing the RVE surface boundary into two opposing parts. Each pair of sides, denoted as i , comprises equally sized subsets.

$$\Gamma_i^+ \quad \textit{and} \quad \Gamma_i^-, \tag{4.2.3}$$

of $\partial\Omega_\mu$, with respective unit normals

$$n_i^+ \quad and \quad n_i^-, \tag{4.2.4}$$

to satisfy

$$n_i^- = -n_i^+. \tag{4.2.5}$$

For each point $\boldsymbol{y}^+$ located on the boundary of Γ_i^+, there should correspond an associated point $\boldsymbol{y}^-$ on the boundary of Γ_i^-. The primary kinematic constraint of the models is that displacement fluctuation must exhibit periodicity on the boundary of the RVE. Therefore, for each pair $\{\boldsymbol{y}^+, \boldsymbol{y}^-\}$ of boundary material points the following is satisfied

$$\tilde{\boldsymbol{u}}_\mu(\boldsymbol{y}^+, t) = \tilde{\boldsymbol{u}}_\mu(\boldsymbol{y}^-, t), \tag{4.2.6}$$

accordingly, the space $\tilde{\mathscr{K}_\mu}$ is defined as

$$\tilde{\mathscr{K}_\mu} = \tilde{\mathscr{K}_{per}} = \Big\{\tilde{\boldsymbol{u}}_\mu, \textit{suff.reg.} \mid \tilde{\boldsymbol{u}}_\mu(\boldsymbol{y}^+, t) = \tilde{\boldsymbol{u}}_\mu(\boldsymbol{y}^-, t) \ \forall \ \textit{pairs} \ \{\boldsymbol{y}^+, \boldsymbol{y}^-\}\Big\}. \tag{4.2.7}$$

4.3 Evaluation of Elastic Moduli

For orthotropic materials under plane stress state (*i.e.* laminate materials), the composite behaviour can be defined using four independent engineering constants: the longitudinal modulus of elasticity E_x, the transverse modulus of elasticity E_y, Poisson's ratio ν_{xy}, ν_{yx} and shear modulus G_{xy}. These material properties can be derived using an empirical method known as the rule of mixtures. The rule of mixtures is established based on the concept that the composite property is defined as the sum of each individual constituent properties multiplied by its volume fraction. This method is used to describe various micro-mechanical problems.

The four independent material constants for defining the properties of fiber-matrix combinations and their volume fraction ratios can be estimated as follows

1) Longitudinal modulus of elasticity E_x: The longitudinal tensile strength of composite materials is primarily determined by the strength and volume content of the fibre reinforcement. When considering a unidirectional lamina subjected to a uniaxial force along the fiber direction with the assumption that the fiber and matrix experience identical strain rate (i.e. $\varepsilon_x = \varepsilon_f = \varepsilon_m$).

$$E_x = E_f V_f + E_m V_m. \tag{4.3.1}$$

where V_f and V_m is the volume of the fibers and matrix, respectively. The fibers and matrix acts like two spring in parallel.

2) Transverse modulus of elasticity E_y: The representative volume element is subjected to a transverse stress with the assumption that the fiber and matrix experience identical stress state (i.e. $\sigma_x = \sigma_f = \sigma_m$) in order to compute the transverse Young's modulus of a unidirectional lamina.

$$E_y = \frac{E_f E_m}{E_m V_f + E_f V_m}. \tag{4.3.2}$$

3) Poisson's ratio ν_{xy}, ν_{yx}: The mathematical expression to determine the effective in-plane Poisson's ratio of composite can be define as

$$\nu_{xy} = V_f \nu_f + V_m \nu_m. \tag{4.3.3}$$

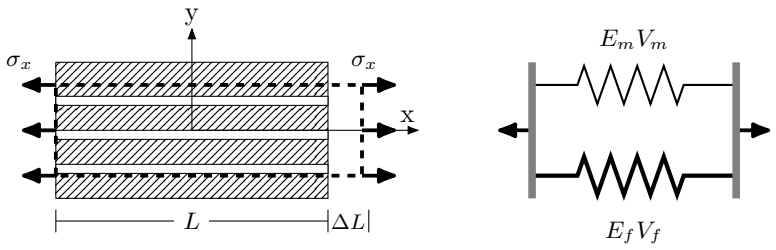
4) Shear modulus G_{xy}: A similar assumption is applied to the determination of the transverse modulus of elasticity E_y, implying that the shear stress τ of the fibers and matrix are the identical (*i.e.* $\tau = G_f \nu_f = G_m \nu_m$). The fibers and matrix contribute in proportions that are proportional to their respective volume fractions to the shear strain:

$$\gamma = V_f \nu_f + V_m \nu_m. \tag{4.3.4}$$

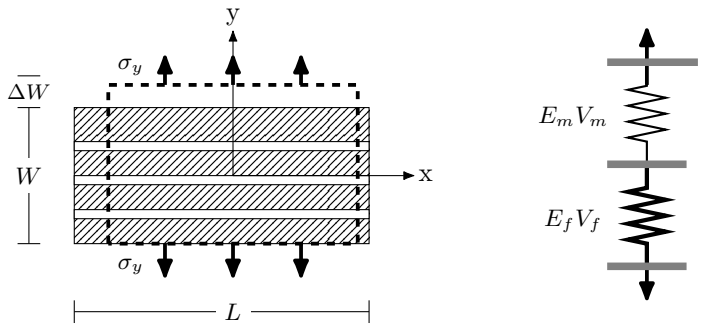
Accordingly, the shear modulus G_{xy} can obtained with the serial-spring model as following:

$$G_{xy} = \frac{\tau}{\gamma} = \frac{G_f G_m}{G_m V_f + G_f V_m}. \tag{4.3.5}$$

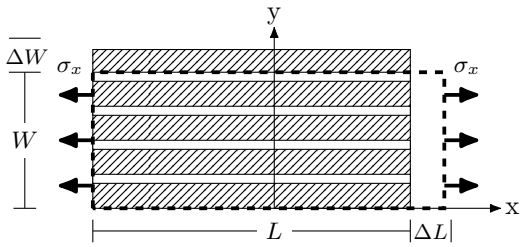
1) Longitudinal modulus of elasticity E_x



2) Transverse modulus of elasticity E_y



3) Poisson's ratio ν_{xy}, ν_{yx}



4) Shear modulus G_{xy}

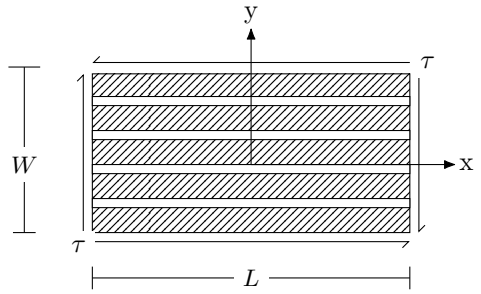


Figure 4.3.1: The mechanical behaviour of lamina to determine the 4 independent material constants.

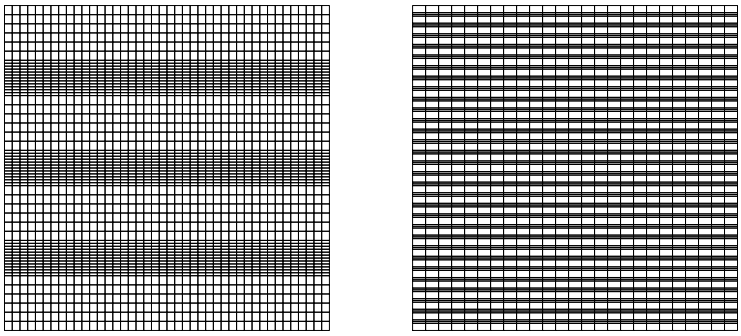
Example 1: RVE Layered Composite Model

Two RVE meshes with same total volume of $8464mm^3$ (*i.e.* the RVE length is $L = 92mm$ and width is $W = 92mm$) were generated. Each mesh has different matrix and fiber element sizes and layer distributions. The first mesh consists of concentrated layers, while the second mesh contains distributed layers, as shown in Figure 4.3.2. A linear boundary conditions were applied to the RVE models. The elastic properties used

for the matrix and fibers are as follows

Matrix: $E_m = 20GPa$, $\nu_m = 0.35$, $G_m = 7.407GPa$ and the volume is $V_m = 0.674mm^3$.

Fiber: $E_f = 200GPa$, $\nu_f = 0.45$, $G_f = 68.966GPa$ and the volume is $V_f = 0.326mm^3$.



(a) Concentrated layers (b) Distributed layers

Figure 4.3.2: RVE meshes.

The main objective of this example is to illustrate the effect of RVE distribution and density on determining the associated stresses in a plane strain case. The calculated stresses follow the concept of elastic moduli approximation described in Section 4.3. The analytically approximated values for RVE E_x,E_y,ν_{xy},ν_{yx} and G_{xy} were found to be $78.696GPa$, $28.3077GPa$, 0.383, 0.1378 and 10.449, respectively. Table 4.3.1 indicates that increasing the RVE distribution and density results in better approximation. The computed error for the concentrated layers is 10.9%, whereas, for the distributed layers is 4.5%.

Table 4.3.1: RVE subjected to prescribed macro-strain of $\varepsilon_{xx} = 0.1$, $\varepsilon_{yy} = 0.3$ and $\varepsilon_{xy} = 0.4$.

f	σ_{xx}	σ_{yy}	σ_{xy}
Calculated	11.7368	10.1076	4.1794
Concentrated layers	12.357	11.599	4.708
Distributed layers	12.099	11.037	4.2279

4.4 A Specific RVE Results in a Specific Yield Criterion

Understanding the yield surface under various loading conditions is crucial for modeling the plastic behavior of materials, as it defines the limit of material elasticity. Typically the material yield surface is presented in terms of $\bar{p} - \bar{q}$ diagram where $\bar{p}$ is the hydrostatic pressure and $\bar{q}$ is the norm of the deviatoric stress. For solid

block RVE composed of von Mises type elasto perfectly plastic material, the associated plastic deformation within the RVE occurs when no changes in macroscopic (homogenised) stress are observed while the load factor increases [97].

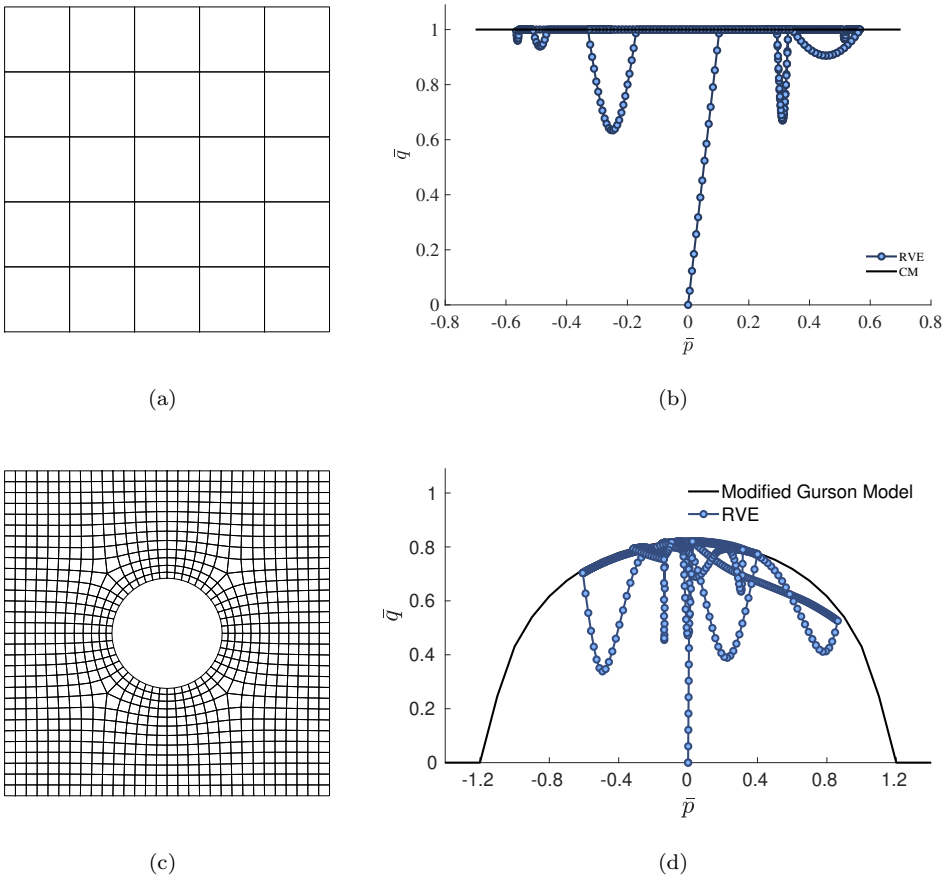


Figure 4.4.1: (a) Solid block RVE, (b) the associated yield surface, (c) RVE with a centre circular hole and (d) the associated yield surface.

However, the case differs in porous materials as the present space is responsible for a compression cap of plasticity which results in lowering the yield stress value. Different void ratios leads to different yield surfaces. Figure 4.4.1 compares the different yield surfaces of a solid block RVE and a RVE with circular void in the center. Any point on the yield surface exhibits plastic deformation and any point below that exhibits elastic behaviour.

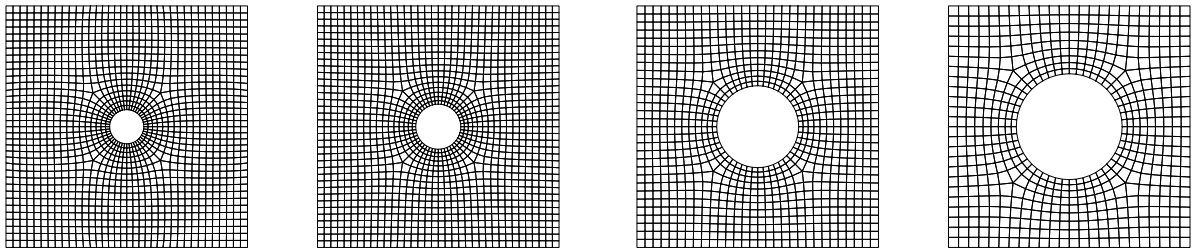
4.4.1 RVE Void Volume Fraction

In porous materials, the void volume fraction f can be calculated as

$$f = \sum_{i=1}^{n} \frac{\Omega_i}{V}, \tag{4.4.1}$$

where n is the number of voids in the RVE with a volume of Ω_i and V is the RVE total volume. When $f = 0$ the material is fully dense and $f = 1$ describes a fully voided material.

To study the influence of RVE linear and periodic boundary conditions on the homogenised yield surfaces, different void volume fractions $f = 2\%$, 5%, 10% and 20% are used as shown in Figure 4.4.2.



(a) $f = 2\%$ **(b)** $f = 5\%$ **(c)** $f = 10\%$ **(d)** $f = 20\%$

Figure 4.4.2: RVE geometries and finite element meshes.

4.4.2 Porous Elastic Materials

Analytical methods for porous elastic materials can be used to validate the computational multi-scale model. These methods are restricted to elastic homogeneous solids which contain a dilute distribution of cavities that are located adequately far apart to neglect the void interaction. A linear boundary conditions is used during the simulation.

Under the prescribed *macro-stress* assumption and plane stress conditions, the effective Young modulus $\bar{E}$ and Poisson ratio $\bar{\nu}$ [74] can be described as

$$\frac{\bar{E}}{E} = \Big(1 + 3f\Big)^{-1}, \tag{4.4.2}$$

$$\frac{\bar{\nu}}{\nu} = \Big(1 + f\frac{1}{\nu}\Big)\Big(1 + 3f\Big)^{-1}. \tag{4.4.3}$$

And, under the prescribed *macro-strain* assumption and plane stress conditions, the effective Young modulus $\bar{E}$ and Poisson ratio $\bar{\nu}$ [74] can be described as

$$\frac{\bar{E}}{E} = \Big(1 - f\frac{4}{1+\nu}\Big)\Big(1 - f\frac{2}{1-\nu}\Big)\Big(1 - f\frac{3-2\nu+3\nu^2}{1+\nu^2}\Big)^{-1}, \tag{4.4.4}$$

$$\frac{\bar{\nu}}{\nu} = \Big(1 + f\frac{1-6\nu+\nu^2}{\nu(1-\nu^2)}\Big)\Big(1 - f\frac{3-2\nu+3\nu^2}{1+\nu^2}\Big)^{-1}. \tag{4.4.5}$$

The determined results are presented in Tables 4.4.1 for plane strain and 4.4.2 for plane stress, a good level of agreement is found between the RVE and analytical solution.

Table 4.4.1: Macro-strain prescribed. (Plane strain) - For a material whose matrix properties are $E = 70GPa$ and $\nu = 0.2$ imposed with $\varepsilon_{xx} = 0.3$, $\varepsilon_{yy} = 0.5$ and $\varepsilon_{xy} = 0.4$.

	Analytical			**RVE**		
f	σ_{xx}	σ_{yy}	σ_{xy}	σ_{xx}	σ_{yy}	σ_{xy}
2%	31.9077	42.7966	10.8889	30.4756	41.125	10.6164
5%	30.2753	39.9976	9.7222	27.0406	36.5704	9.3652
10%	27.9630	35.7407	7.7778	22.205	30.3701	7.7743
15%	26.8981	32.7315	5.8333	18.3726	25.509	6.5463

Table 4.4.2: Macro-strain prescribed. (Plane stress) - For a material whose matrix properties are $E = 70GPa$ and $\nu = 0.2$ imposed with $\varepsilon_{xx} = 0.3$, $\varepsilon_{yy} = 0.5$ and $\varepsilon_{xy} = 0.4$.

	Analytical			**RVE**		
f	σ_{xx}	σ_{yy}	σ_{xy}	σ_{xx}	σ_{yy}	σ_{xy}
2%	27.8056	38.6944	10.8889	27.1137	37.71449	10.5648
5%	25.7639	35.4861	9.7222	24.3034	33.7379	9.2597
10%	22.3611	30.1389	7.7778	20.2482	28.2482	7.61521
15%	18.9583	24.7917	5.8333	16.8900	23.8735	6.36976

4.4.3 Gurson Model

In 1977, Gurson [41] proposed a macroscopic yield surface for porous metals. The model was based on the approximate limit-analysis of a hollow sphere made of a rigid-ideal plastic material that complied with the von Mises criterion and was subjected to the homogeneous boundary strain rate conditions [65, 43]. Many applications of the Gurson model can be seen in literature, such as [71, 13]. The mathematical expression proposed by Gurson [41] for defining the yield surface is as follows

$$\Phi = q - \left\{ \frac{1}{C_{eq}} \left[1 + f^2 - 2f \cosh\left(\frac{\sqrt{3}p}{\sigma_y} \right) \right] \right\}^{\frac{1}{2}} \sigma_y, \tag{4.4.6}$$

where

$$C_{eq} = (1 + 3f + 24f^6)^2. \tag{4.4.7}$$

where q is the hydrostatic component of the stress tensor, f denotes the RVE void volume fraction, p represents the von Mises equivalent stress and σ_y is the microscopic equivalent tensile yield stress of the matrix material.

A modification of the Gurson yield function has been later proposed by Tvergaard [102] (Gurson-Tvergaard equation). The modification included additional parameters, which take into consideration the influence of triaxiality and Lode parameter. This is to enhance the original Gurson function approximation of the yield surface when compared with numerical computation resulting in:

$$\Phi = \left(\frac{\sigma_e}{\sigma_y} \right)^2 + 2q_1 f \cosh\left(- \frac{3q_2}{2\sigma_y} \sigma_{kk} \right) - (1 + q_3 f^2) = 0. \tag{4.4.8}$$

where the new introduced three parameters are q_1 , q_2 , and q_3.

A comparison between the original Gurson model and the modified Gurson model yield surfaces is shown in Figure 4.4.3 in normalised $\bar{p}$-$\bar{q}$ space. This comparison includes different void ratios: $f = 2\%$, $f = 5\%$, $f = 10\%$ and $f = 20\%$. In addition, when setting $f = 0$, both models coincide with the standard J_2 theory, which follows the von Mises yield criterion. The modified Gurson model employs three parameters: $q_1 = 1.8$, $q_2 = 1.0$, and $q_3 = 3.24$. As the void volume fraction f increases, the material tends to yield at lower values of both von Mises stress and hydrostatic stress.

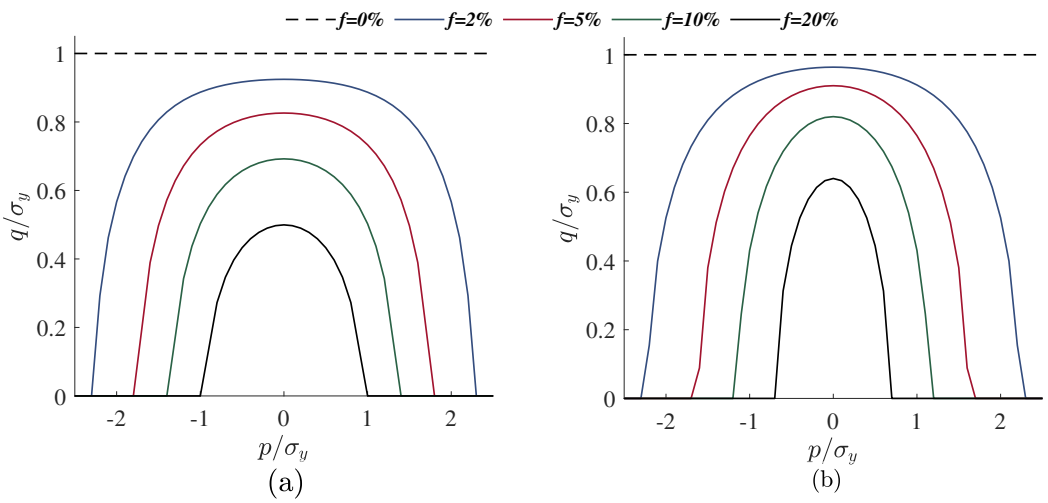


Figure 4.4.3: The yield surface of the (a) Gurson model and (b) modified Gurson model.

4.4.4 Yield Surface Approximation

The material properties of the ideally plastic RVE matrix are set as yield stress $\sigma_y = 240$, Young's modulus $E = 200GPa$ and Possion's ratio $\nu = 0.3$. The monotonic loading methodology consist of imposing the RVE with a prescribed macroscopic strain path expressed as

$$\boldsymbol{\varepsilon}(\gamma) = \gamma\bar{\boldsymbol{\varepsilon}}, \tag{4.4.9}$$

The loading is parameterised by a time factor γ, and a unit strain tensor that satisfies $|\bar{\boldsymbol{\varepsilon}}| = 1$ by

$$\bar{\boldsymbol{\varepsilon}} = \alpha \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \end{bmatrix} + \sqrt{1-\alpha^2} \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \tag{4.4.10}$$

where the prescribed α factor defines the direction of $\boldsymbol{\varepsilon}$ in the strain space and it varies between -1 and 1 (i.e. $-1 < \alpha < 1$). This allows covering the loading spectrum in a spherical direction when $\alpha = \pm 1$, pure shear strain direction when $\alpha = 0$ and a combination of both when α falls within that range.

Multi-scale RVEs with different void volume ratio f as presented in Figure 4.4.2 are utilised. For each f value, a finite element simulation is performed using 21 different values of α for all configurations to accurately construct the yield surface. The prescribed strain loading for the positive spectrum is shown in Table 4.4.3.

Table 4.4.3: Load factor α from 1 to 0.

α	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0
ε_{xx}	0.7071	0.6364	0.5657	0.4940	0.4243	0.3536	0.2828	0.2121	0.1414	0.0707	0
ε_{yy}	0.7071	0.6364	0.5657	0.4940	0.4243	0.3536	0.2828	0.2121	0.1414	0.0707	0
ε_{xy}	0	0.3082	0.4243	0.5050	0.5657	0.6124	0.6481	0.6745	0.6928	0.7036	0.7071

The associated macroscopic stress tensor is obtained by averaging the finite element microscopic stress field over the deformed state of the RVE. The material yield surface can be recovered by determining the hydrostatic and von Mises components of the macroscopic deformed stress.

The Gurson yield surface in Equation (4.4.6) is compared to the estimated yield surface generated with four different RVE geometries (*i.e.* $f = 0.6\%$, $f = 3\%$, $f = 10\%$ and $f = 15\%$) under two different boundary conditions (*i.e.* linear and periodic boundary conditions). In Figure 4.4.4, the black solid line represents the Gurson model and the red dotted line is the response of loading the RVE with a prescribed macroscopic strain path. When assuming the upper bound is the linear boundary condition and the lower bound is the periodic boundary condition in porous materials, it can be observed that:

- As the void ratios increase, the gap between the estimated yield surfaces for the RVE under the two different linear and periodic boundary conditions also increases.
- The Gurson surface lies between the upper and lower bounds for higher void ratios. Pure hydrostatic stress cases are closer to the upper bound (linear boundary condition), while pure shear states are found to be closer to the lower bound (periodic boundary condition).
- With lower void ratios, for certain range of hydrostatic and von Mises equivalent stress, the Gurson surface exceed the yield limit of the upper bound. This indicates that under such circumstances, the Gurson criterion overestimates the yield strength of the porous metal.

The model approximated yield surface is compared to the results in [39] and good agreements is observed for both linear and periodic boundary conditions.

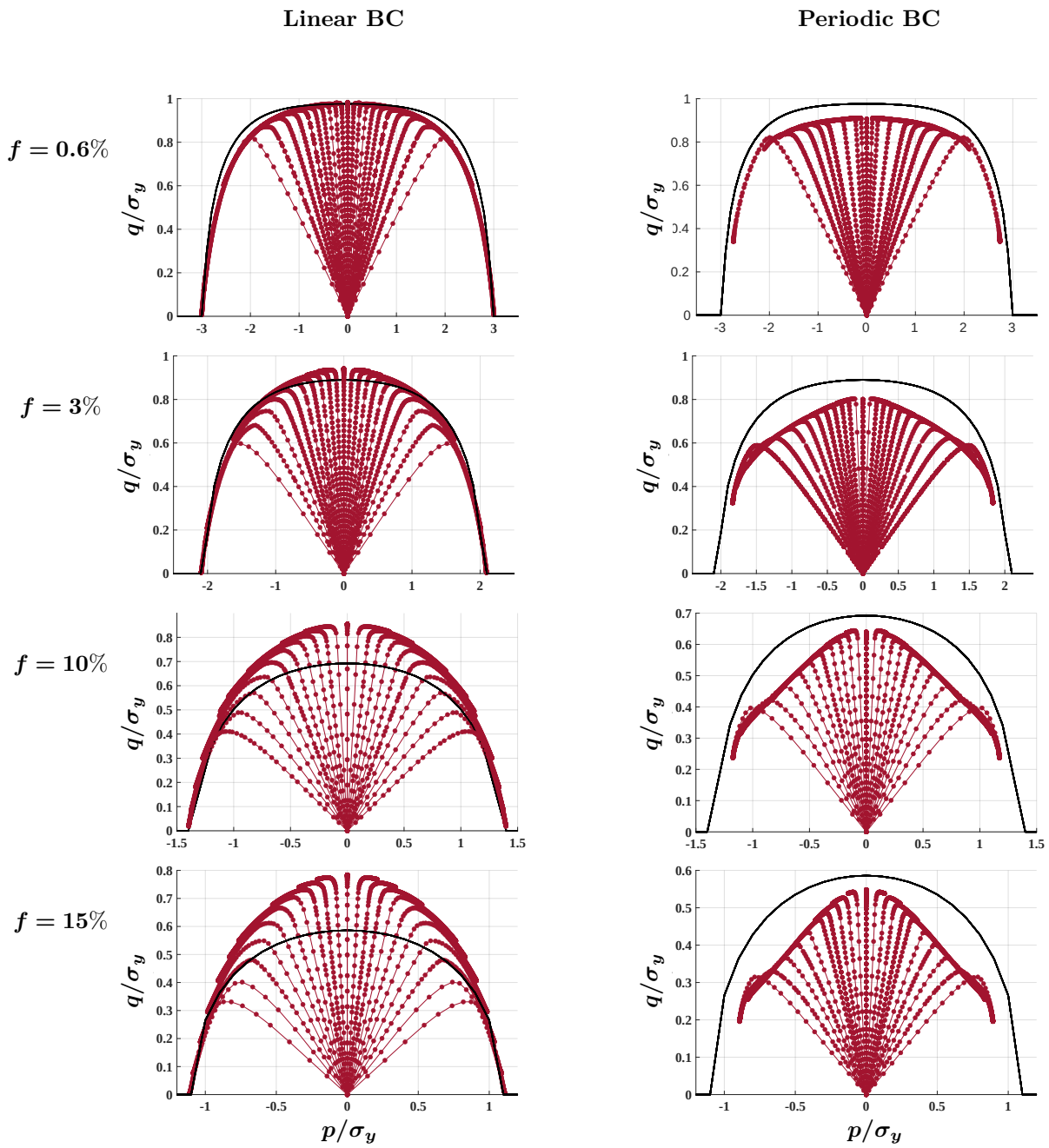


Figure 4.4.4: Gurson yield surface compared to different RVE geometries and boundary conditions.

Chapter 5

Machine Learning

Machine learning is a branch of computer science and Artificial Intelligence (AI) that enables computer algorithms to automatically learn, detect patterns and improve with more data. Machine Learning models can be used to solve for classification (unsupervised) or regression (supervised) problems. In recent years, neural networks have proven their ability to model complex relationships, in this case material behaviors.

This chapter is divided into three subsection which are neural networks in Section 5.1, time dependent problems in Section 5.2 and training with evolutionary optimisation in Section 5.3. The neural network subsection provides a brief introduction to the artificial neural network structure, neuron elements as well as the training and validation techniques for adequate model performance. The time dependent problems subsection addresses the challenges associated with training for time-dependent problems and present approaches to solve these challenges. The third subsection presents an alternative training method that involves using evolutionary optimization techniques.

5.1 Neural Networks

Neural networks are composed of connected neuron layers that mimic the internals of the human brain. The presence of components like activation functions allows it to model non-linear relationships. The basic ANN known as FeedForward Neural Network (FFNN) is used for static problems where the information flows

sequentially through neurons. The generic internal process of a neuron is described by

$$a_i = f(s_i) \quad \text{with} \quad s_i = \sum_j^R w_{ij}\, p_j + b_i \tag{5.1.1}$$

where a_i is the output of the neuron i, $f(\bullet)$ is the activation function, R is the number of inputs, w_{ij} are the weights connected to the inputs p_j and b_i is the bias term.

ANNs that consist of more than one hidden layer is known as Deep Neural Network (DNN) [36]. DNN have shown promising capabilities in solving number of large-scale complex real-world problems including language processing [22] and image recognition [21]. However, the multi-layer nonlinear structure of DNN makes it difficult to understand how the model arrived at a particular decision [86].

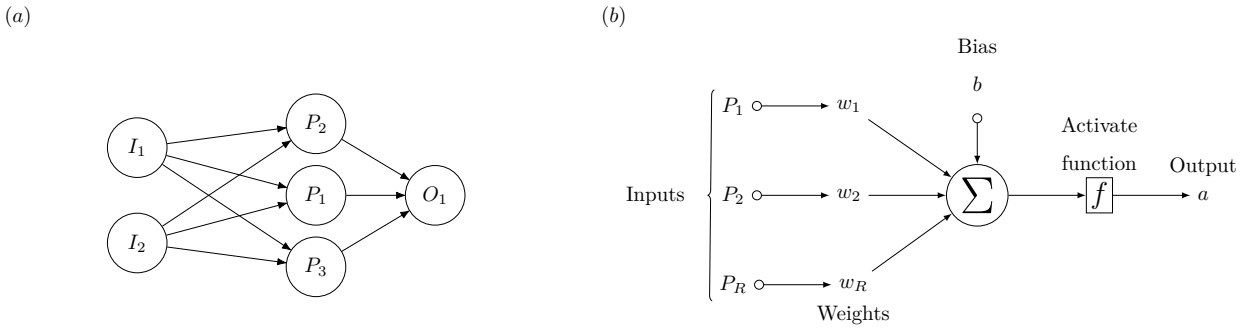


Figure 5.1.1: (a) ANN structure with 2 inputs, 1 hidden layer with 3 nodes and 1 output. (b) Artificial neuron elements.

In order to enhance the network learning, improve prediction performance and reduce computational time it is critical to clean and standardise the input data prior to a training procedure [96]. The model training process can be summarised in the following steps:

1) **Input layer:** The standardised input data is processed with a given combination of weights and biases.

2) **Hidden layer:** The weighted combinations are then routed through an activation function, a mathematical equation that determines the output of a layer. Depending on the problem, there are multiple types of activation functions that can be utilized. The three most commonly used activation functions are shown in Figure 5.1.2 and can be expressed as

- The Rectified Linear Unit (ReLU)

$$f(x) = max(0, x). \tag{5.1.2}$$

- The logistic (Sigmoid)

$$f(x) = \frac{1}{1 + e^x}. \tag{5.1.3}$$

- The hyperbolic tangent (Tanh)

$$f(x) = \tanh x. \tag{5.1.4}$$

3) **Output layer:** Return the computed model output.

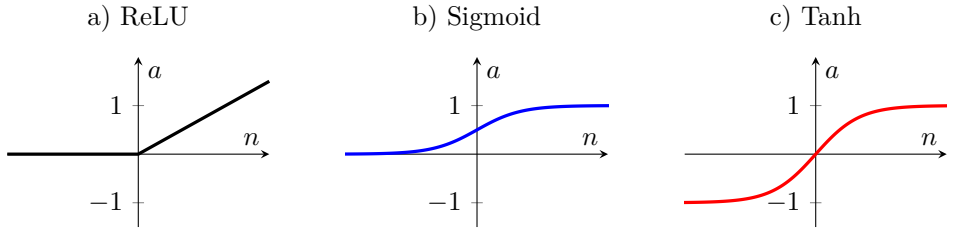


Figure 5.1.2: The most used activation functions.

5.1.1 Training and Validation:

The classic technique for training a neural network is the use of gradient-based algorithms such as back-propagation, which was established by Rumelhart et al. [85]. The back-propagation approach minimises the error function, the difference between predicted and actual values, by updating the weights based on the steepest descent direction. The network performance depends on the selected network architecture and hyper-parameters such as learning rate.

For optimal model performance, it is essential to well train the model to capture crucial relationships and perform well on both familiar (training) and unfamiliar (validation) data. Underfitting in a neural network occurs when the model is too basic to effectively understand the data, resulting in unsatisfactory performance on both training data and validation data. While, further training enhances the model performance for both training and validation data. Overfitting occurs when the model is excessively trained, fitting the training data too closely and struggling to generalise effectively to new, validation data

as shown in Figure 5.1.3. The concept of model fit is graphically illustrated in Figure 5.1.4.

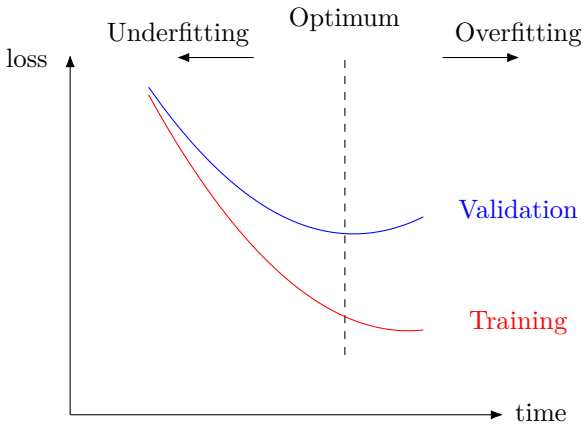


Figure 5.1.3: The learning curve.

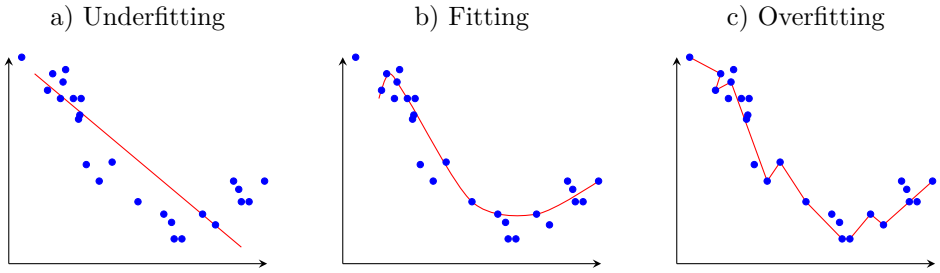


Figure 5.1.4: The underfitting and overfitting problems caused while training.

To avoid underfitting and overfitting, a stopping criteria can be use. Examples of such criteria include: Setting a maximum number of epochs or a computational time limit. Another technique to address overfitting and improve generalisation is using the dropout method where random neurons in the network are switched off at different epochs. The drawback of the dropout method is that an increased number of epochs would be needed for the model to converge.

The performance of the neural network model can evaluated with appropriate error metrics such as Mean Squared Error (MSE), which calculates the average of the square of the difference between the target values and the predicted values.

5.2 Time Dependent Problems

Time dependent behavior is found in constitutive modelling of engineering materials that exhibit irreversible deformation. Examples of this include, the constitutive law of plastic hardening [92], visco-elasticity [48] and visco-plasticity [93].

The classical approach to numerically understand these materials mechanical properties is by introducing internal state variables, which estimate deformation history [47]. Recently, machine learning techniques like neural networks have been used to understand material behaviors. While, the standard neural network performance in solving time dependent problems is less effective [119, 118], recurrent neural network architectures [42, 40, 72] include memory elements which can help store material state.

5.2.1 Recurrent Neural Network (RNN)

RNNs are mainly used to process sequences of data. Where the output from the first sequence is stored and then fed back along with inputs for the following sequence. The model can maintain a hidden state h over time that gets updated with every sequence. The limitation of RNN are the computation time is considerably slow due to the recurrent procedure, training is complicated and the vanishing and exploding gradients problem that occurs in long-term dependency problems (details in Section 5.2.2).

5.2.2 Vanishing/Exploding Gradients Problem (VEGP)

The learning process of RNN utilise the gradients, which represents the derivative of the loss function with respect to the weights. The weight values are then updated during back propagation to minimise the loss function. The vanishing gradient problem occurs when the norm of the gradient value is very small making it challenging to learn long-term dependencies and increasing the computational time [78]. In contrast, the exploding gradient problem occurs when the gradient value is very large making it difficult to find the global minima solution.

Several gradient based methods to address the VEGP have been tested, including gradient clipping, long short-term memory in Section 5.2.3 and gated recurrent unit in Section 5.2.4. Alternatively, gradient free optimisation strategies such as particle swarm optimisation can be used.

5.2.3 Long Short-Term Memory (LSTM)

The LSTM unit includes a state cell c_t, which is a way to store, process and pass information about previous time-steps to the end of the network. The information flow is maintained with three multiplicative gate units, specifically forget gate, input/update gate and output gate that are used to observe the information and control the error of the unit cell.

LSTMs improve the model performance by enhancing the network learning capability. While, increasing the model complexity compared to standard RNN [46], due to the two additional input and output gates that are needed for each memory cell.

5.2.4 Gated Recurrent Unit (GRU)

Unlike LSTMs, GRUs include two gates, reset gate r_t and the update gate z_t only. With the use of the update and reset gates GRUs internal memory is capable of storing and filtering important related information that improves the network performance. The weights are updated using back propagation through time (BTT) with stochastic gradient descent to minimize the loss function [37, 46].

Similar to LSTM, the GRU is suited for sequential data [17], which include natural language processing [4], short-text conversation [89], and time series [14].

5.3 Evolutionary Optimisation

To enhance the neural networks ability to find the optimum solution quickly, researchers have proposed training strategies coupled with nature inspired evolutionary optimisations methods, such particle swarm optimisation, cuckoo search, and ant colony optimisation. These hybrid method have been applied to different classification and prediction problems (more details are shown in Table 5.3.1). In addition, they could be an alternative approach to avoid the vanishing/exploding gradient problems.

In general, the meta-heuristic algorithms that depend on the search process can be divided into a single solution based, and population based algorithms. The followed search process in the former is initiated with an individual potential solution (*i.e.* search agent or object) and the solution improves over a specific number of iterations. Whereas, the latter search process is initiated with a defined value of members and each member is considered to be a search agent in the solution space. The group of potential solutions get

improved over defined number of iterations and the one with optimal solution is selected based on good fitness performance.

The two principal components in the search methodology are exploration and exploitation. The exploration component measure the algorithm ability in search the entire space for a better potential solution and mainly utilised at the initial stages of the search [77]. While, the exploitation component measure the algorithm ability to search a promising region for a better solution within the neighborhood in the search space which is performed with iterations to achieve the optimal solution. A well suited balance between the two search behaviors is required as the success of these algorithms highly depends on them.

Table 5.3.1: Training ANN with meta-heuristic algorithm.

Training ANN with	Advantages	Problems	References
Particle Swarm optimisation (PSO)	Enhance classification accuracy Reduce training time	Fruit classification Iris classification Medical field	[64, 116, 115]
Genetic Algorithm (GA)	Prevent premature convergence Determine weight values and number of nodes	ECG beat classification Land cover classification	[26, 62]
Artificial Bee Colony (ABC)	Dimensionality reduction optimise FNN parameters	DNA microarray classification MR brain image classification Medical field	[33, 117, 56]
Improved Cuckoo Search (ICS)	Improve the accuracy and convergence rate	Breast cancer classification Iris classification	[103]
Ant Colony optimisation (ACO)	optimisation of weight Good initial weights	Pattern classification Medical field	[8, 67]

Chapter 6

Exact Representation of Uniaxial Elastoplasticity

The study is restricted to rate-independent models which can have a purely elastic or a combination of elastic and plastic (elastoplastic) behaviors. The fundamental challenge associated with inelastic materials is the path-dependent material characteristics which means that the stresses depend on current deformation and on deformation history. Hence, neural network training should be in terms of data sequences rather than individual input-output data pairs. To achieve this, two potential solutions have been proposed which are: History based and Internal variable based models. The input to the history based model are the stress and strain histories. While, the input to the internal variable based model is the new material state which depends on previous state. Both proposed solutions feed the output of the network as the next stage input. The flowchart in Figure 6.0.1 illustrates the process described above.

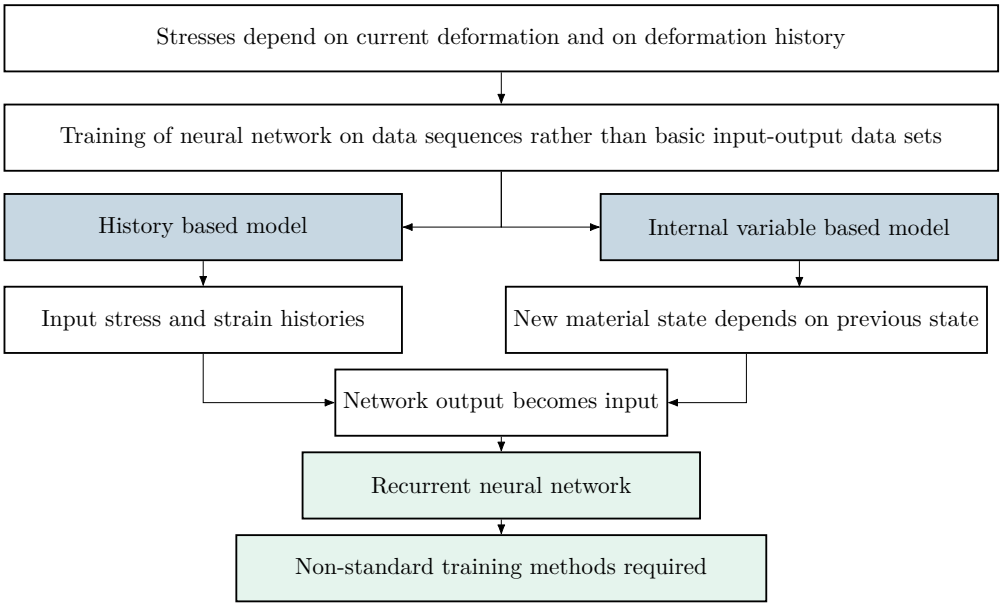


Figure 6.0.1: Proposed neural network training process for inelastic material.

6.1 History-Based Strategy

The behavior of rate-independent inelastic solid materials depend on the historical values of stresses and strains. These historical values can be represented in an RNN architectures to estimate the material mechanical behaviour. However, the process of training RNNs using gradient-based methods can be affected by the vanishing and exploding gradient problem [82], leading to unstable training. In order to avoid the occurrence of this problem, a simplified architecture is adopted which is similar to the Nonlinear Autoregressive Network with Exogenous Inputs (NARX). The network can be trained as a simple static feed-forward network. Moreover, the model includes feedback connections that relate multiple network layers. And to generate the input series, two different architecture modes which are the Series-Parallel (SP) mode and Parallel (P) mode [68] are employed.

The NARX (SP) mode, also known as the open-loop form, is utilised for training neural networks as it enables estimating the time-series for the next time-step. This mode is only employed during the first iteration of the training process. The sequence of past input $u(t)$ and output $y(t)$ values can be expressed as

$$\hat{y}(t+1) = f(\boldsymbol{x}(t)), \tag{6.1.1}$$

where $\hat{y}(t+1)$ is the predicted output for the next time step $t+1$, the employed activation function is $f(.)$ and $\boldsymbol{x}(t)$ is the regressor vector that consist of a finite number of previous inputs and outputs:

$$\boldsymbol{x}(t) = [y(t), y(t-1), ..., y(t-n_y), u(t+1), u(t), u(t-1)..., u(t-n_u)]^T. \tag{6.1.2}$$

Once the (SP) mode training process is completed, the NARX-based neural model adopts the (P) mode which is also known as the close-loop form. The close loop form predicts the next values of the target series by redirecting the output back $(y(t), y(t-1), ..., y(t-n_y))$ to the input layer. In addition, the given values of an independent input $u(t)$ are also introduced into the network as a history input vector that corresponds with each output $y(t)$. The lengths n_y and n_u need to be defined in line with the data and the problem restrictions. There is a trade off between network complexity (size) and its prediction accuracy. To minimize the network size and improve computational efficiency, one can use less historical values which results in less accurate predictions. On the other hand, in order to achieve higher accuracy additional historical values could be considered. The sequence of the history inputs in this case can be expressed as

$$\hat{y}(t+1) = f(\hat{y}(t), \hat{y}(t-1), ..., \hat{y}(t-n_y), u(t+1), u(t), u(t-1)..., u(t-n_u)). \tag{6.1.3}$$

A schematic representation of the history-based neural network in terms of stresses and strains is shown in Figure 6.1.1. The predicted next step stress σ_{n+1} is fed back to the input of the feed-forward neural network. Hence, training with the simplified feed-forward network architecture.

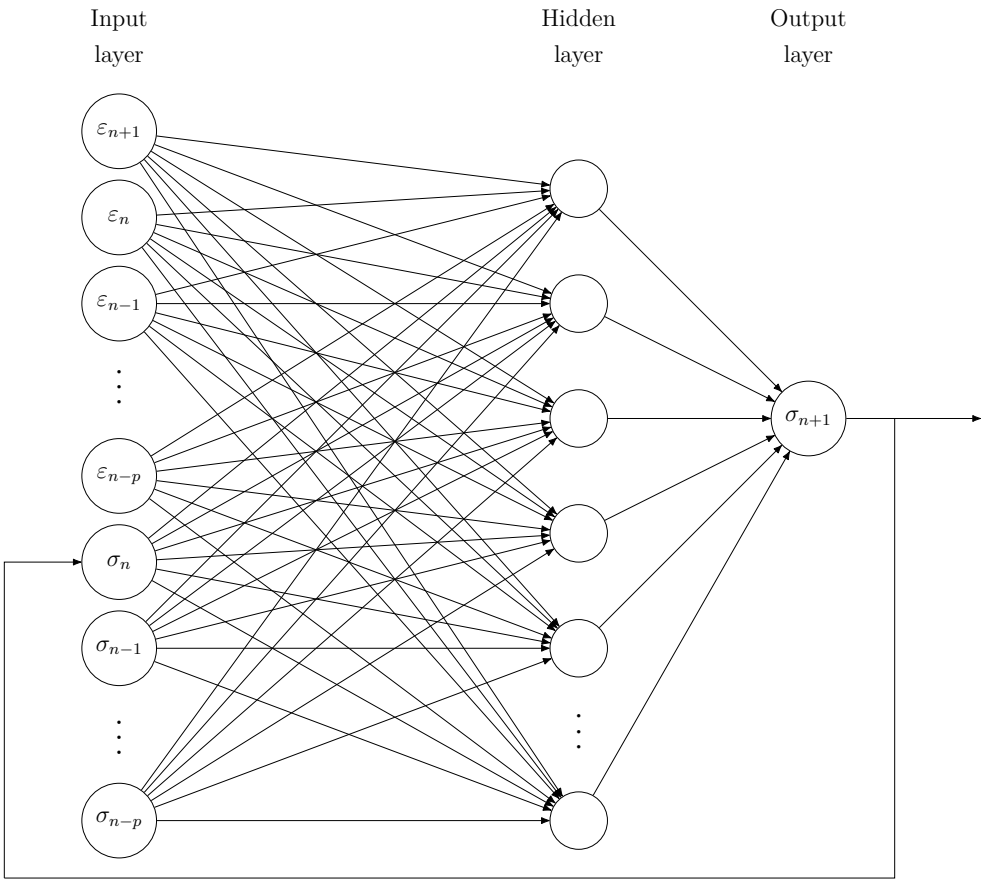


Figure 6.1.1: History-based strategy in terms of stresses and strains.

The adopted neural network architecture significantly influence the model performance in training and validation. Hence, a case study on neural architectures was carried out to obtain the optimal network architecture in terms of modeling accuracy, while requiring minimum number of hidden layers and nodes per layer. Furthermore, this study illustrates how the complicated elastic-plastic behaviour can be represented using linear algebra operations and neural network model assessment.

1D ideal elastoplasticity without hardening: The proposed neural network indicates that for the history-based strategy with no hardening, a history of 1 is required to reconstruct the elastoplastic constitutive model behaviour. Figure 6.1.2 demonstrates the ideal minimum network architecture with defined weights displayed on the arrows, and biases shown inside the circles which represent the hidden neurons. The network generates identical constitutive model predictions and is characterised with the following:

- The 3 inputs ε_{n+1}, ε_n and σ_n are inserted into the input layer.
- The node connectivity is then modified by employing the right combination of weight W_{ij} and bias b_i parameters. The weighted sum is subsequently sent to the next layer, the first hidden layer, which consists of 4 neurons.
- The Rectified Linear Unit is chosen as the hidden layer activation function.
- A linear activation function is employed for the output layer, and the given weights for the output layer are used to adjust and calculate the following step σ_{n+1}.

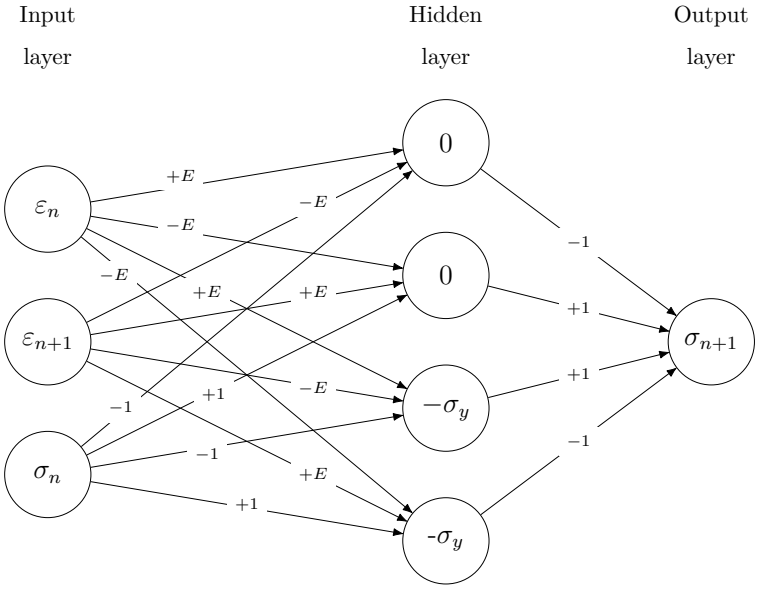
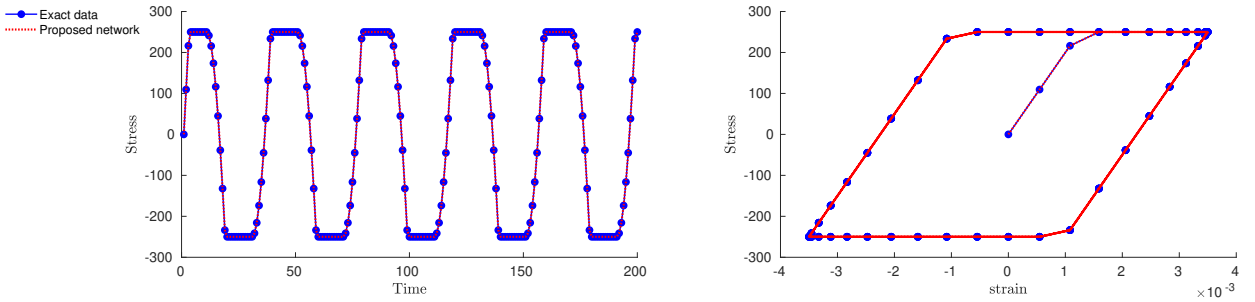


Figure 6.1.2: 1D ideal linear elastoplasticity model in a network form.

The material properties of steel A36 were used in all the examples presented in this chapter. The Young's modulus is set as E= 200 GPa, when isotropic hardening modulus is introduced to the problem it is given a value of H= 10 GPa and yield stress σ_y= 250MPa.

(a) Cyclic strain



(b) Random strain path

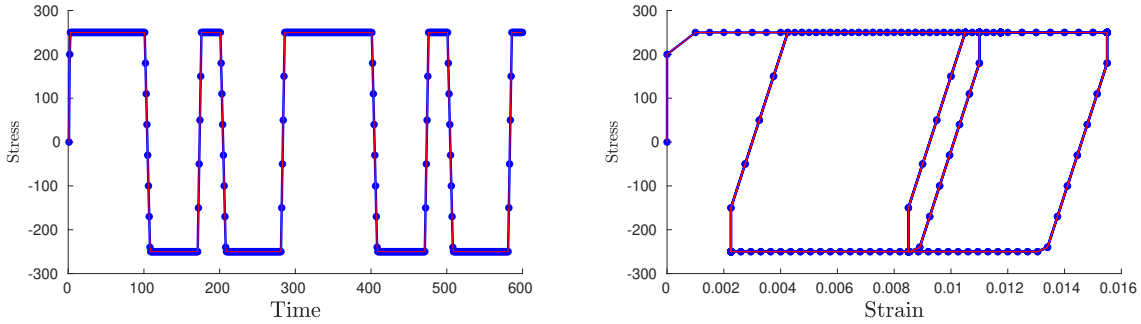


Figure 6.1.3: The performance of proposed history based strategy without hardening presenting stress over time on the left and the stress and strain relationship on the right.

The prediction of the defined network in Figure 6.1.2 for history-based strategy compared to the exact analytical solution computed with the standard return mapping algorithm is shown in Figure 6.1.3. Where, the red dashed line represent the proposed network output and the solid blue is the exact data generated by the constitutive model. An exact representation can be observed for both cyclic strain and random strain path cases in Figure 6.1.3.

1D elastoplasticity with hardening: When isotropic hardening is introduced, the level of complexity increases. This is due to the internal hardening variable α, which should be described in terms of historical strains and associated stresses. The study demonstrated that to obtain adequate estimation of the elastoplastic behavior while considering hardening a history value of 2 is needed. The derivation process for determining the equivalent term of α is briefly described:

- First, the total stresses during plasticity can be calculated by

$$\begin{aligned}\sigma_{n+1} &= \sigma_{n+1}^{trial} - \frac{E}{E+H}\text{sign}(\sigma_{n+1}^{trial})\Big(\big|\sigma_{n+1}^{trial}\big| - (\sigma_y + H\alpha_n)\Big),\\ \sigma_n &= \sigma_n^{trial} - \frac{E}{E+H}\text{sign}(\sigma_n^{trial})\Big(\big|\sigma_n^{trial}\big| - (\sigma_y + H\alpha_{n-1})\Big).\end{aligned} \tag{6.1.4}$$

- Then, solving for α_n yields

$$\alpha_n = \frac{\big|\sigma_n^{trial}\big| - \sigma_y + \text{sign}(\sigma_n^{trial})(\sigma_n - \sigma_n^{trial})}{H}, \tag{6.1.5}$$

- Finally, σ_{n+1} can be expressed in terms of historical strains and it corresponding stresses as

$$\sigma_{n+1} = \sigma_{n+1}^{trial} - \frac{E}{E+H}\text{sign}(\sigma_{n+1}^{trial})\left(\big|\sigma_{n+1}^{trial}\big| - \left(\sigma_y + H\Big(\frac{\big|\sigma_n^{trial}\big| - \sigma_y + \text{sign}(\sigma_n^{trial})(\sigma_n - \sigma_n^{trial})}{H}\Big)\right)\right). \tag{6.1.6}$$

The proposed neural network for 1D elastoplasticity when hardening introduced in Figure 6.1.4 could be summarised as

1) The input layer holds 5 inputs, including two historical stress and strain values:

$$\boldsymbol{x} = \begin{bmatrix}\varepsilon_{n+1} & \varepsilon_n & \varepsilon_{n-1} & \sigma_n & \sigma_{n-1}\end{bmatrix}^T. \tag{6.1.7}$$

2) As previously stated, the complexity level increases when isotropic hardening is introduced due to the associated non-linearity of the hardening parameter α. In this case, the defined total combination of weights is 40, whereas the total number of assigned biases is 8, as shown in Equation (6.1.8). The proposed neural network performance is not exact but with relatively acceptable level of accuracy compared to the 1D elastoplasticity material behavior described by the constitutive model. Furthermore, the selected activation function for the hidden layer is the ReLU.

$$
\mathrm{H} = \mathrm{ReLU}\left(\begin{bmatrix}
E & -E & 0 & 1 & 0 \\
-E & E & 0 & -1 & 0 \\
\frac{-E^2}{E+H} & \frac{E^2}{E+H} & 0 & \frac{-E}{E+H} & 0 \\
\frac{E^2}{E+H} & \frac{-E^2}{E+H} & 0 & \frac{E}{E+H} & 0 \\
0 & \frac{-E^2}{E+H} & \frac{E^2}{E+H} & 0 & \frac{-E}{E+H} \\
0 & \frac{E^2}{E+H} & \frac{-E^2}{E+H} & 0 & \frac{E}{E+H} \\
0 & \frac{-E^2}{E+H} & \frac{E^2}{E+H} & \frac{E}{E+H} & \frac{-E}{E+H} \\
0 & \frac{E^2}{E+H} & \frac{-E^2}{E+H} & \frac{-E}{E+H} & \frac{E}{E+H}
\end{bmatrix}\begin{bmatrix}
\varepsilon_{n+1} \\ \varepsilon_n \\ \varepsilon_{n-1} \\ \sigma_n \\ \sigma_{n-1}
\end{bmatrix} + \begin{bmatrix}
0 \\ 0 \\ \frac{-E\sigma_y}{E+H} \\ \frac{-E\sigma_y}{E+H} \\ \frac{-E\sigma_y}{E+H} \\ \frac{-E\sigma_y}{E+H} \\ 0 \\ 0
\end{bmatrix}\right) \tag{6.1.8}
$$

3) Finally, the output of the neural network is calculated with the with given combination of weights $W_{ij}^{(O)}$ and linear activation function in the output layer

$$
W_{ij}^{(O)} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \end{bmatrix}^T \tag{6.1.9}
$$

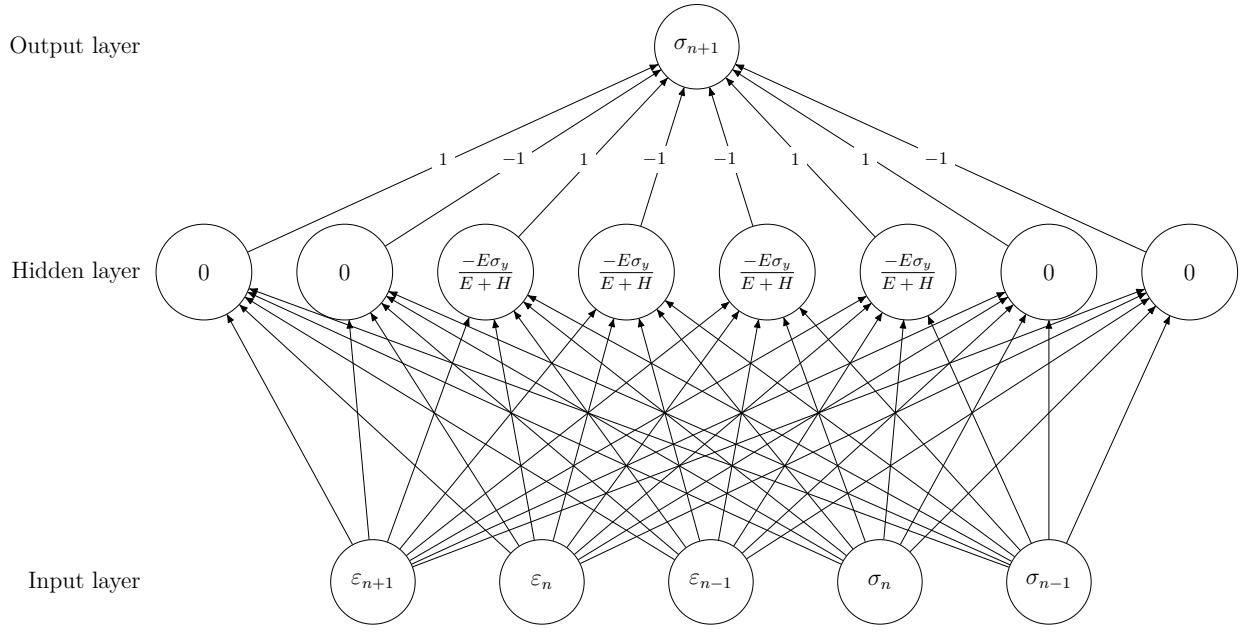
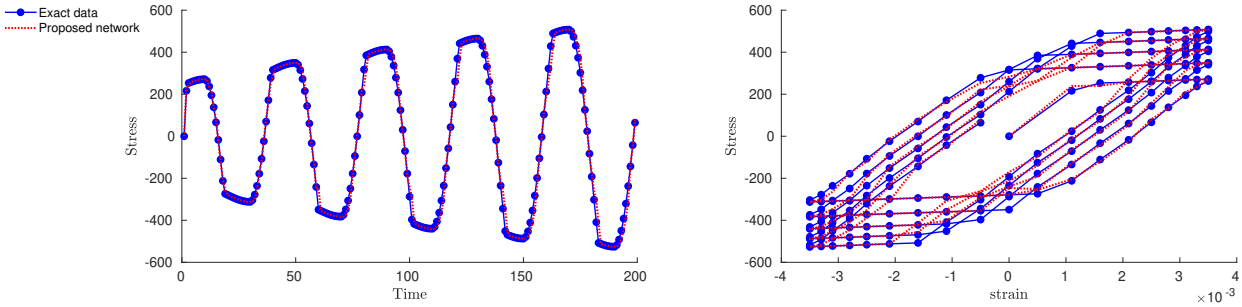


Figure 6.1.4: The proposed network form for 1D elastoplasticity with hardening model.

(a) Cyclic strain



(b) Random strain path

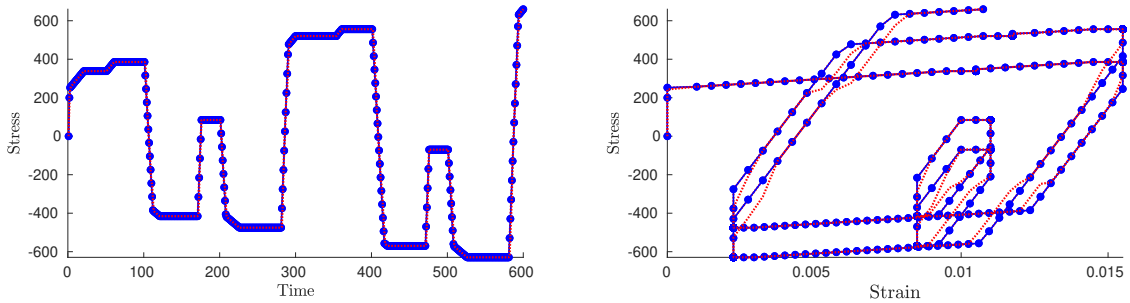


Figure 6.1.5: The response of the suggested architecture for history based strategy with hardening presented in stress over time (left) and stress and strain relationship (right).

As it can be observed from Figure 6.1.5 the behavior of the material during elastic and plastic phases can be reproduced with adequate accuracy. However, the predicted behavior for the transition from elastic to plastic phase is relatively poor. In order to improve the performance of the network response, it is necessary to include additional historical stresses and strains. The increase in the historical values may results in expensive computational cost.

6.2 Internal Variable-Based Strategy

Since internal variables are not directly observable, the classic approach to define material properties is by applying the plasticity theory, which consist of a conventional system of equations including the yield criterion and its associated flow rule. The proposed internal variable-based strategy takes only observable data as input to train the network. And the changes in internal variable values can be extracted as a result of the training procedure. The illustrative examples presented in this section assumes that the internal variables

are equivalent to the material plastic strain.

The widely used return mapping algorithm for von Mises elastoplasticity with linear isotropic hardening is described in [91, 23]. For the uniaxial case, the following expressions for the update of the stress and the internal variables are used

$$\begin{aligned}
\sigma^{\text{trial}} &= E\left(\varepsilon_{n+1} - \varepsilon_n^{(p)}\right), && (6.2.1)\\
\Phi^{\text{trial}} &= |\sigma^{\text{trial}}| - (\sigma_y + H\,\alpha_n), && (6.2.2)\\
\Delta\gamma &= \left\langle \frac{\Phi^{\text{trial}}}{E+H} \right\rangle, && (6.2.3)\\
\Delta\varepsilon^{(p)} &= \Delta\gamma\,\text{sign}(\sigma^{\text{trial}}), && (6.2.4)\\
\Delta\alpha &= \Delta\gamma, && (6.2.5)\\
\varepsilon_{n+1}^{(p)} &= \varepsilon_n^{(p)} + \Delta\varepsilon^{(p)}, && (6.2.6)\\
\alpha_{n+1} &= \alpha_n + \Delta\alpha, && (6.2.7)\\
\sigma_{n+1} &= E\left(\varepsilon_{n+1} - \varepsilon_{n+1}^{(p)}\right). && (6.2.8)
\end{aligned}$$

where E, σ_y and H are, Young's modulus, yield stress and hardening modulus, respectively. The plastic strain is represented by $\varepsilon^{(p)}$, while the plastic increment is denoted by $\Delta\gamma$ and α denotes the accumulated plastic strain. The ramp function is defined as $\langle\bullet\rangle = \max(0, \bullet)$.

1D ideal elastoplasticity without hardening: The return mapping algorithm is used to develop the proposed neural network constitutive model. For perfect elastoplasticity with $H = 0$ the accumulated plastic strain α can be eliminated. Hence:

$$\Delta\varepsilon^{(p)} = \text{sign}\left(\varepsilon_{n+1} - \varepsilon_n^{(p)}\right)\ \left\langle \left|\varepsilon_{n+1} - \varepsilon_n^{(p)}\right| - \frac{\sigma_y}{E} \right\rangle\ , \tag{6.2.9}$$

which is equivalent to

$$\Delta\varepsilon^{(p)} = +\left\langle \varepsilon_{n+1} - \varepsilon_n^{(p)} - \frac{\sigma_y}{E} \right\rangle - \left\langle \varepsilon_n^{(p)} - \varepsilon_{n+1} - \frac{\sigma_y}{E} \right\rangle\ . \tag{6.2.10}$$

The study concluded that only one internal variable ξ is needed to reproduce 1D elastoplasticity without hardening. The following is an explanation of the steps used by the proposed two network formulation model presented in Figure 6.2.1:

1) First the state network $\mathscr{G}$ computes the plastic corrector in the return mapping procedure. The network consists of an input layer with 2 neurons then 1 hidden layer with 2 neurons and finally an output layer with 1 neuron. The neurons in the hidden layers have ReLU activation functions while the neuron in the output layer has a linear activation function. The total strain ε_{n+1} and internal variable which is equivalent to plastic strain *i. e.* $\xi = \varepsilon_n^p$ are fed into the input layer and the network computes an incremental plastic strain $\Delta\varepsilon^p$. It is essential to mention that in this case the historical strain ε_n is not used as an input to the state network $\mathscr{G}$.

2) Then the next step plastic strain ε_{n+1}^p is updated by adding the internal variable, representing the plastic strain, ε_n^p and the output of incremental plastic strain $\Delta\varepsilon^p$ from state network $\mathscr{G}$.

3) Following that, the updated internal variable of ε_{n+1}^p is given to the second neural network, which is the response network $\mathscr{F}$ that performs the elastic predictor step in the return mapping algorithm. The response network $\mathscr{F}$ consist of two inputs of ε_{n+1} and ε_{n+1}^p connected with a linear activation function to an output layer holding the desired updated stress σ_{n+1}.

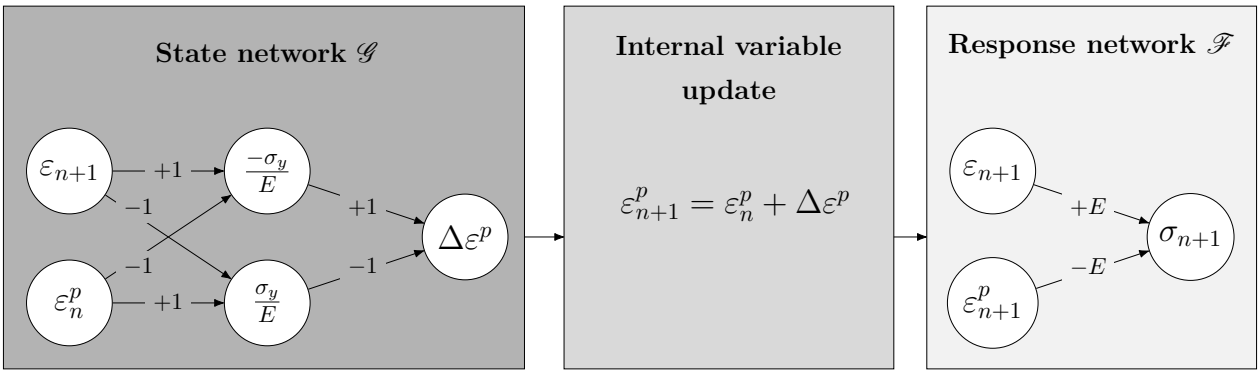
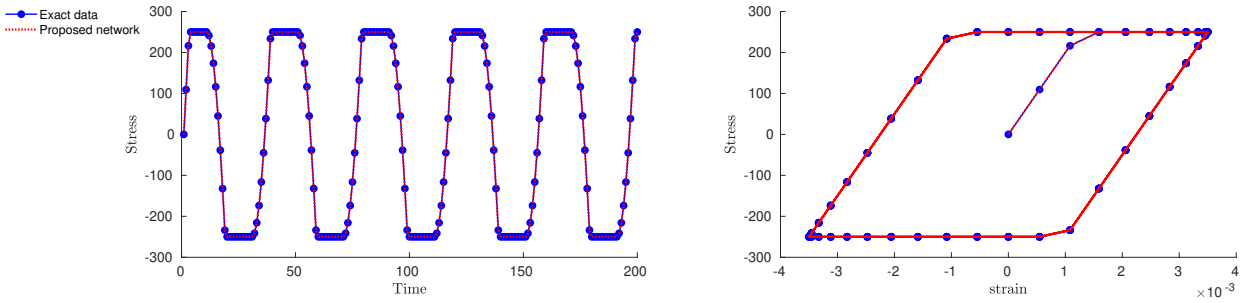


Figure 6.2.1: Two neural network formulation of 1D ideal linear elastoplasticity model.

In order to evaluate the proposed strategy, Figure 6.2.2 is used to illustrate a standard example of the well known cyclic strain and random strain path behaviours for 1D linear elastoplasticity. In which, the neural network prediction defines exactly the stress evolution in both cases.

(a) Cyclic strain



(b) Random strain path

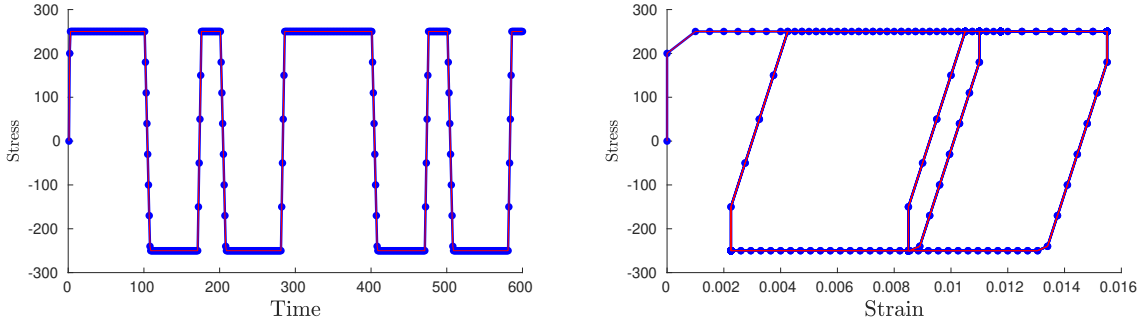


Figure 6.2.2: The two network formulation performance for 1D linear elastoplasticity presented in stress over time on the left and stress and strain relationship on the right.

1D elastoplasticity with isotropic hardening: When introducing isotropic hardening *i. e.* $H > 0$ to the problem, the network would need two internal variables ξ to reproduce the constitutive relationship. Similar procedure is followed to formulate the proposed neural network in Figure 6.2.3 and could be summarised as 1) The state network $\mathscr{G}$ takes three inputs namely the total strain ε_{n+1} and the two internal variables that are equivalent to the plastic strain $\xi_1 = \varepsilon_n^p$ and hardening parameter $\xi_2 = \alpha_n$. The input layer is linked to one hidden layer that includes two neurons with a ReLU activation function. In addition, the state network $\mathscr{G}$ in Figure 6.2.3 is characteristics with 12 weights shown on the arrows and 2 given biases displayed inside the hidden neuron circles. The computed results are then passed to the output layer which includes two output neurons of $\Delta\varepsilon^p$ and $\Delta\alpha_n$.

2) Following that, the two internal variables of plastic strain ε_n^p and α_n are updated by adding them to the incremental output computed in the first network as following:

$$\varepsilon^P_{n+1} = \varepsilon^P_n + \Delta\varepsilon^p. \tag{6.2.11}$$

$$\alpha_{n+1} = \alpha_n + \Delta\alpha_n. \tag{6.2.12}$$

3) Next, the updated internal variables are transferred to the response network $\mathscr{F}$ which performs a linear operation to represent the elastic predictor in the return mapping algorithm. The network consists of 3 inputs of ε_{n+1}, ε^p_{n+1} and α_{n+1} that are connected linearly with the updated stress σ_{n+1} in the output layer.

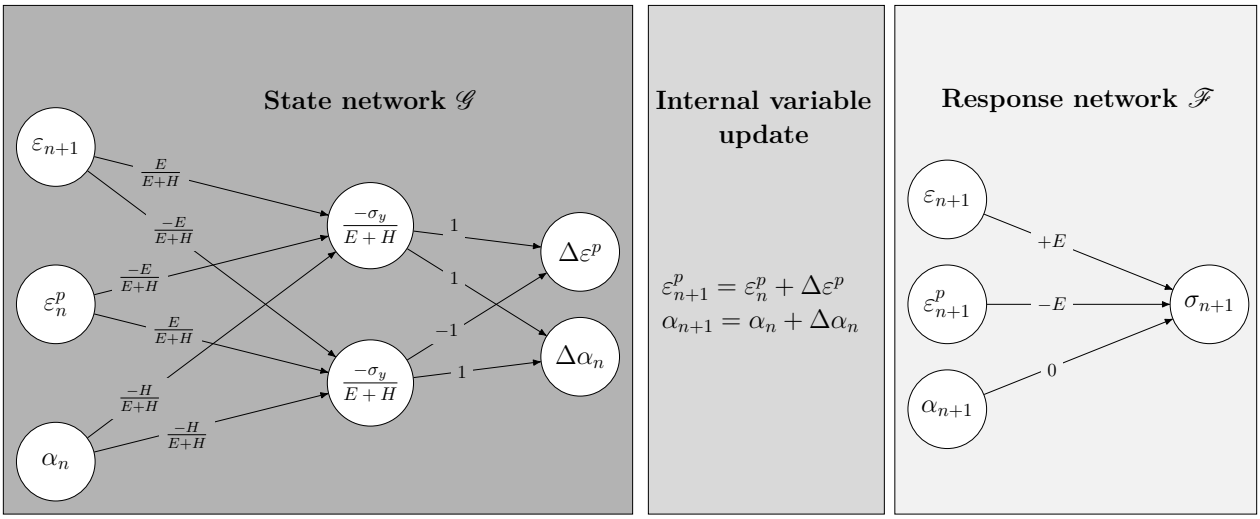
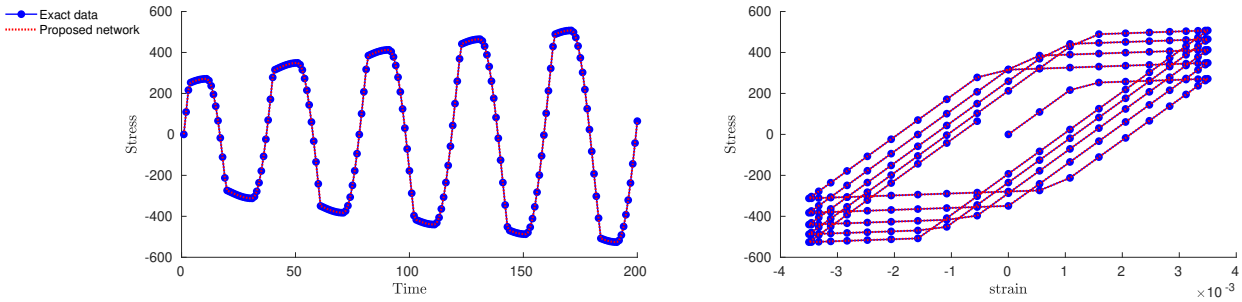


Figure 6.2.3: 1D elastoplasticity with hardening model in a network form.

With the given architecture and defined combination of weights and biases in Figure 6.2.3 an identical stress response could be achieved for the symmetric strain cycling and random strain path as illustrated in Figure 6.2.4. The behaviour is displayed in terms of stress over time and stress/strain relationship with the constitutive model data displayed in solid blue and the proposed network output represented in dashed red.

(a) Cyclic strain



(b) Random strain path

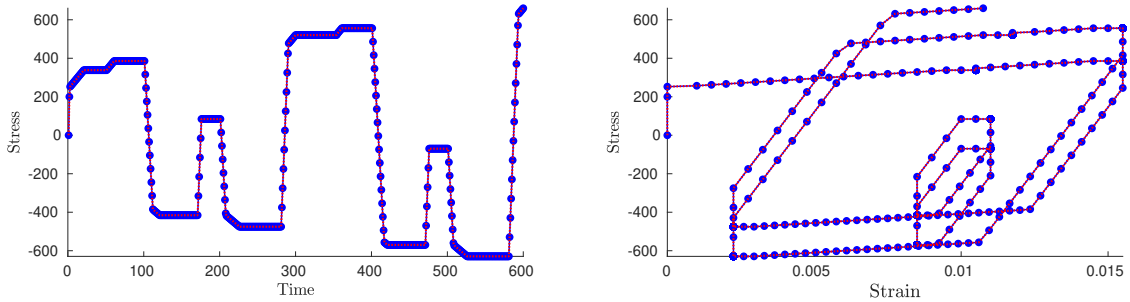


Figure 6.2.4: The proposed network response with internal variables strategy for 1D elastoplasticity with hardening shown in terms of stress over time (left) and stress and strain relationship (right).

6.3 Interpretation

Two neural network-based strategies, history-based and internal variable, have been utilised to mimic the complicated non-linear constitutive relations of solid materials with adequate computing efficiency. The data used to assess the network performance were generated using the von Mises model and implemented using the return mapping technique for elastic and plastic deformation.

The networks were designed to reproduce the behavior of the established constitutive model exactly. For history-based with isotropic hardening an acceptable level of accuracy is obtained but inaccuracies are observed when the direction of the strain loading is changed. Therefore, the performance of the internal variable-based strategy is more suited to capturing the behaviour of elastoplastic type materials. The results were achieved using theoretically determined weights and biases in the proposed optimal minimum designs for both strategies, rather than neural network training. Moreover, because to differences in scaling, equivalent network performance might be obtained with different combinations of weights and biases.

Chapter 7

Generalised Neural Network Based Constitutive Model

The remainder of this work follows the proposed neural network based algorithmic constitutive model with internal variables. When using the methodology, no prior knowledge of the material is needed. The values and evolution of the internal variables are obtained by the training procedure and therefore have no physical interpretation for them.

7.1 Neural Network Based Stress Update

The stress tensor $\boldsymbol{\sigma}$ depends on the strain tensor $\boldsymbol{\varepsilon}$ and the scalar internal variables $\boldsymbol{\xi} = \{\xi^{(1)}, \xi^{(2)}, ..., \xi^{(M)}\}$ can be defined as

$$\boldsymbol{\sigma} = \mathscr{F}\left(\boldsymbol{\varepsilon}, \boldsymbol{\xi}\right) \ , \tag{7.1.1}$$

where $\mathscr{F}$ denotes a suitable feedforward artificial neural network. For rate-independent materials the update of the internal variables from time instant t_n to time instant t_{n+1} can be expressed as

$$\boldsymbol{\Delta\xi} = \mathscr{G}\left(\boldsymbol{\varepsilon}_{n+1}, \boldsymbol{\varepsilon}_n, \boldsymbol{\xi_n}\right), \tag{7.1.2}$$

and

$$\boldsymbol{\xi}_{n+1} = \boldsymbol{\xi}_n + \boldsymbol{\Delta\xi} \ , \tag{7.1.3}$$

where $\mathscr{G}$ represents a second suitable feedforward artificial neural network. Given the new strains $\boldsymbol{\varepsilon}_{n+1}$ and the historic quantities $\boldsymbol{\varepsilon}_n$ and $\boldsymbol{\xi}_n$, the complete stress update procedure consists of evaluating Equations (7.1.2) and (7.1.3) followed by applying Equation (7.1.1) to the quantities at t_{n+1}, *i.e.*

$$\boldsymbol{\sigma}_{n+1} = \mathscr{F}\left(\boldsymbol{\varepsilon}_{n+1}, \boldsymbol{\xi}_{n+1}\right) \ . \tag{7.1.4}$$

In the following, the Networks $\mathscr{G}$ and $\mathscr{F}$ are referred to as the *state* and the *response* networks, respectively.

7.2 Neural Network Architecture

The proposed artificial state and response neural networks $\mathscr{F}$ and $\mathscr{G}$ consist of an input layer, a small number of fully connected hidden layers and an output layer. This study focuses on small strain regime, hence, the response network $\mathscr{F}$ may not possess any hidden layers and correspond to a basic linear operator. The generic internal process of a neuron is described by

$$a_i = f(s_i) \quad \text{with} \quad s_i \ = \ \sum_j^R w_{ij}\, p_j + b_i \tag{7.2.1}$$

where a_i is the output of the neuron i, $f(\bullet)$ is the activation function, R is the number of inputs, w_{ij} are the weights connected to the inputs p_j and b_i is the bias term. In this work the piecewise linear function known as the Rectified Linear Unit (ReLU) is used.

$$f(s) \ = \ \max(0, s) \ . \tag{7.2.2}$$

The output neurons are linear operators, *i.e.* they are similar to neurons in the hidden layers but do not change the weighted sum of the inputs.

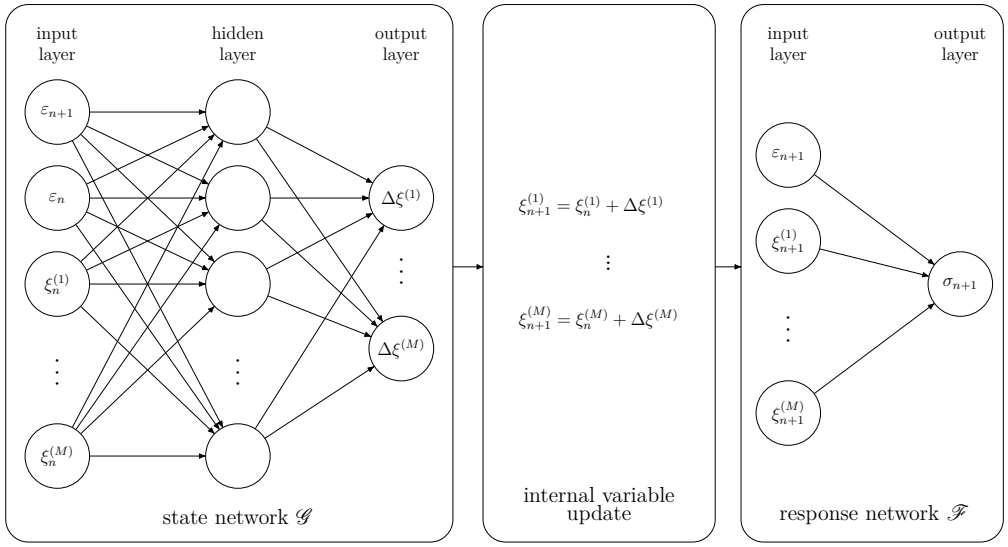


Figure 7.2.1: Stress update procedure based on the state and response neural networks $\mathscr{G}$ and $\mathscr{F}$.

The stress update procedure defined by Equations (7.1.2) to (7.1.4) is shown in Figure 7.2.1 for the uniaxial load case where stresses and strains are each represented by a single coefficient. In addition, the state network $\mathscr{G}$ possesses one hidden layer of neurons, while the response network $\mathscr{F}$ consists only of the input and output neurons. For the three dimensional case, the stresses and strains are represented by six coefficients each and the number of input and output neurons must be increased accordingly. For the plane strain case, three strain coefficients induce four stress coefficients and, again, the number of network inputs and outputs must be adjusted accordingly.

Remark on physical constraints: The networks $\mathscr{F}$ and $\mathscr{G}$ are subject to the following *physically* motivated constraints:

1. If all the given inputs are zero, all the generated outputs of the state network $\mathscr{G}$ must equal to zero. This causes the biases of the output neurons to be dependent on the biases of the hidden layers.
2. If all the given inputs are zero, all the generated outputs of the response network $\mathscr{F}$ must equal to zero. This results in a dependency of the biases of the output neurons on the biases in the hidden layers. In the absence of any hidden layers, the biases in the output neurons must equal to zero.
3. If the given new strain ε_{n+1} is identical to the previous strain ε_n (in all its components), then all outputs of the state network $\mathscr{G}$ must be equal to zero.

7.3 Recurrent Structure and Training

In order to determine an accurate representation of the material characteristics, the system must be *trained*. In addition, the training data used must accurately and comprehensively represent the material behaviour. The appropriate values for the weights and biases of the state and response networks $\mathscr{F}$ and $\mathscr{G}$ are obtained during the training process. Prior to designing an appropriate training technique the system overall nature must be considered carefully.

Training on data sequences: Although the neural networks $\mathscr{F}$ and $\mathscr{G}$ are introduced as feedforward networks, the *recurrent* nature of the overall strategy is apparent: The updated internal variables which result from network output of the previous load step are used as network input in the next load step. Moreover, the internal variables act as hidden states similar to those found in Long-Short Term Memory (LSTM) networks or Gated Recurrent Units (GRU). The overall strategy must be trained on several sequences of stress-strain data. These data sequences must represent strain paths and the corresponding stress responses as obtained numerically or by performing physical experiments. Due to the lack of suitable information, the internal variables are always initialised with the value of zero. Hence, the utilized training data sequences must be generated with experiments that begin from undeformed material.

Avoiding exploding/vanishing gradients by evolutionary optimisation: Training recurrent networks with standard gradient-based methods such as backpropagation, encounters issues with exploding or vanishing gradients. The occurrence of this problem is due to the frequently repeated application of the chain rule needed to back-track the gradients through the entire training data series. In this work, the problem is avoided by formulating the training procedure on a gradient-free evolutionary optimisation method. Details of this strategy is presented in Chapter 8.

Remarks on alternative strategies: A methodology for stress updating can be formulated based on LSTM or GRUs networks [46, 18]. The interconnection of these strategies are well established and include hidden states that serve the same purpose as the internal variables ξ_i in Equation (7.1.1) to reflect the material memory. Because of the integrated-in mechanisms of *forgetting* these approaches do not suffer from exploding/vanishing gradients. The gradient-free optimisation step is followed by a gradient-based training procedure to further improve results.

Despite their abilities, LSTMs and GRUs are not suitable in this framework for the following reasons:

- Networks based on LSTM or GRU add a significant amount of complexity. The numerical experimentation described in Section 9.2.2 demonstrates that this additional complexity is not needed. As the obtained results from the proposed strategy are more accurate based on less hidden states and network parameters.

- It is argued in Chapter 6 that the basic network designs employed in this study have qualities that are particularly advantageous for constitutive modelling.

- There are no memory loss mechanisms in the proposed strategy, which are essential in LSTM or GRU based architectures. This is advantageous in the context of constitutive modelling, where memory loss would be physically inaccurate.

- The proposed strategy is intended for use in the FEM for solid mechanics, where it substitutes the standard constitutive model-based stress update procedure at the Gauß point level and is normally repeated several thousand times in each load step. Therefore, the compact and basic architecture of the proposed stress update model enables efficient computation. In addition, existing implementations of LSTM or GRU networks, offered by deep learning platforms such as PyTorch or TensorFlow, are most efficiently utilised when implemented in the scripting language Python. Their efficient integration in programming language based finite element code is challenging.

- The only arguable disadvantage of the proposed strategy in comparison with LSTM or GRU based networks is the difficulty of gradient-free training. However, gradient-free network training based on evolutionary optimisation approaches has been applied in a number of publications, for instance in [7, 52, 25, 27, 6, 99]. The used strategy in this work is explained in Section 8.

7.4 Thermodynamic Consistency

The mathematical term of the second principle of thermodynamics expressed in terms of the local Clausius–Duhem inequality which also known as dissipation inequality is given in Section 2.2. The thermodynamic consistency of an internal variable based constitutive model is satisfied if the mechanical dissipation D is non-negative and can be mathematically expressed as

$$D = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{(i)} + \; q\,\dot{\xi} \;\; \geq \;\; 0, \tag{7.4.1}$$

with

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{(e)} + \boldsymbol{\varepsilon}^{(i)}, \tag{7.4.2}$$

where $\boldsymbol{\varepsilon}^{(e)}$ and $\boldsymbol{\varepsilon}^{(i)}$ represent the elastic and inelastic components of the strain tensor $\boldsymbol{\varepsilon}$, respectively. While q is the internal force associated with the internal variable ξ.

The inelastic strains $\boldsymbol{\varepsilon}^{(i)}$ and the force q are not derived in the constitutive model proposed in Equations (7.1.2) - (7.1.4). Hence, thermodynamic consistency of the proposed model is tested by making sure the dissipation inequality in Equation (7.4.1) is satisfied if the work done through any closed path in the deformation or strain space is non-negative, *i.e.*

$$D^{\circ} \;=\; \oint \boldsymbol{\sigma} : \mathrm{d}\boldsymbol{\varepsilon} \;\geq\; 0 \,. \tag{7.4.3}$$

There is no mathematical derivation from the Clausius-Duhem form in Equation (7.4.1) to the proposed dissipation inequality in Equation (7.4.3). Due to the lack of physical interpretation of the network internal variables, the Clausius-Duhem inequality can not be evaluated. The algorithmic implementation of the proposed model as given by Equations (7.1.2) - (7.1.4) is based on considering an incremental total strain step from $\boldsymbol{\varepsilon}_n$ to $\boldsymbol{\varepsilon}_{n+1}$. Hence, a closed path in the strain space is constructed as following:

1 Advancing step from $\boldsymbol{\varepsilon}_n$ to $\boldsymbol{\varepsilon}_{n+1}$.
2 Backward step to $\boldsymbol{\varepsilon}_n$.

The two-step path is schematically presented in Figure 7.4.1. The work done though this path is approximated with a second order expression as

$$D^{\circ} \;=\; \frac{1}{2}(\boldsymbol{\sigma}_n + \boldsymbol{\sigma}_{n+1}) : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n) \;+\; \frac{1}{2}(\boldsymbol{\sigma}_{n+1} + \boldsymbol{\sigma}^*_n) : (\boldsymbol{\varepsilon}_n - \boldsymbol{\varepsilon}_{n+1}) \,, \tag{7.4.4}$$

where $\boldsymbol{\sigma}^*_n$ is the stress response of the model after returning to the origin of the closed path at $\boldsymbol{\varepsilon}_n$. Equations (7.4.3) and (7.4.4) reduce to:

$$D^{\circ} \;=\; \frac{1}{2}(\boldsymbol{\sigma}_n - \boldsymbol{\sigma}^*_n) : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n) \;\geq\; 0 \,. \tag{7.4.5}$$

The proposed inequality is weaker than the expression given in Equation 7.4.1 since it is based on two steps. Yet, it can be easily implemented and requires only the evaluation of the model response to an additional fictitious step in the strain space. The relation between the expressions D and D° based on the example of uniaxial elastoplasticity is demonstrated in Figure 7.4.2.

In the remainder of this work, Equation (7.4.5) is used to test the thermodynamic consistency of the trained constitutive models. Moreover, the integration of the criterion into the training process may help to discard inconsistent network parameters at an early stage and thereby accelerate the training process. This is beyond the scope of this work and will be the subject of future investigation.

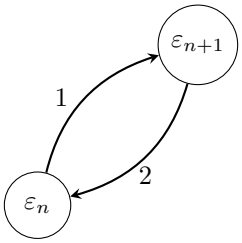


Figure 7.4.1: A two steps closed path in the strain space.

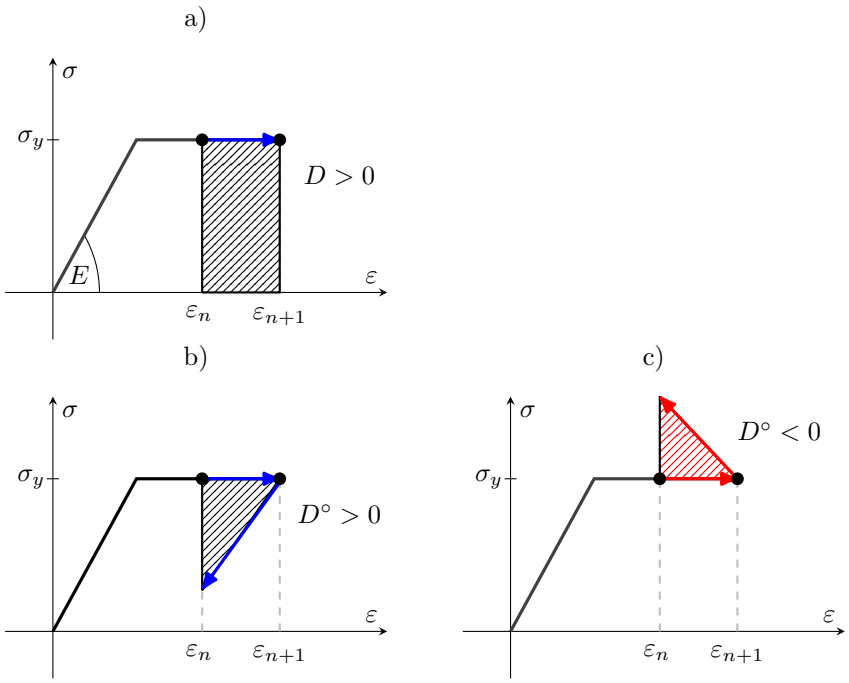


Figure 7.4.2: Illustration of a fully plastic step in perfect uniaxial elastoplasticity with $\varepsilon_{n+1}^{(i)} - \varepsilon_n^{(i)} = \varepsilon_{n+1} - \varepsilon_n$: (a) dissipation $D = \sigma_y\,(\varepsilon_{n+1}^{(i)} - \varepsilon_n^{(i)})$ from Equation (7.4.1), (b) dissipation $D^\circ = \frac{E}{2}\,(\varepsilon_{n+1}^{(i)} - \varepsilon_n^{(i)})^2$ from Equation (7.4.5), and (c) dissipation D° violating Equation (7.4.5).

Chapter 8

Network Training

The network weights and biases are obtained during the training phase to accurately fit the training data sequences. The chosen training method is a gradient-free training strategy based on evolutionary optimisation. This work employs a Multi-Objective Particle Swarm optimisation (MOPSO) approach that is integrated within a wrapper function to manage and control multiple parallel and sequential executions of the MOPSO algorithm. The used cost functions in this study are $J_s^{(k)}(x)$, where $x = \{w_{ij}, b_i\}$ denotes the weights and biases of the state and response networks, respectively. The cost function can be expressed as

$$J_s^{(k)}(x) \;=\; \sqrt{\sum_{n=1}^{N_s} \left(\sigma_n^{(k)}(x) \;-\; \widehat{\sigma}_n^{(k)}\right)^2}. \tag{8.0.1}$$

where the subscript s represent the training data series, while the superscript k identifies the stress coefficient. In addition, N_s denotes the number of load steps in the data series, while the terms $\sigma_n^{(k)}(x)$ and $\widehat{\sigma}_n^{(k)}$ represent the network based stress response to the strain data sequence and the given stress data that must be fitted, respectively.

8.1 Supervised Parallel and Sequential Evolutionary Optimisation

The optimisation challenges caused by the proposed training methodology include:

- The large number of design variables, as there are numerous weights and biases that must be obtained.
- The large number of objectives, as multiple stress-strain data series and spatial coefficients must be approximated.
- The problem is highly nonlinear, making it prone to the presence of multiple local minima, which can result in premature convergence of numerical optimisation techniques and complicate the detection of the global minimum.

The performance of the proposed strategy demonstrated that as the evolutionary optimisation explore the search space randomly, it faced the presence of large numbers of local minina as indicated from the approximation error, *i.e.* cost function given in Equation (8.0.1). Therefore, the particle swarm strategies or genetic algorithms employed converge to several different configurations associated with significantly different levels of approximation errors.

In order to efficiently execute a large number of optimisation procedures and detect early-stage premature convergence, the following strategy for supervised parallel and sequential evolutionary optimisation have been developed and implemented:

- A parallel computing platform is used with the total number of CPUs denoted as N_{CPU}. Among these processors, one processor is assigned to be the supervisor, while the remaining CPUs are divided into N_{team} teams.
- Each team follows a collaborative learning process to accomplish the particle swarm optimisation procedure as described in Section 8.2. A schematic representation of CPUs different roles is displayed in Figure 8.1.1.
- The supervisor is responsible for communicating with the team leaders and thereby frequently initiates, monitors and terminates the optimisation procedures.
- Each new instance of the particle swarm method is initialised with the best solutions obtained at that point in the optimisation process.

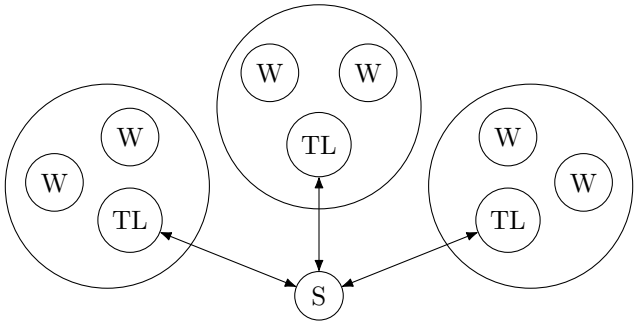


Figure 8.1.1: The different roles of the CPUs in the proposed supervised optimisation strategy: supervisor (S), team leaders (TL) and workers (W). The example shows $N_{\mathrm{CPU}} = 10$ CPUs and $N_{\mathrm{team}} = 3$ teams.

The mechanism of supervisor judgment to terminate a specific instance of the particle swarm method is a critical component of this strategy:

The team leaders report the cost and position of their best particle to the supervisor regularly (every 100 generations). If the cost received from a specific team does not show enough progress in the last N_{stall} messages, then the supervisor regards the optimisation process as *stalled*, *i.e.*

$$\frac{\epsilon_m}{\epsilon_{m-N_{\mathrm{stall}}}} > 1 - tolerance \qquad \Rightarrow \qquad \text{Optimisation has stalled.} \tag{8.1.1}$$

In Equation (8.1.1), ϵ_m represents the average of the cost values $J_s^{(k)}$ from Equation (8.0.1) received by the supervisor in the m-th message from the corresponding optimisation team. Importantly, N_{stall} is related to the current error level ϵ_m, *i.e.*

$$N_{\mathrm{stall}} = \bar{N}_{\mathrm{stall}} \left(\frac{\bar{\epsilon}}{\epsilon_m} \right)^p \tag{8.1.2}$$

where $\bar{N}_{\mathrm{stall}} = 5$ and $\bar{\epsilon}$ are reference values. Common values for the exponent $p > 0$ are 1, 2 or 3. It is advantageous to set $\bar{\epsilon}$ to the value of the error associated with the first occurrence of stall based on $N_{\mathrm{stall}} = \bar{N}_{\mathrm{stall}}$. Once a certain optimisation process has stalled, the supervisor assess the error by comparing the value of ϵ_m to the recent error values reported by the other optimisation processes. If the stalled optimisation process is one of the $N_{\mathrm{keep}} < N_{\mathrm{team}}$ better performing processes it is permitted to continue running. However, if it is not among those better performing processes, the supervisor will request that it be terminated, and a new optimisation process will be initiated. This process will continue until N_{run} procedures have been completed.

The approach described here has been fundamental to the success of the proposed stress update strategy. Alternative strategies for integrating several instances of optimisation procedures are detailed in [84, 87, 20, 106, 76, 108, 95].

8.2 Multi-Objective Particle Swarm Optimisation

The principal features of the multi-objective particle swarm optimisation strategy employed in this work are explained below. Extensive numerical testing with various particle swarm approaches resulted in the strategy. Although the optimisation process is not the primary subject of this study, it is discussed briefly in the following section for completeness. Furthermore, different methodologies may be equally or more appropriate for training the suggested stress update procedure. An overview of particle swarm based multi-objective optimisation techniques is provided in [83]. The key features can be summarised as following:

1) Continuous weight aggregation: If S denotes the number of data series, *i.e.* experiments, available for training and K is the number of stress coefficients ($K = 1$, 4 or 6 for uniaxial, plane strain or three dimensional case, respectively), then the number of objective functions represented by Equation (8.0.1) is $N_{\text{obj}} = S \times K$. A randomised, continuous and smooth weight aggregation is used repeatedly in varying order to shift the focus of the optimisation on single objectives and different weighted averages. While the objective of the training process is to minimise the average approximation error, testing shows that continuously changing the attention to different sections of the pareto front accelerates the overall convergence considerably.

2) Selection of best particles and parallelisation: As illustrated in Figure 8.1.1, the swarm is divided evenly across the CPUs that form the team. The team leader collects the best particles from all workers before each velocity and position update, finds the global best, and sends it back to each worker. The best particles are chosen in each generation based on the current weights. A smooth and steady alteration of the weights is maintained by continually setting random target weights and utilising approximately 100 generations to gradually move from one set of target weights to the next. The team leader monitors the target weights and communicates them to the workers. In addition, the team leader maintains a repository of best particles, i.e. the most advanced pareto front. The global best is continuously identified in the repository, which keeps the swarm focused and prevents deviation from good solutions.

3) Scaling of training data: The stress data is scaled to the interval $[-1, +1]$ before each training procedure to center the values around 0. This can be accomplished in three ways:

- Each stress coefficient series is scaled individually. This results in $N_{\text{obj}} = S \times K$ scaling factors.
- Each stress coefficient is scaled uniformly across all experiments. This results in K scaling factors.
- All stress coefficients are scaled with a single scaling factor.

The third approach is utilised to scale the training data of the numerical examples presented in this work, because the maximum and minimum values of all stress coefficients are of the same order of magnitude. For materials or applications where the extreme values of axial stresses are significantly higher than shear stresses, the second scaling technique can be more suitable.

4) Communication with supervisor: The team leaders communicate the current global best particle cost and position to the supervisor at regular intervals and receive an instruction to continue or restart the optimisation process in response.

Chapter 9

Numerical Examples

In this section, several numerical examples are considered based on the proposed strategy in Section 7. The objective of these examples is to demonstrate that the proposed methodology is generic, trustworthy and compliant with physical constraints. The methodology is applied to a number of rate-independent inelastic solid materials. Furthermore, the method is applied to multi-scale homogenisation problem in small strain for porous elastoplastic material.

9.1 Validation

The presented numerical examples in this section for linear elasticity, uniaxial perfect elastoplasticity and uniaxial elastoplasticity with hardening demonstrates that any piecewise linear behaviour can be recovered exactly by the model.

9.1.1 Example 1: Linear Elasticity

The described methodology is applied to a simple linear elasticity example. Given sets of strain and stress data, the training procedure is expected to accurately recover the compliance matrix values. As the problem is elastic, the state network is omitted, and only the response network is considered. The employed network architecture consist of three input neurons representing the strains, which are linearly connected to three output neurons representing the stresses (3→3). Hence, the total number of network weights is 9.

Training Data

The training data is generated with RVE that is subjected to 50 different prescribed macro-strain under plane strain conditions. The model is trained with input/output data sets (*i.e.* inputs: $\varepsilon_{xx}, \varepsilon_{yy}$ and ε_{xy}) and the associated macro-stress (*i.e.* output: σ_{xx}, σ_{yy} and σ_{xy}). The RVE mesh utilised consists of distributed layers of matrix and fiber materials, with a rotation angle of $\alpha = 30^o$, as shown in Figure 9.1.1. The material properties are assigned to each layer as follows

Matrix: $E_m = 20MPa$ and $\nu_m = 0.35$.

Fiber: $E_f = 200MPa$ and $\nu_f = 0.45$.

Furthermore, the same 50 prescribed strains values, matrix and fiber parameters are used to approximate the corresponding stresses under plane strain with the compliance matrix values obtained by rule of mixtures in Section 4.3 and using the transformation matrix given in Equation (2.4.2).

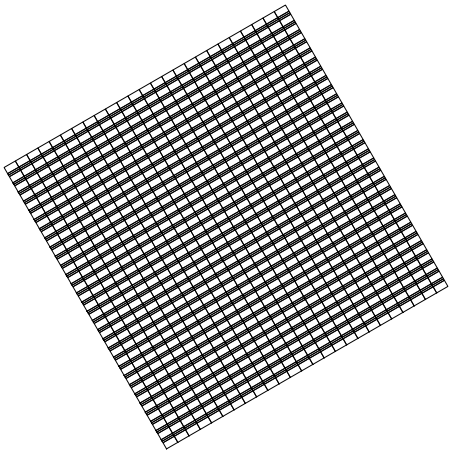


Figure 9.1.1: The used 30^o *rotated RVE mesh.*

Results

The model is trained twice with stresses approximated by: (a) rule of mixtures (RoM) and (b) the RVE. The compliance matrix values obtained by rule of mixtures (*i.e.* the target values presented in Table 4.3.2) are compared to the weights determined when training linear network with data generated by RoM and RVE.

Table 9.1.1: The compliance matrix values obtained by rule of mixtures are compared to the weights determined by the trained model.

w	c_{xx}	c_{yy}	c_{zz}	c_{xy}	c_{yz}	c_{zx}	c_{yx}	c_{zy}	c_{xz}
Target	60.7182	34.1238	19.5032	20.4874	6.2880	16.7434	20.4874	6.2880	16.7434
Training RoM	60.7182	34.1238	19.5032	20.4874	6.2880	16.7434	20.4874	6.2880	16.7434
Training RVE	61.1123	35.4844	19.8130	21.1272	5.8924	16.3513	21.2152	5.7264	16.2984

It can be observed from Table 4.3.2 that the target weights are identical to the trained network with the rule of mixture data and matrix symmetry is successfully achieved. While, the weights of model trained with the RVE data were slightly different but still demonstrated good accuracy. Increasing the number of matrix and fiber layers is recommended to further enhance accuracy.

9.1.2 Example 2: Uniaxial Perfect Elastoplasticity

The described methodology is applied to uniaxial perfect von Mises elastoplasticity. It is shown in Section 6.2 that the the standard return mapping scheme can be exactly represented by the proposed neural network based stress update strategy. Hence, it is expected that the training procedure renders a stress update model which recovers exactly the training and validation data. The state network employed has three input neurons, one hidden layer with two neurons and one output neuron (3→2→1). The response network possesses two input and one output neurons only (2→1). Hence, the networks have the same architecture as shown in Figure 6.2.1 except for the additional input of ε_n to the state network. The variables to be determined by training therefore comprise two biases and ten weights.

Training and Validation Data

The standard return mapping algorithm shown in Equations (6.2.1) to (6.2.8) is used to generate the training and validation data. The parameters used are $E = 200$ GPa, $H = 0$ and $\sigma_y = 250$ MPa, which, except for the hardening modulus, correspond to the properties of Steel A36. Figures 9.1.2 (a) and (b) show, respectively, 16 cyclic strain data series and the corresponding stress-strain diagrams as obtained from the standard return mapping algorithm. Each data series consists of 200 data points. Twelve data series are used for training and four for validation.

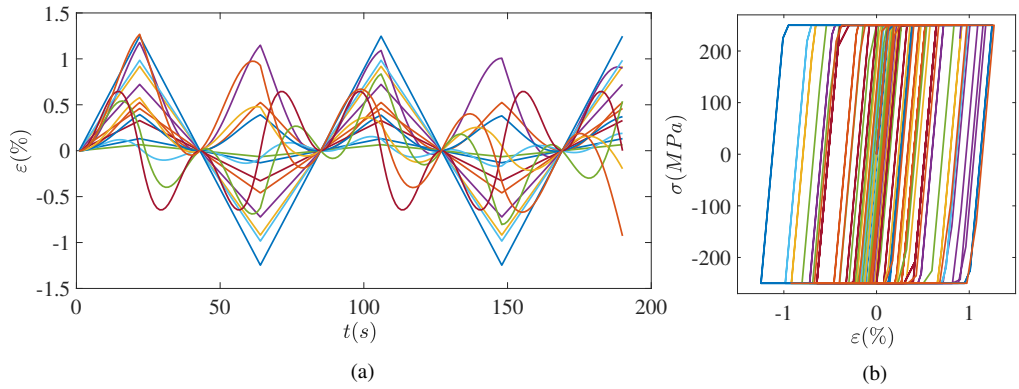


Figure 9.1.2: Different strain sequences (a) and associated stress-strain diagrams (b) for uniaxial perfect elastoplasticity.

Convergence

The determined average error convergence during training is presented in Figure 9.1.3. The estimated error decreases to the level of machine accuracy. It is worth mentioning that for demonstrational purposes, the convergence of training shown in 9.1.3 was exaggerated, hence, the training was terminated once the machine accuracy was achieved. In addition, the adopted strategy is able to achieve high accuracy predictions during seconds or a few minutes on a standard laptop.

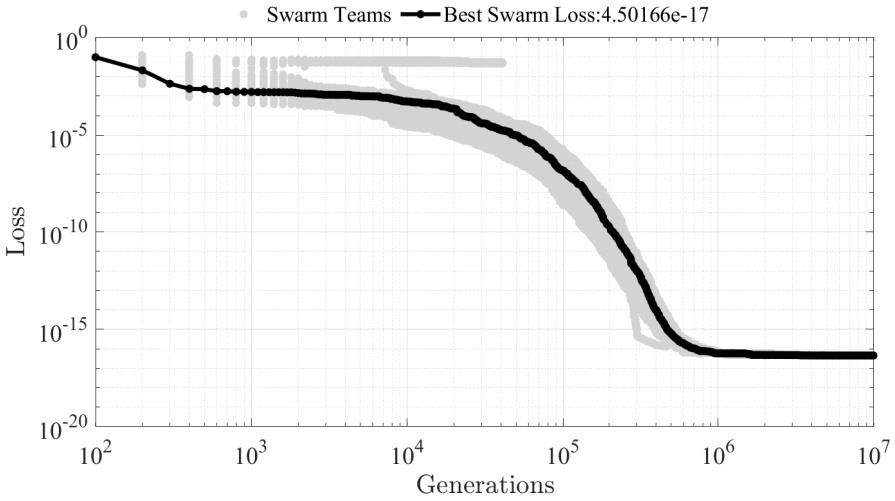


Figure 9.1.3: Training convergence for uniaxial perfect elastoplasticity.

Results

The achieved stress response results when training the network based on the stress update procedure and the standard return mapping algorithm are identical. In Figures 9.1.4, 9.1.5, 9.1.6 and 9.1.7 the obtained training results are presented in blue while the validation predictions are presented in green and the network response is presented in red. However, the associated internal state variable of the network based model is unlikely to be exactly identical to the internal plastic strain $\varepsilon^{(p)}$ because the set of weights and biases that generate the correct stress is not unique.

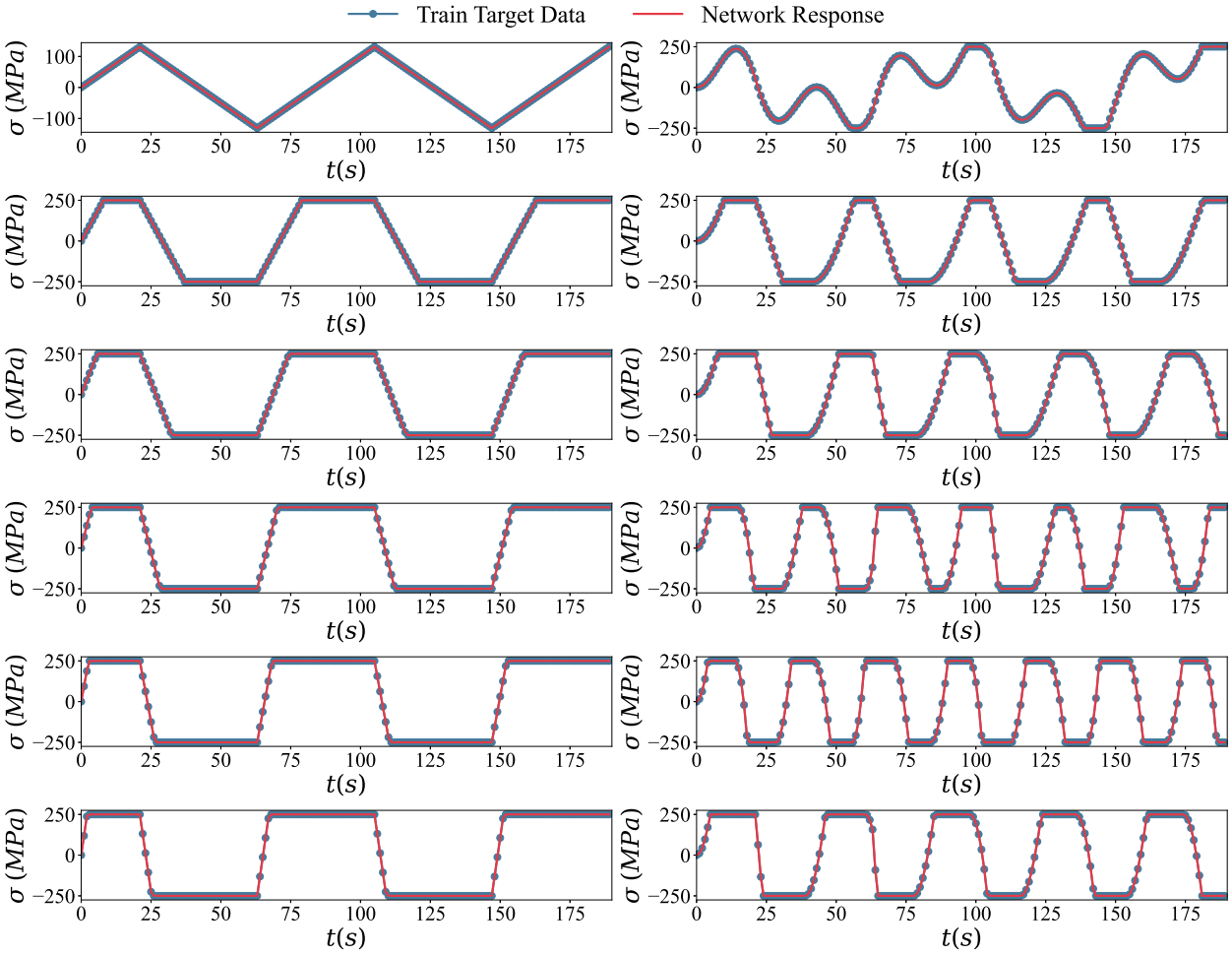


Figure 9.1.4: Stress sequences over time for the training set.

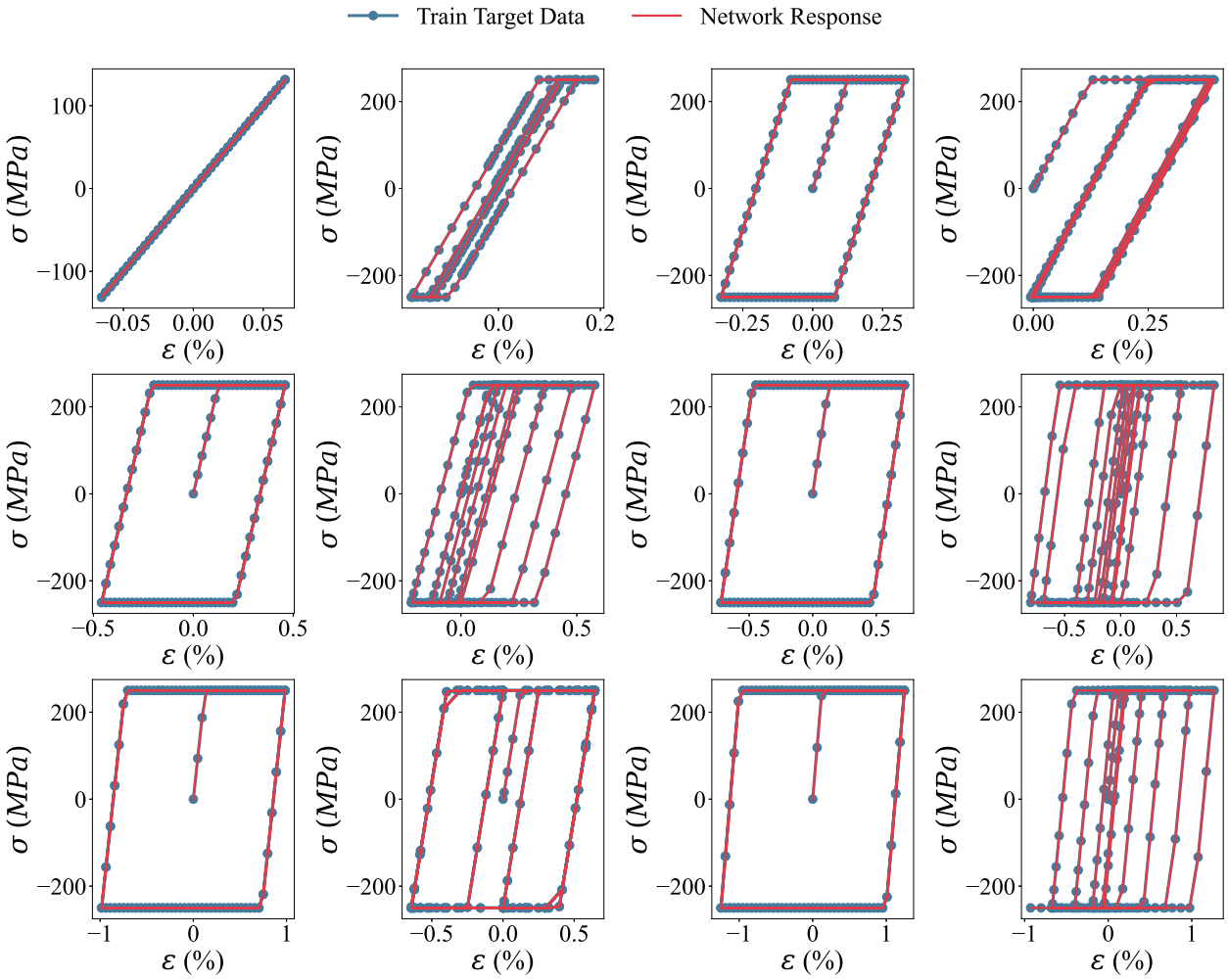


Figure 9.1.5: Stress-strain relationship for the training set.

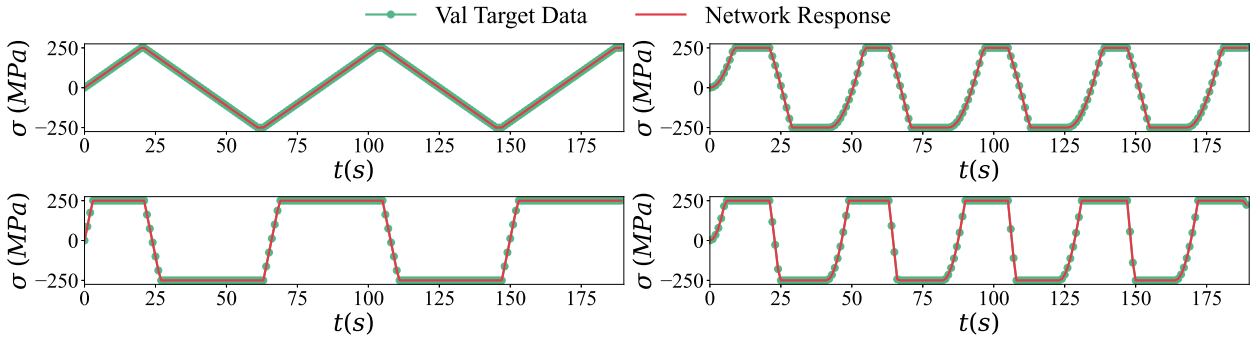


Figure 9.1.6: Stress sequences over time for the validation set.

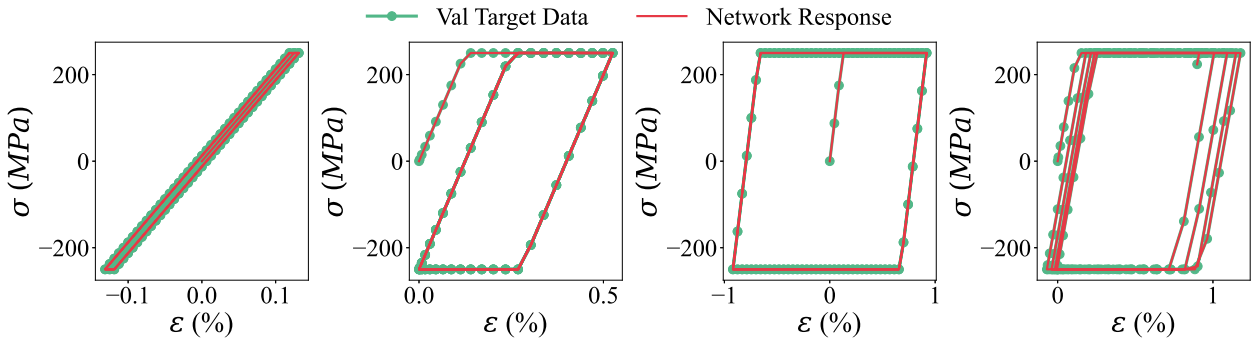


Figure 9.1.7: Stress-strain relationship for the validation set.

Thermodynamic Consistency

Thermodynamics consistency is verified after the training stage using the proposed dissipation inequality described in Section 7.4 and implemented as a bidirectional network. This means that, when a prediction at time instant $(n+1)$ is completed, a backward (unloading) prediction is also performed. The results in Figure 9.1.8 show the same sequences presented in Section 9.1.2, but now displaying the intensity and time instant when the proposed dissipation inequality is violated. As can be observed, the time points where the inequality is not fulfilled are mostly located when change of the load direction takes place. Moreover, the color map of each time step shows how large is the violation of the inequality. It should be noted that the maximum value of the dissipation violation is of the order of machine precision suggesting that it is related to the numerical approximation of the unloaded stresses by the network rather than the dissipation violation.

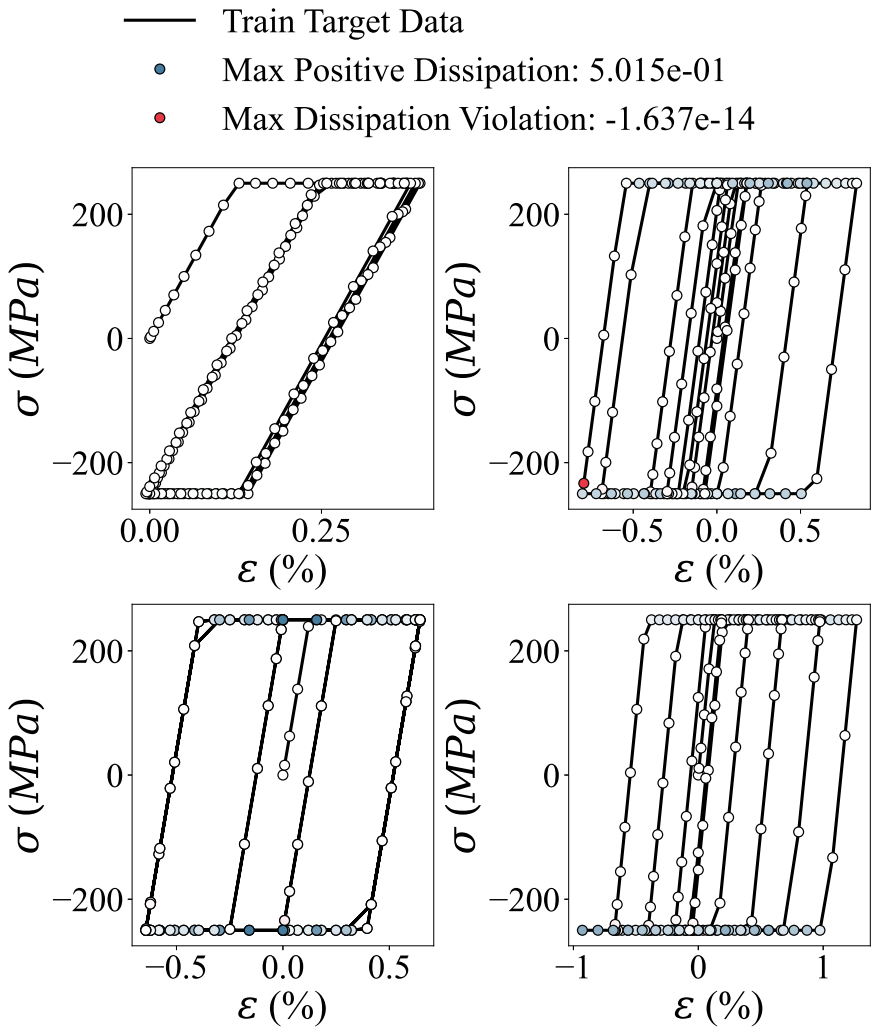


Figure 9.1.8: Stress-strain diagrams with dissipation criterion for uniaxial perfect elastoplasticity.

9.1.3 Example 3: Uniaxial Elastoplasticity with Hardening

The described methodology is applied to uniaxial von Mises elastoplasticity with isotropic linear hardening. It is shown in Section 6.2 that the standard return mapping scheme can be exactly represented by the proposed neural network based stress update strategy. Hence, it is expected that the training procedure renders a stress update model which recovers exactly the training and validation data. The state network employed has four input neurons, one hidden layer with two neurons and two output neurons (4→2→2). The response network possesses three input and one output neurons only (3→1). Hence, the networks have the same architecture as shown in Figure 6.2.3 except for the additional input of ε_n to the state network. The variables to be determined by training therefore comprise two biases and 15 weights.

Training and Validation Data

The standard return mapping algorithm shown in Equations (6.2.1) to (6.2.8) is used to generate the training and validation data. The parameters used are identical to those described in Section 9.1.2 except for $H = 10$ GPa. Figures 9.1.9 (a) and (b) show, respectively, 16 cyclic strain data series and the corresponding stress-strain diagrams as obtained from the standard return mapping algorithm. Each data series consists of 200 data points. Twelve data series are used for training and four for validation.

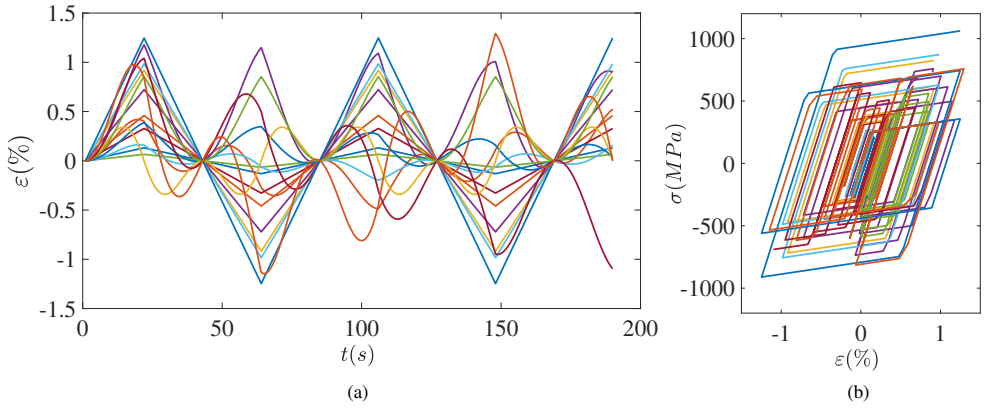


Figure 9.1.9: Different strain sequences (a) and associated stress-strain diagrams (b) for uniaxial hardening elastoplasticity.

Convergence

The convergence of the average error during training is visualised in Figure 9.1.10. Clearly the approximation error reduces to the level of machine accuracy. Similarly to Section 9.1.2 the training effort was exaggerated for demonstrational purposes.

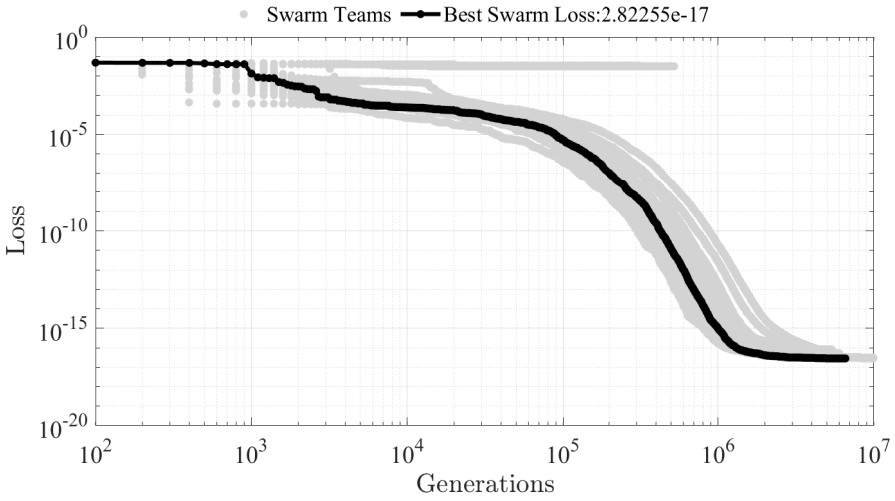


Figure 9.1.10: Training convergence for uniaxial hardening elastoplasticity.

Results

As expected the training process renders a network based stress update procedure which generates exactly the same stress responses as the standard return mapping algorithm. Figures 9.1.11 to 9.1.14 show the agreement of the model response with the training and validation data.

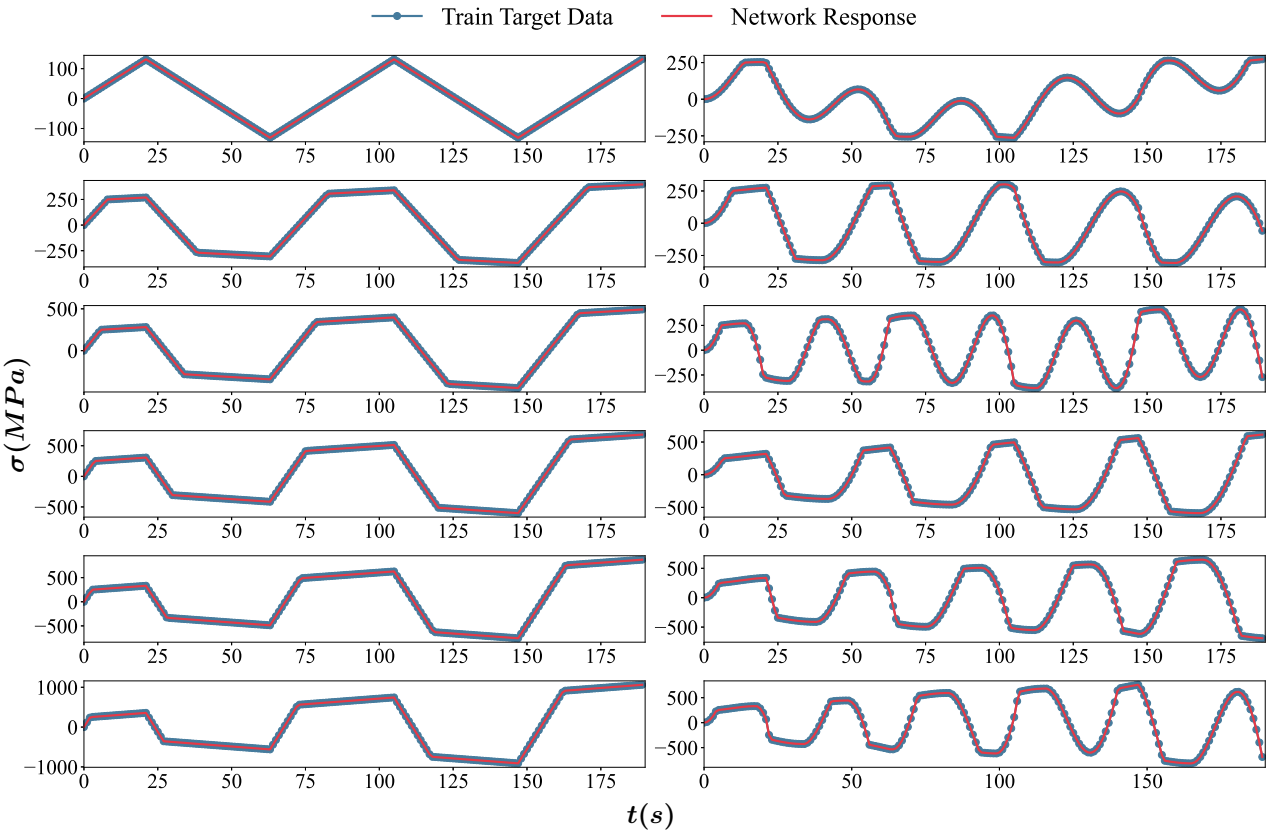


Figure 9.1.11: Stress sequences for training of uniaxial hardening elastoplasticity.

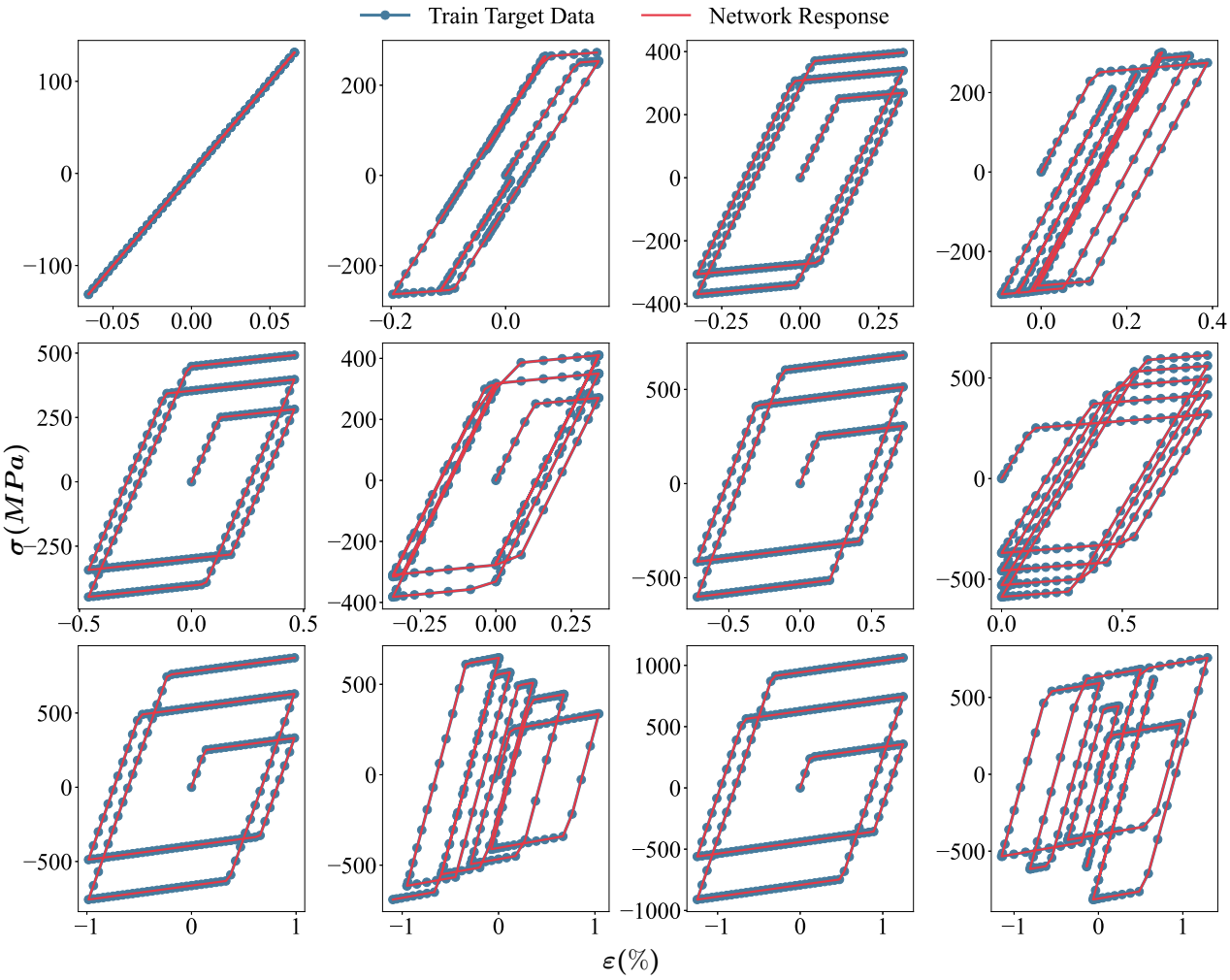


Figure 9.1.12: Stress-strain diagrams for training of uniaxial hardening elastoplasticity.

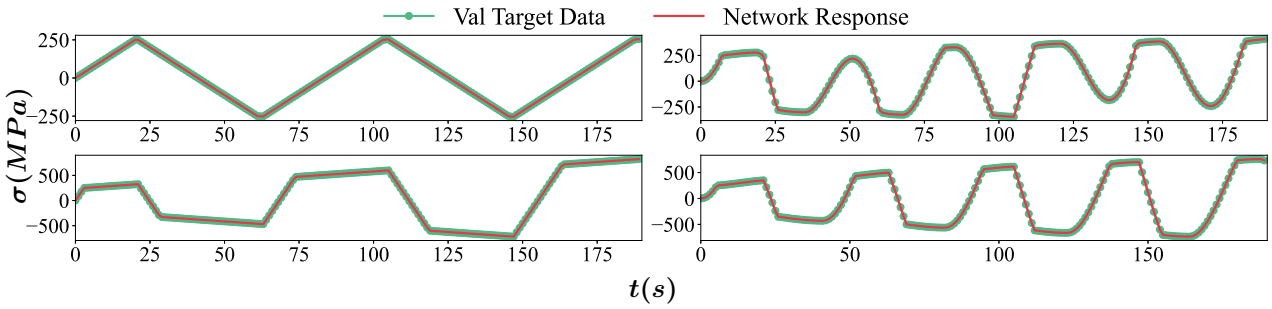


Figure 9.1.13: Stress sequences for validation of uniaxial hardening elastoplasticity.

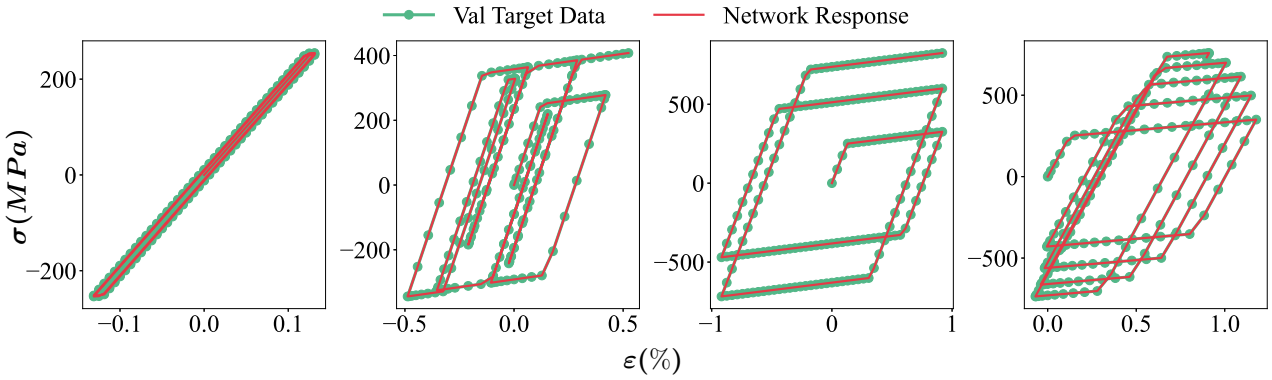


Figure 9.1.14: Stress-strain diagrams for validation of uniaxial hardening elastoplasticity.

Thermodynamic Consistency

Based on the trained model the dissipation D° as defined in Equation (7.4.5) can be evaluated at each step of any load path. Figure 9.1.15 visualises the resulting values of D° along four different strain data series. It is observed that, as expected, the largest positive values coincide with the load steps that induce purely plastic deformation, while the largest negative values occur when the direction of the loading changes, *i.e.* during the first elastic load step following a series of plastic steps. However, since the maximum violation of the dissipation criterion from Equation (7.4.5) lies well within machine accuracy, the tests do not show any instances of thermodynamic inconsistency within the trained model.

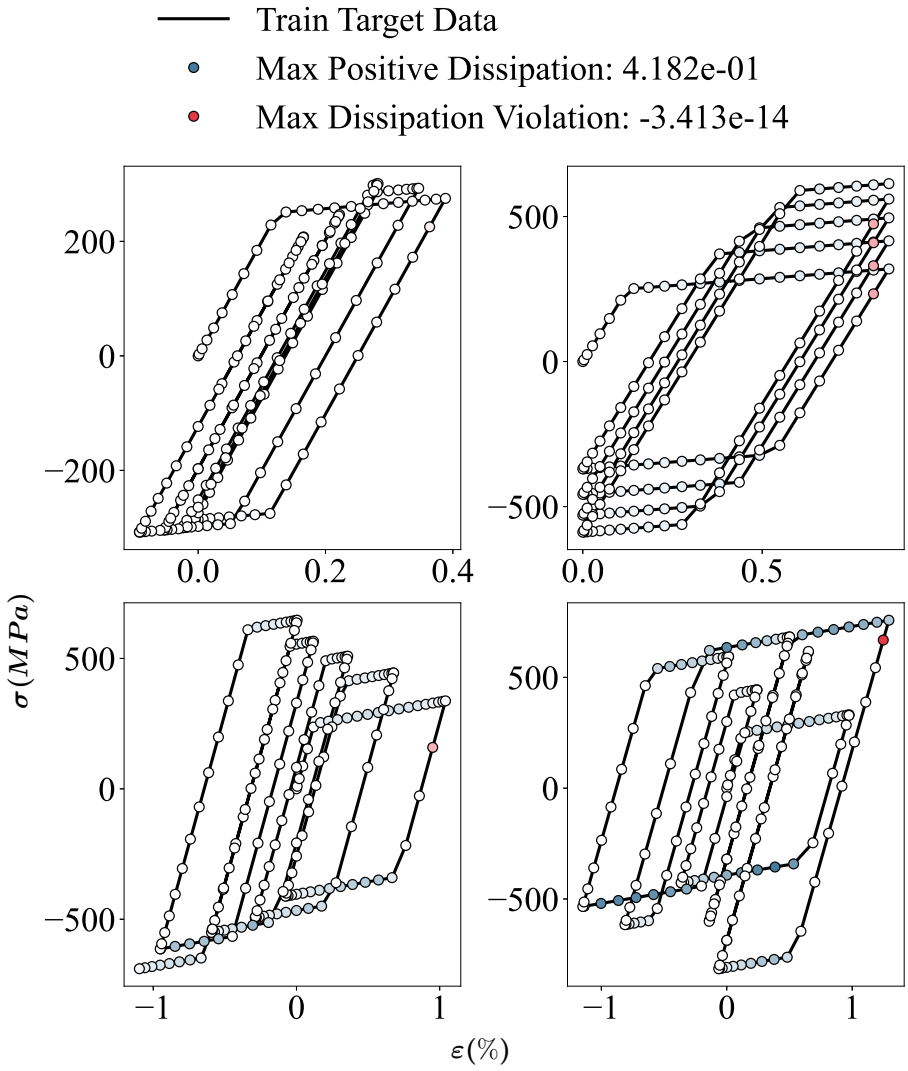


Figure 9.1.15: Stress-strain diagrams with dissipation criterion for uniaxial hardening elastoplasticity.

9.2 Manufactured Data

In this section, manufactured data is employed, generated using the associated von Mises constitutive model for uniaxial damage and plane strain elastoplasticity, to train the neural network.

9.2.1 Example 4: Uniaxial Elastoplastic Damage

The constitutive model for elastoplastic damage described in [104] is inherently nonlinear and cannot be represented exactly by the proposed network based stress update strategy. The algorithmic return mapping scheme presented in [104] is adapted to the special case of uniaxial stress and used to generate the training and validation data. The number of internal state variables used is chosen identical to the number of internal variables required by the constitutive model, *i.e.* three internal states are used. The state network employed therefore has five input neurons, one hidden layer with nine neurons and three output neurons (5→9→3), while the response network requires four input neurons and one output neuron (4→1). The variables to be determined by training comprise nine biases and 76 weights.

Training and Validation Data

The parameters are set to the same values as in [104], *i.e.* Young's modulus $E = 21,000$ kN/cm^2, isotropic hardening modulus $H = 1,500$ kN/cm^2, yield stress $\sigma_y = 70$ kN/cm^2 and the initial damage threshold $r_0 = 0.117$ kN cm. Twelve cyclic strain data series are generated and the associated stresses are computed from the algorithmic stress update presented in [104] and given in Algorithm 3. Each data series consists of 200 data points. Eight data series are used for training, the resulting four are used for validation. Figures 9.2.3 and 9.2.5 show the training and validation data in terms of stress-strain diagrams.

Convergence

The convergence of the average error during training is visualised in Figure 9.2.1. The training was performed on $N_{\text{CPU}} = 131$ CPUs of a high performance cluster with $N_{\text{team}} = 13$ and a team size of 10 CPUs and achieved the error level shown in the Figure 9.2.1.

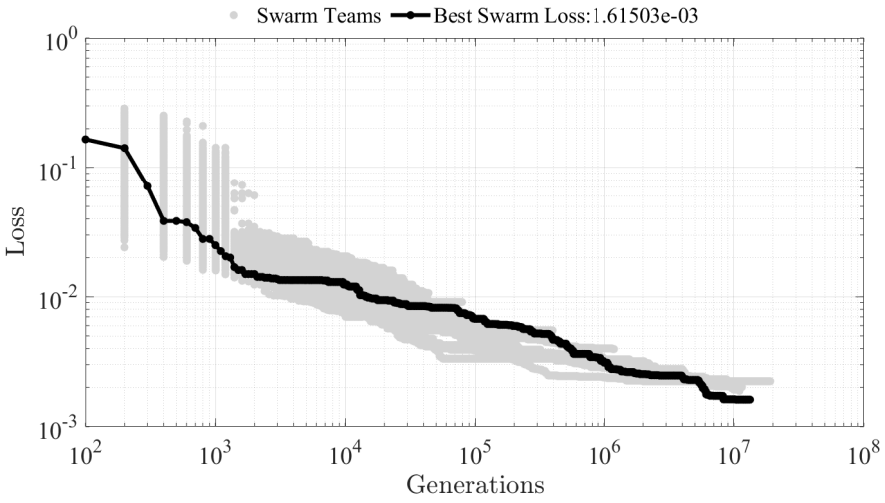


Figure 9.2.1: Training convergence for uniaxial elastoplastic damage.

Results

Figures 9.2.2 to 9.2.5 show the responses of the trained methodology together with the training and validation data. Noticable yet relatively small differences are restricted to the data series with the largest strain values. In general the agreement is observed to be accurate. Note that the Figure 9.2.1 suggests that the training has not yet fully converged and a higher degree of accuracy may be achieved by the selected network architecture.

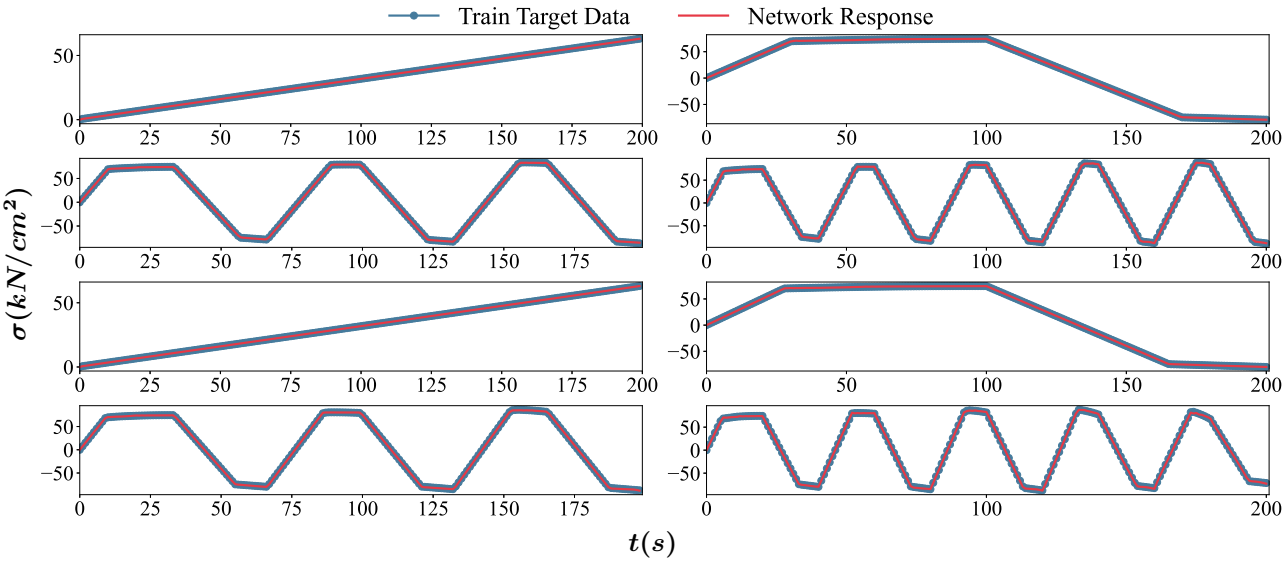


Figure 9.2.2: Stress sequences for training of uniaxial elastoplastic damage.

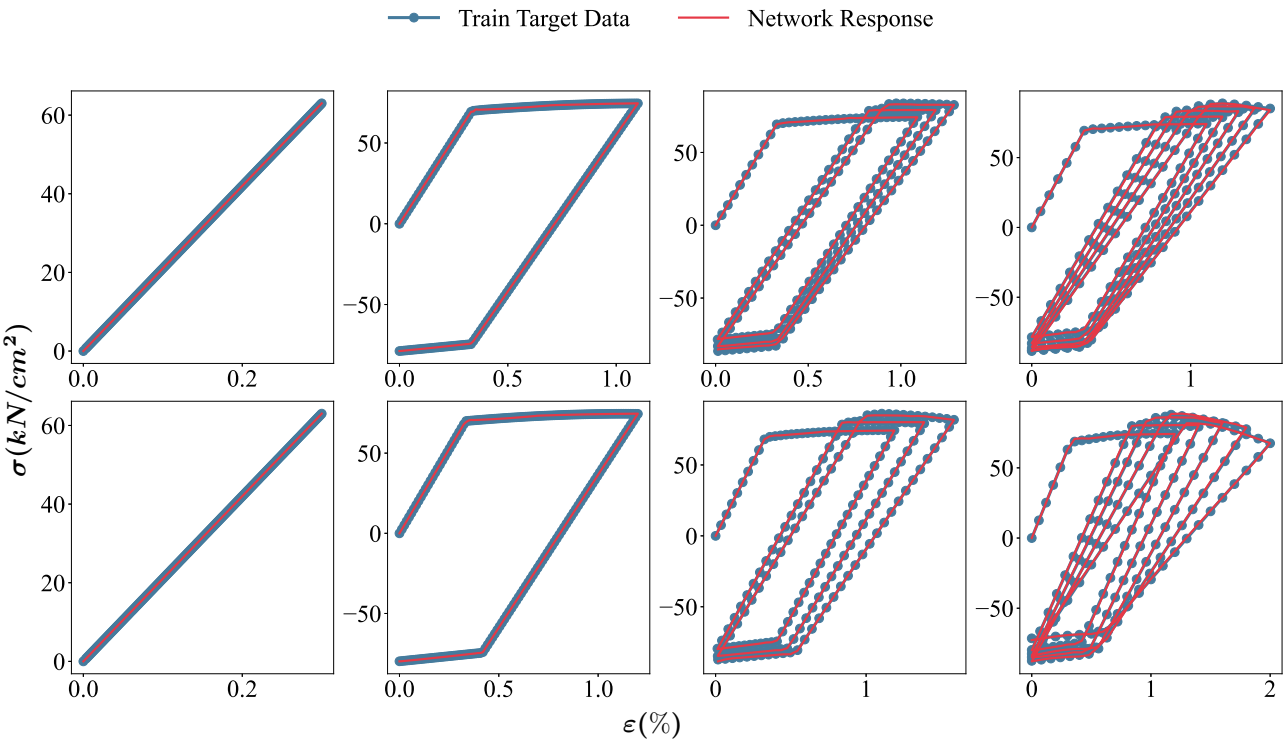


Figure 9.2.3: Stress-strain diagrams for training of uniaxial elastoplastic damage.

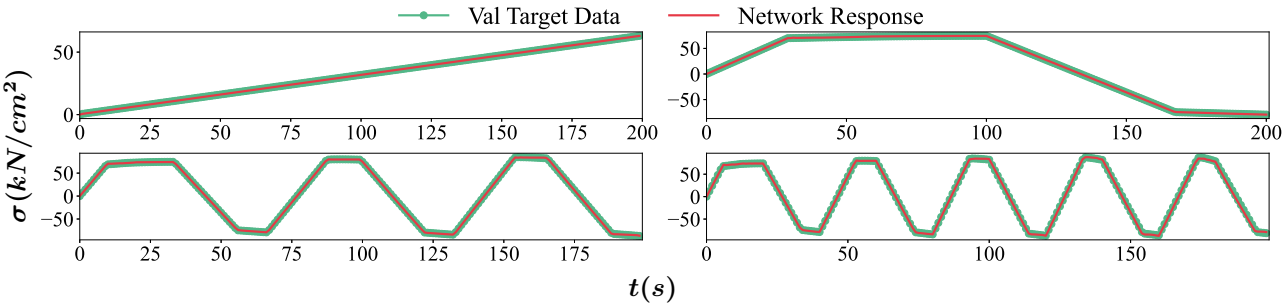


Figure 9.2.4: Stress sequences for validation of uniaxial elastoplastic damage.

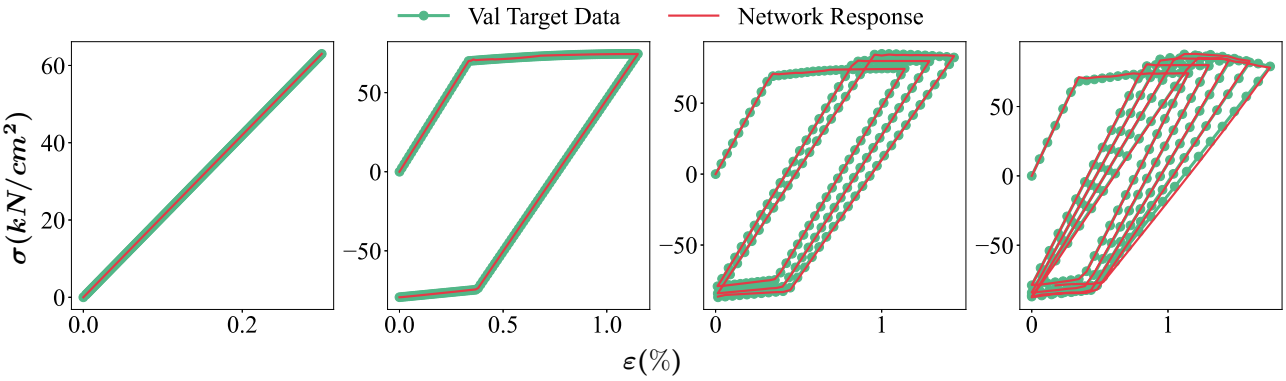


Figure 9.2.5: Stress-strain diagrams for validation of uniaxial elastoplastic damage.

Thermodynamic Consistency

The dissipation criterion proposed in Equation (7.4.5) is evaluated for four data series and visualised in Figure 9.2.6. It can be observed that violation of the criterion is restricted to the load steps immediately following changes of loading direction. The largest negative value does not exceed 9.48% of the largest value of the dissipation D° computed for any of the dissipative load steps.

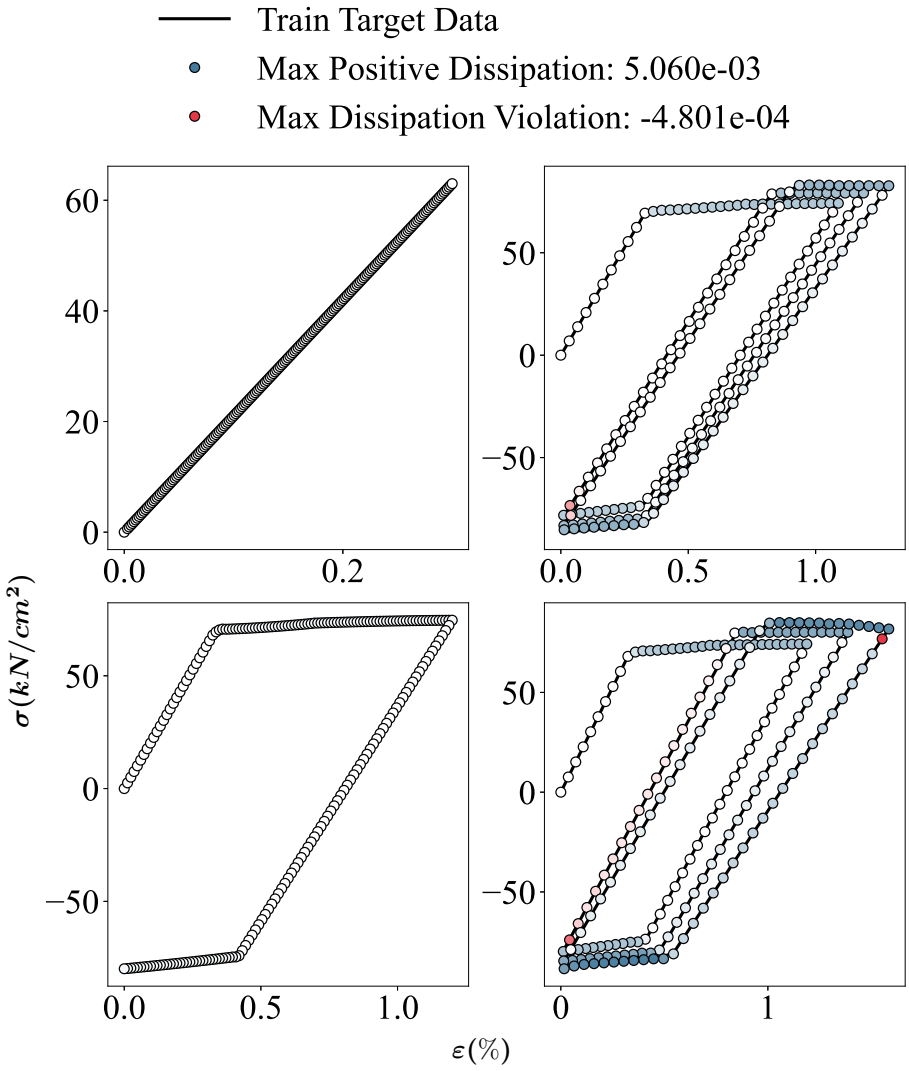


Figure 9.2.6: Stress-strain diagrams with dissipation criterion for uniaxial elastoplastic damage.

9.2.2 Example 5: Plane Strain Perfect Elastoplasticity

The return mapping algorithm for von Mises elastoplasticity is presented, for instance, in [94, 23]. For the multi-dimensional case it is inherently nonlinear due to the tensor norm operator that acts on the trial stress. In the following, the case of plane strain perfect elastoplasticity is considered, *i.e.* $H = 0$. In the plane strain state, $\varepsilon_{yz} = \varepsilon_{zx} = \varepsilon_{zz} = 0$ and $\sigma_{yz} = \sigma_{zx} = 0$. Hence, three strain coefficients $\{\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{xy}\}$ induce a response consisting of four stress coefficients $\{\sigma_{xx}, \sigma_{yy}, \sigma_{xy}, \sigma_{zz}\}$. It is chosen to provide the network based stress update model with four internal state variables, thus allowing for the representation of some form of internal plastic strains. The state network therefore requires ten input neurons and four output neurons. The hidden layer is provided with 15 neurons (10→15→4). The response network requires seven input neurons and four output neurons (7→4). Similarly to the problems presented in Sections 9.1.2 to 9.2.1 a hidden layer is not deemed necessary for the response network. It follows that 15 biases and 238 weights must be determined by the training process.

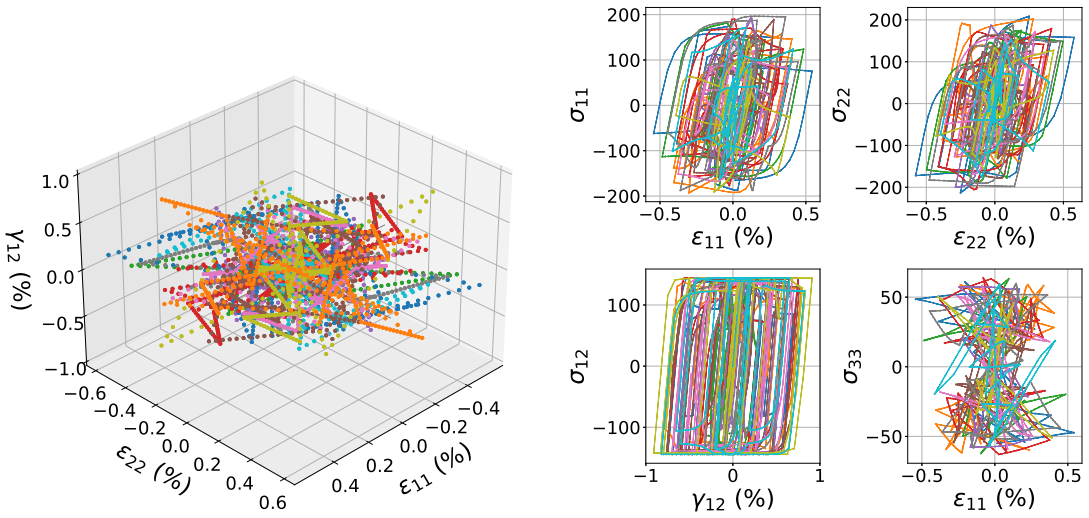
Training and Validation Data

The material parameters used for generating the training and validation data are shown in Table 9.2.1

Table 9.2.1: Material properties of steel A36.

Young's modulus E GPa	Isotropic hardening H GPa	Poisson's ratio ν -	Yield stress σ_y MPa
200	10	0.26	250

Similarly to Sections 9.1.2 to 9.2.1 the training and validation data consists of suitable strain data series and the associated stresses. The latter are computed from the strains with the return mapping scheme presented in [94, 23]. The former need to be generated with care. If the trained model is to be robust and widely applicable, they must be representative of the entire physically feasible deformation space. In this work, each strain data series is based on the linear interpolation between positions in the strain space that do not exceed 0.1% of volumetric strain $\varepsilon_V = \varepsilon_{xx} + \varepsilon_{yy}$ and do not exceed the value of 1% in any of the strain components, but are otherwise arbitrary. Suitable interpolation ensures that the load step sizes are moderate and the strain data represents a clear deformation path. Each data series consists of 750 data points. Ten data series are used for training and five four validation. The training and validation data is visualised in Figure 9.2.7 and also represented in the results in Section 9.2.2.



(a) (b)

Figure 9.2.7: Different strain sequences (a) and associated stress-strain diagrams (b) for plane strain perfect elastoplasticity.

In the problem under consideration, it is known that the data represents elastoplastic material behaviour. In such cases it is generally of particular interest to capture the yield surface accurately. Hence, it is chosen to include two additional quantities in the training, namely the hydrostatic pressure

$$p = \frac{\sigma_{xx} + \sigma_{yy} + \sigma_{zz}}{3}, \tag{9.2.1}$$

and the norm of the deviatoric stress

$$q = \sqrt{\frac{1}{2}\left((\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{xx} - \sigma_{zz})^2 + 6\left(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2\right)\right)}. \tag{9.2.2}$$

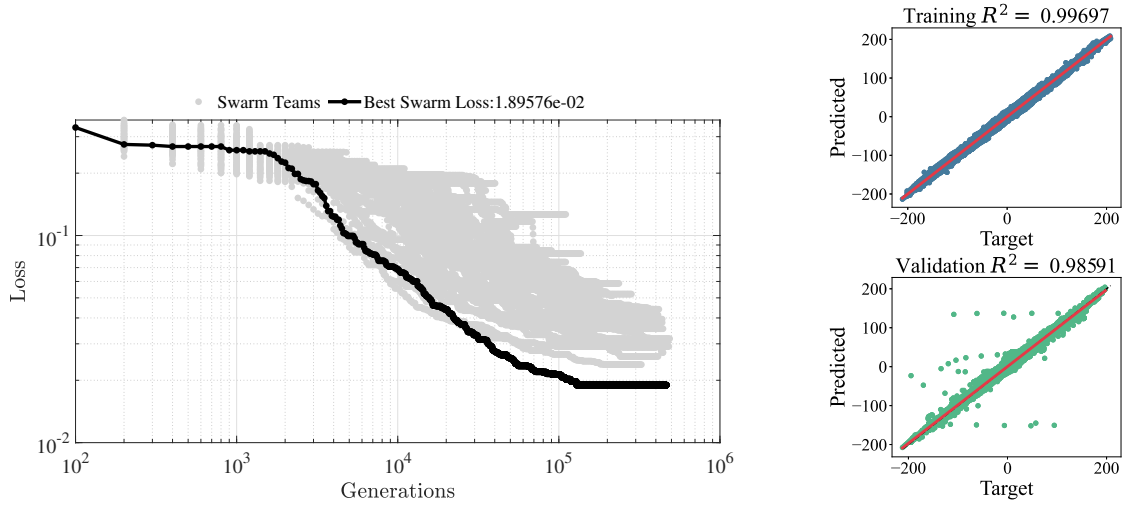
For each data series p and q are computed and corresponding approximation errors are formulated analogously to Equation (8.0.1). Hence, ten training data series, four stress coefficients together with p and q result in 60 objectives to be minimised by the multi-objective particle swarm optimisation described in Section 8.2. Recalling the supervised optimisation framework described in Section 8.1, note that it was chosen to base the messages passed from the team leaders to the supervisor on a weighted average, *i.e.* the approximation errors of p and q are weighted five times more heavily than each of the stress coefficients.

Convergence

The system is trained on $N_{\mathrm{CPU}} = 101$ CPUs of a high performance computer cluster and left to run for 72 hours. The resulting convergence is shown in Figure 9.2.8 (a). In Figure 9.2.8 (b) all stress responses are displayed against their data counterparts. For the data used in training the accuracy is high and the correlation coefficient exceeds the value of 99.6%. For the validation data the correlation coefficient exceeds 98.5%, but a small number of data points show large errors.

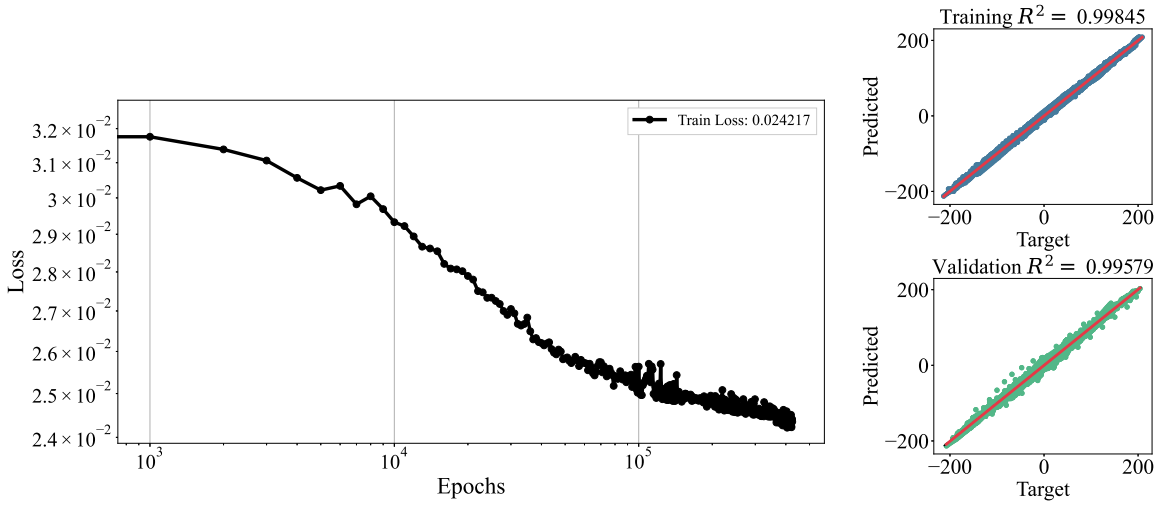
Therefore, a second training stage, *i.e.* fine tuning, based on back-propagation Through Time BPTT is performed, using the Python library PyTorch. The Adam optimiser is employed in combination with a small learning rate of 0.0001. The optimisation is initialised with the network configuration obtained from the PSO based, first stage of training. The additional objectives relating to p and q are ignored and the average approximation error of the stress coefficients is used as the single objective. The PyTorch fine tuning is carried out on a Nvidia V100 GPU for 48 hours with the intention to perform a slow training to avoid exploding gradient problems. Figure 9.2.9 (a) shows the convergence of the fine tuning process. Note that the loss measure is different from the one in Figure 9.2.8 (a) due to including or ignoring p and q. It is easily observed in Figure 9.2.9 (b) that the fine tuning effectively removed all of the poor quality responses to the validation data. For validation as well as for training data the correlation coefficients based on all stress coefficients have improved.

Neural network training, based on evolutionary optimisation followed by fine tuning with a gradient based strategy, has also been employed in, for instance [25, 99]



(a) (b)

Figure 9.2.8: Training convergence for plane strain perfect elastoplasticity: Supervised optimisation based on multiple instances of MOPSO (a) and stress responses versus stress data (b).



(a) (b)

Figure 9.2.9: Training convergence for plane strain perfect elastoplasticity: Fine tuning with PyTorch (a) and stress responses versus stress data after fine tuning (b).

Results

The network stress response and the stress data for one sequence of training data is shown in Figure 9.2.10. Figures 9.2.11 and 9.2.12 present the stress-strain diagrams for all ten training data series. The associated p-q diagrams are presented in Figure 9.2.13. Figures 9.2.14 to 9.2.17 show the corresponding diagrams for the validation data.

The responses obtained from the proposed neural network based stress update procedure are generally observed to agree accurately with the training and validation data. The most notable deviation is noticed in the norm of the deviatoric stresses, q, in Figures 9.2.13 and 9.2.17. This is due to the nonlinear dependency of q on the stress coefficients: Minimising the average of the approximation errors of the stress coefficients is unlikely to simultaneously minimise the approximation error in q. It is also suspected that the particular choices of data preprocessing and scaling prior to training has significant effects as described in Section 8.2.

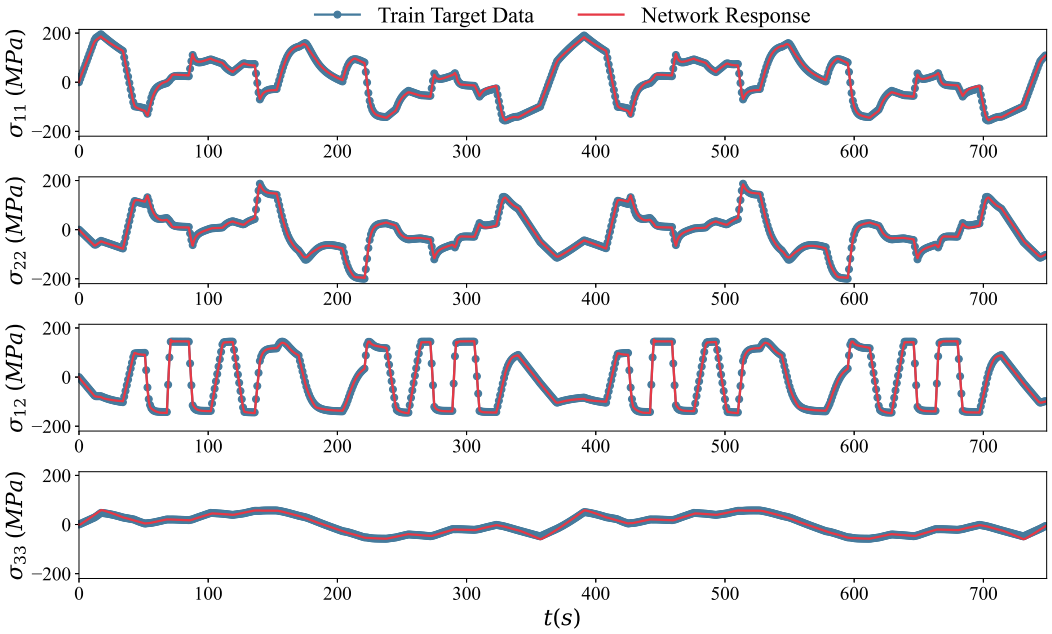


Figure 9.2.10: Stress sequences for training of plane strain perfect elastoplasticity, Data Series 4.

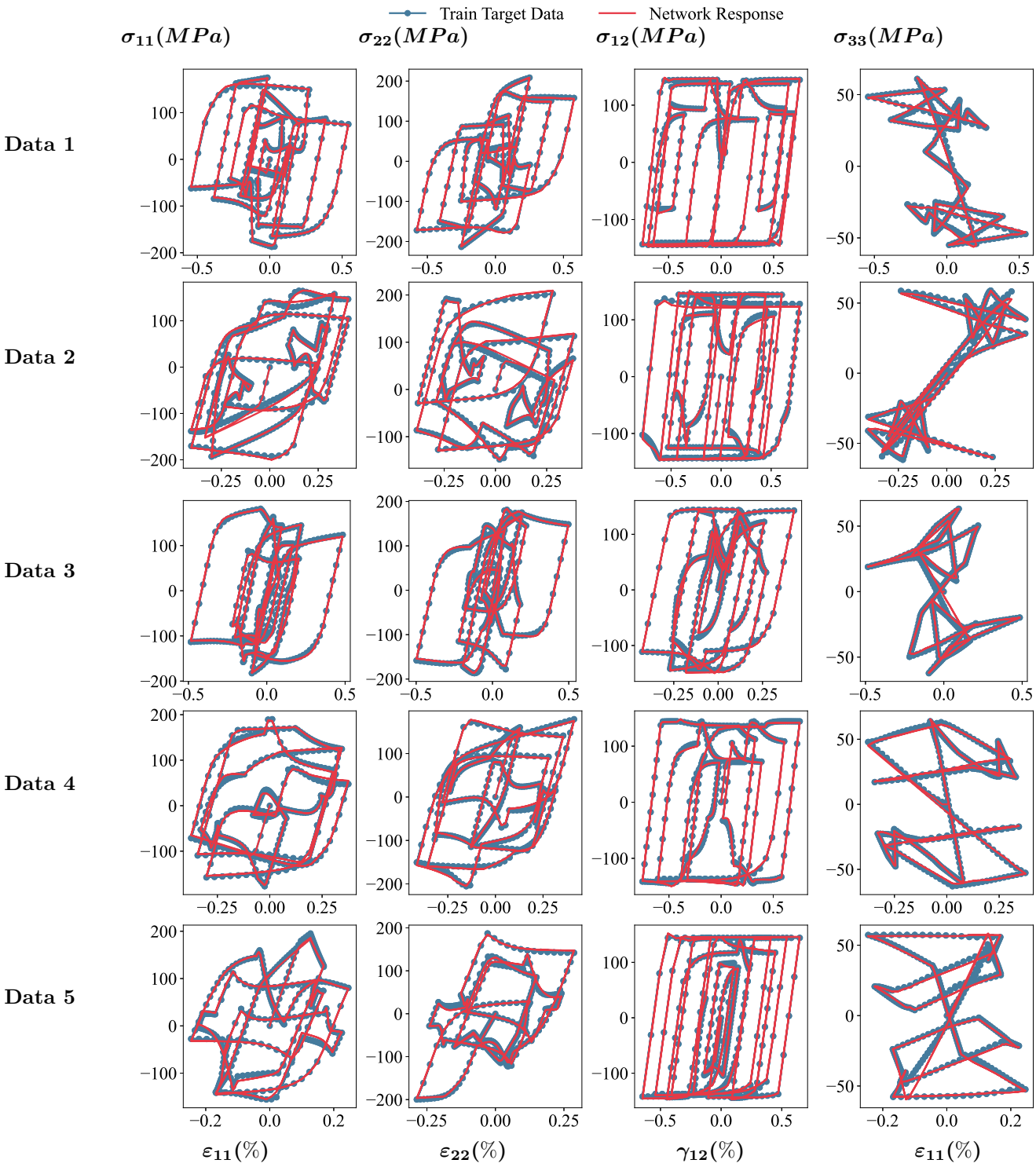


Figure 9.2.11: Stress-strain diagrams for training of plane strain perfect elastoplasticity, Data Series 1 to 5.

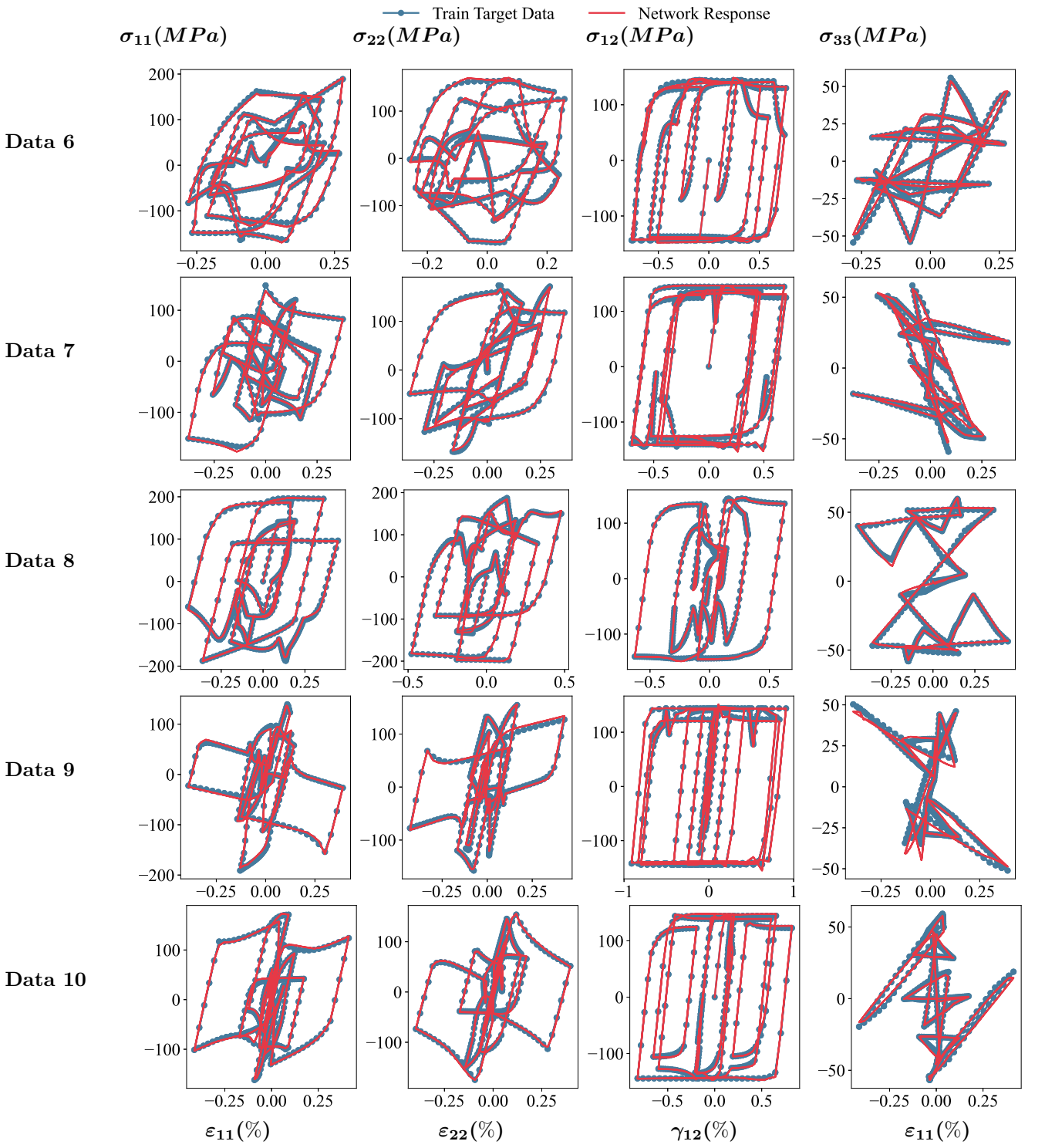


Figure 9.2.12: Stress-strain diagrams for training of plane strain perfect elastoplasticity, Data Series 6 to 10.

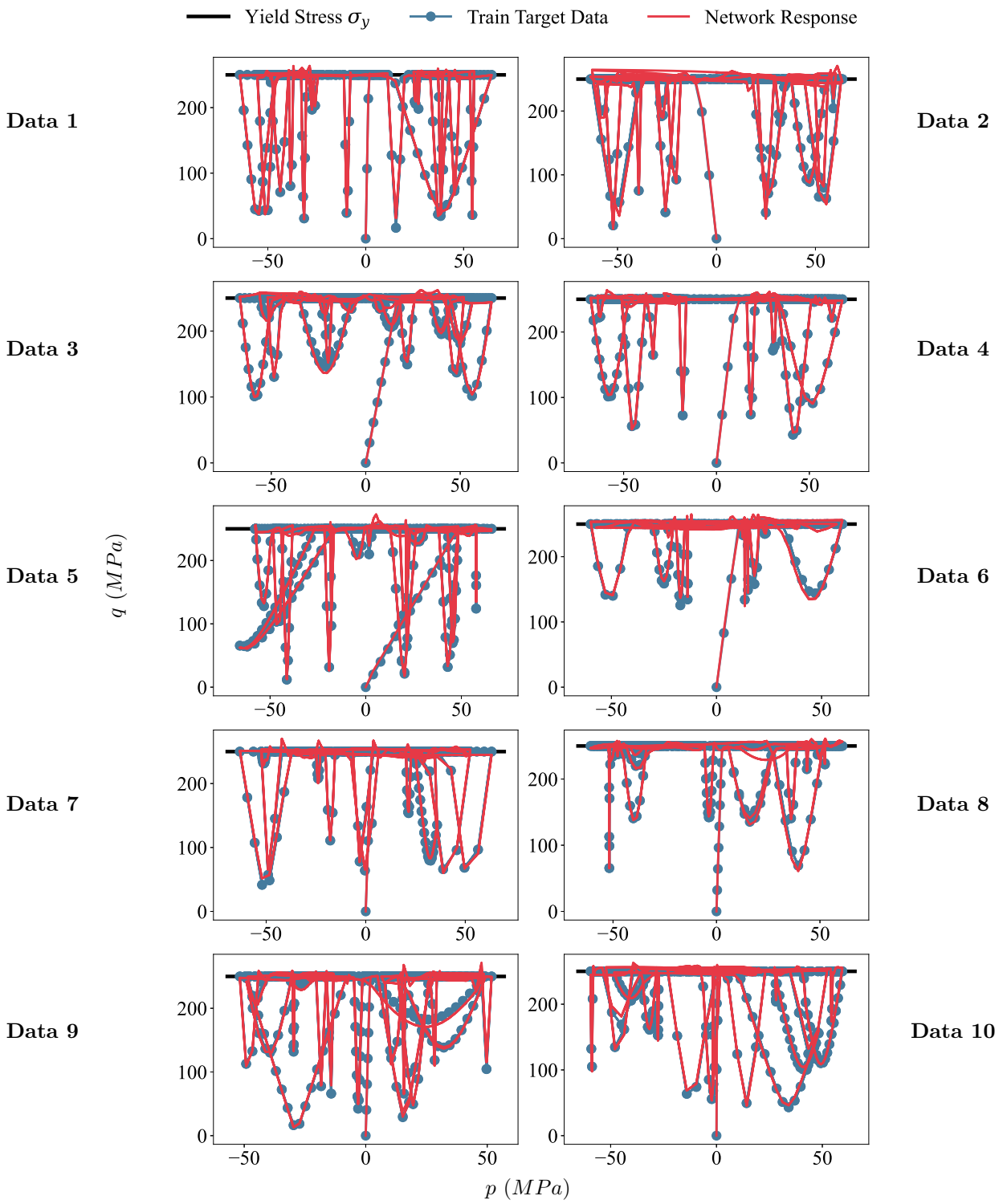


Figure 9.2.13: p-q diagrams for training of plane strain perfect elastoplasticity.

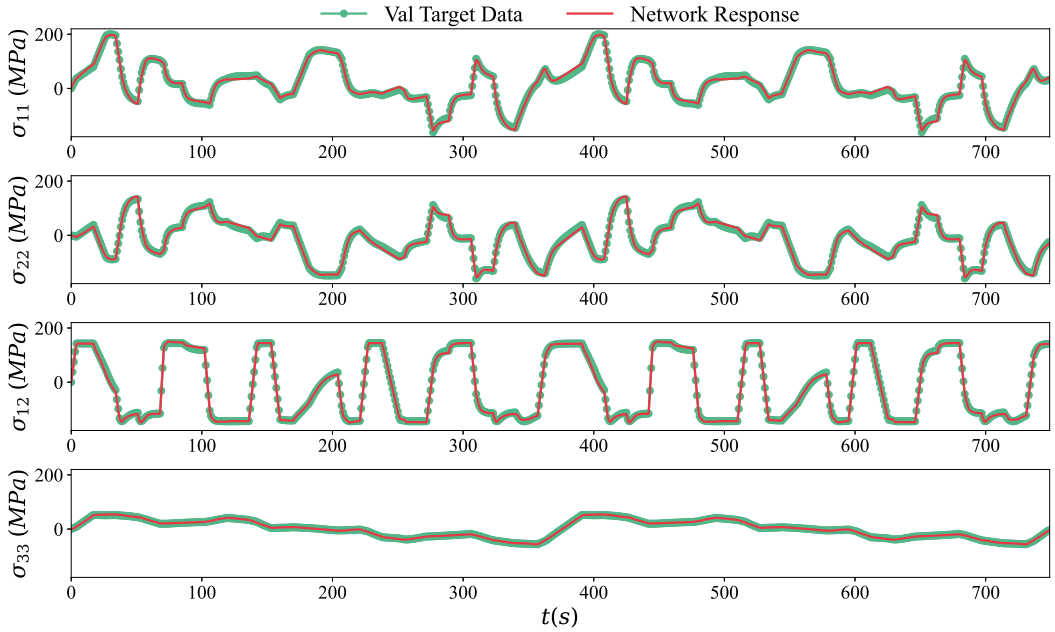


Figure 9.2.14: Stress sequences for validation of plane strain perfect elastoplasticity, Data Series 2.

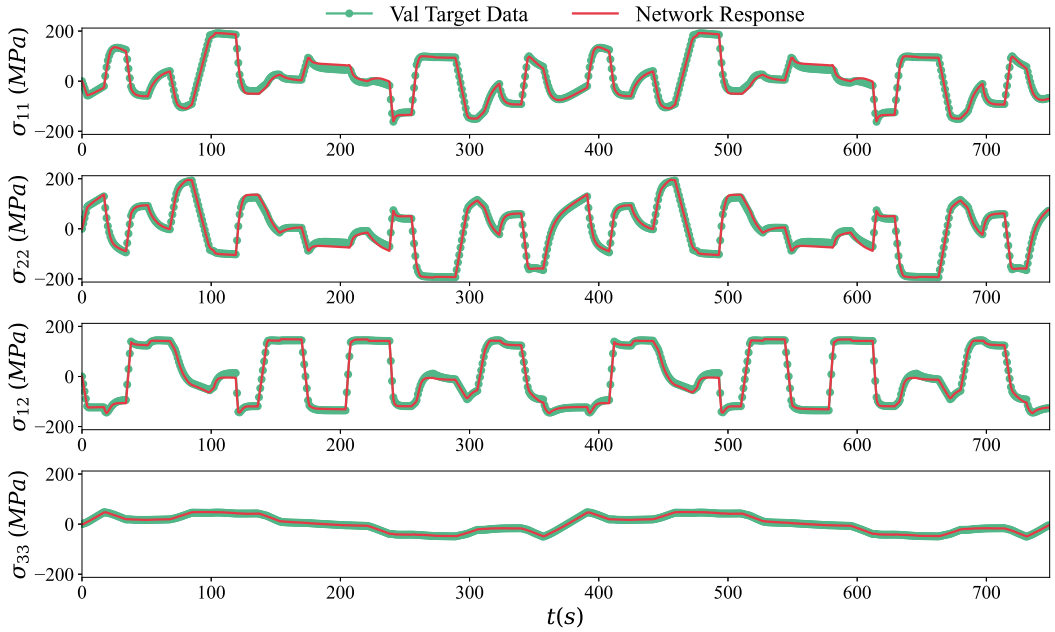


Figure 9.2.15: Stress sequences for validation of plane strain perfect elastoplasticity, Data Series 5.

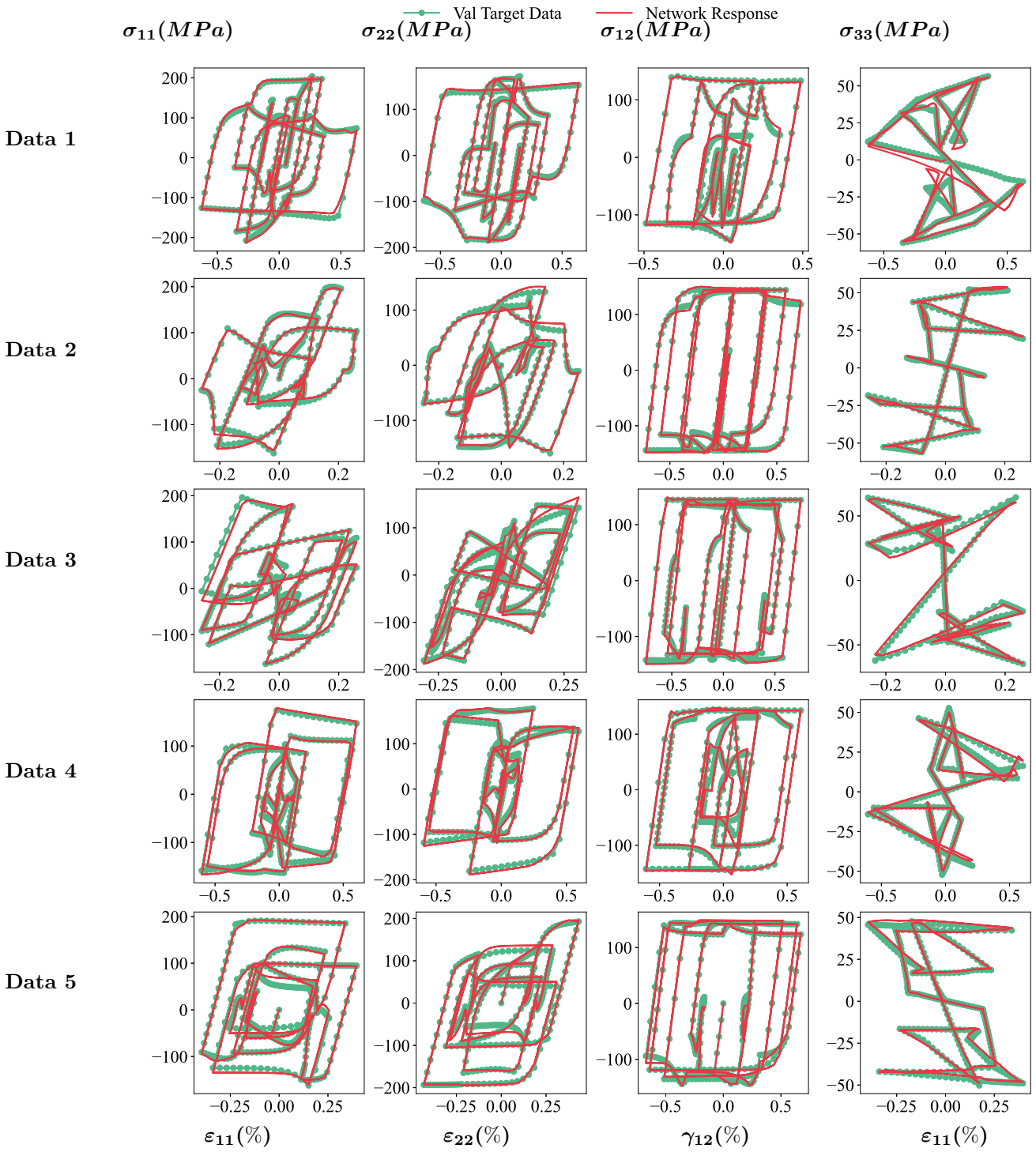


Figure 9.2.16: Stress-strain diagrams for validation of plane strain perfect elastoplasticity.

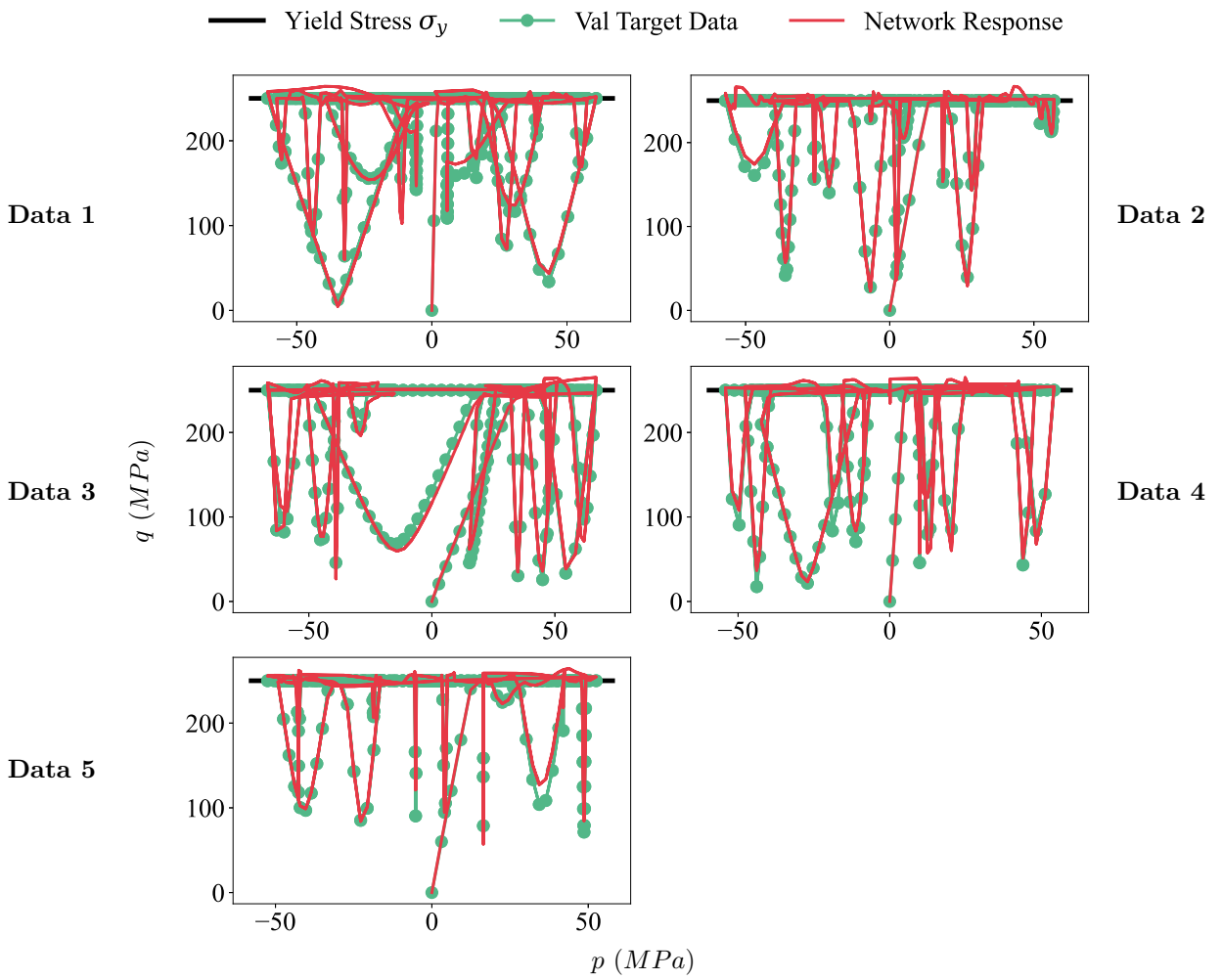


Figure 9.2.17: p-q diagrams for validation of plane strain perfect elastoplasticity.

Thermodynamic Consistency

The dissipation criterion proposed in Equation (7.4.5) is evaluated and visualised for two data series in Figures 9.2.18 and 9.2.19. It can be observed that violation of the criterion is restricted to the load steps immediately following changes of loading direction. The largest negative value does not exceed 6% of the largest value of the dissipation D° computed for any of the dissipative load steps.

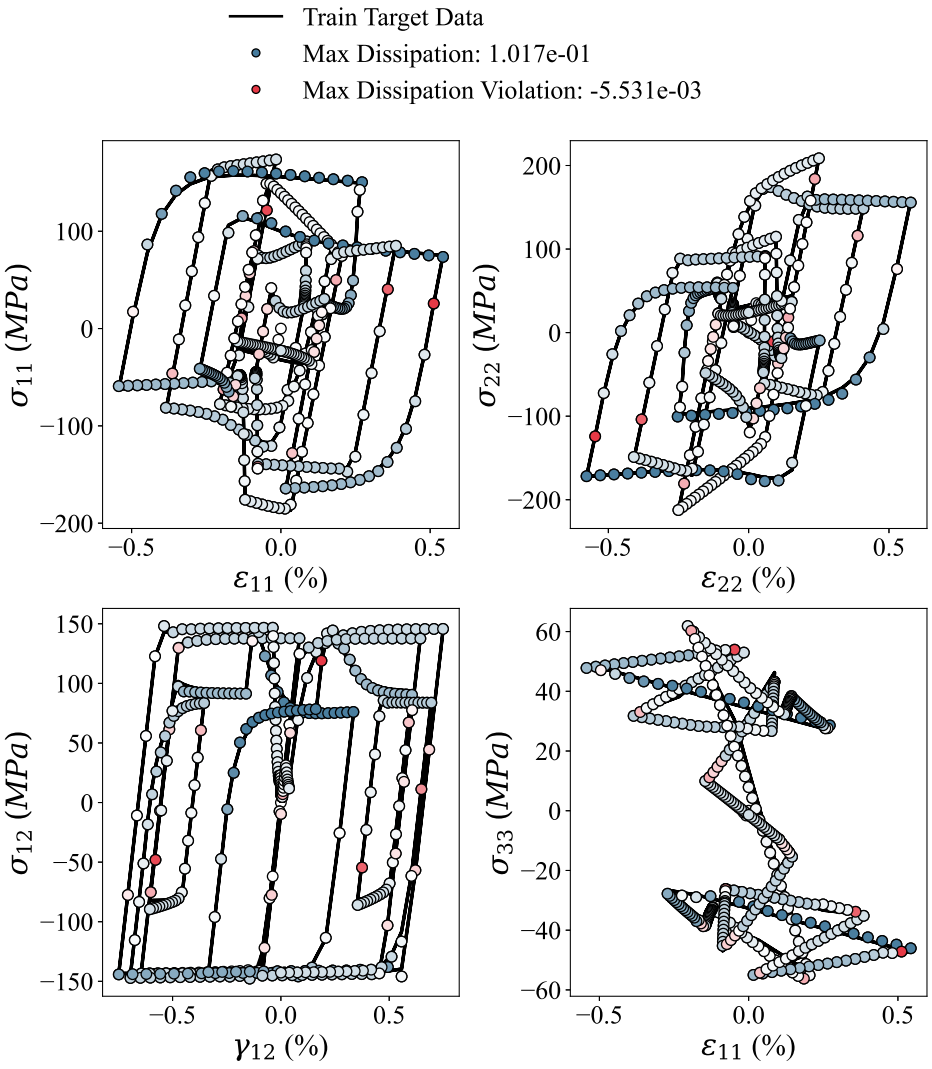


Figure 9.2.18: Stress-strain diagrams with dissipation criterion for plane strain perfect elastoplasticity, Data Series 1.

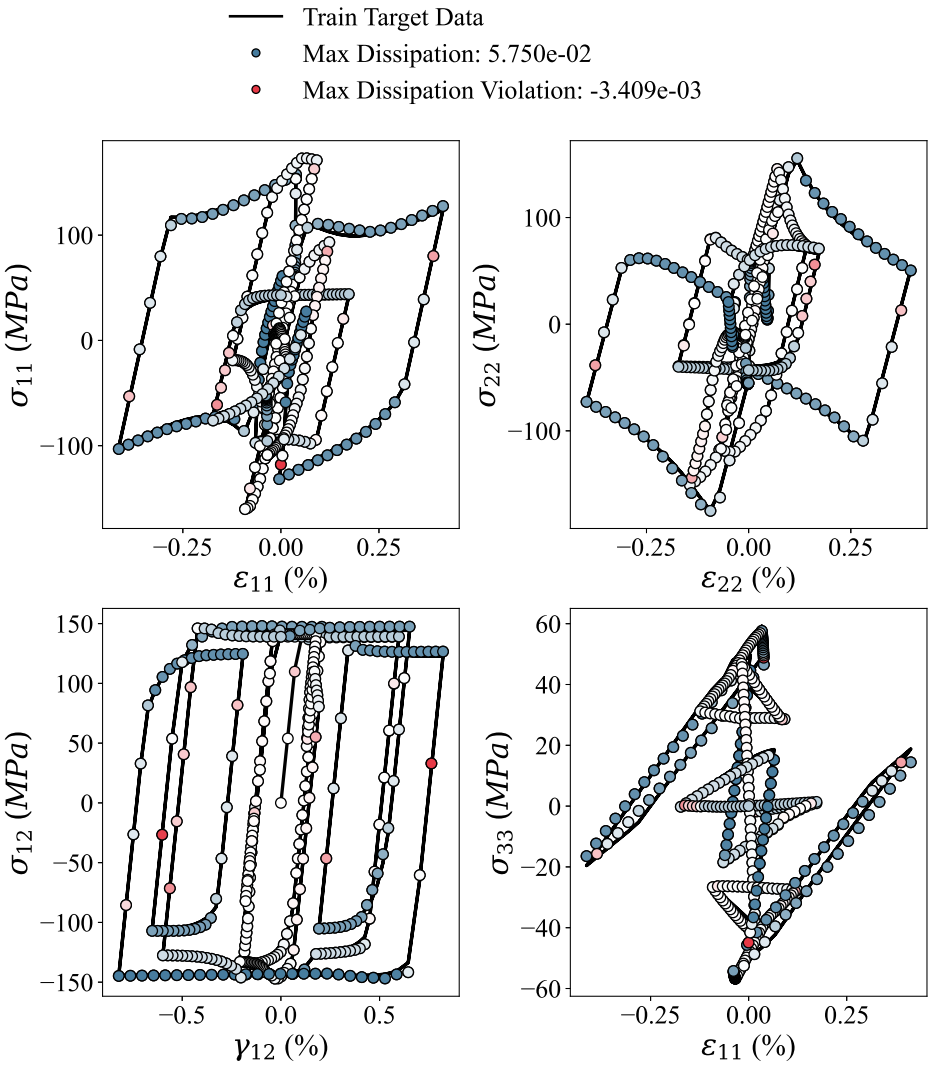


Figure 9.2.19: Stress-strain diagrams with dissipation criterion for plane strain perfect elastoplasticity, Data Series 10.

Comparison to Alternative Network Architectures

In this section the performance of the presented strategy is compared to other generic recurrent neural network architectures. The considered methodologies are Basic Recurrent Neural Network (RNN), Basic Recurrent Neural Network with Gradient Clippling (RNN+clip), Gated Recurrent Units (GRU) and Long-Short Term Memory (LSTM). All architectures are tested in a Python environment based on the library PyTorch and trained with back-propagation on the same data presented in Section 9.2.2. For each strategy, network architectures of a range of sizes are considered, resulting in different numbers of state variables and network parameters.

Figure 9.2.20 shows the approximation errors achieved by the training processes, displayed over the number of state variables or network parameters. Notably the proposed methodology renders the smallest approximation error for the smallest number of internal state variables. Moreover, all other methods considered require significantly more network parameters to achieve the same level of accuracy. In Figures 9.2.21 and 9.2.22 the correlation coefficient R^2 is displayed over the number of state variables and the number of network parameters for, respectively, the training and the validation data sets. In all diagrams it is observed that the level of accuracy of the present results is not achieved by any of the other methods unless the size of the network is significantly increased.

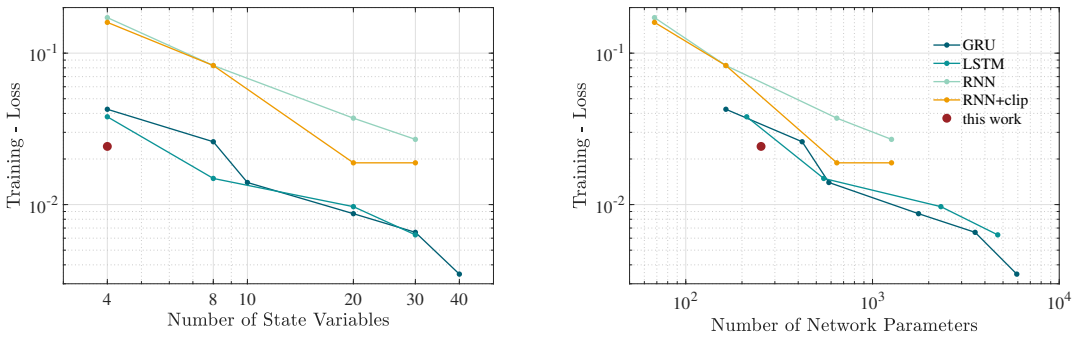


Figure 9.2.20: Loss obtained with different network architectures for plane strain perfect elastoplasticity.

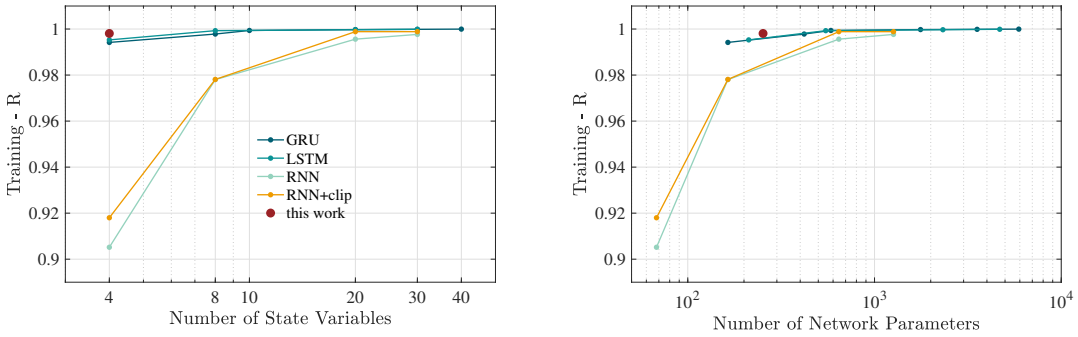


Figure 9.2.21: Training accuracy obtained with different network architectures for plane strain perfect elastoplasticity. The correlation coefficient R^2 is similar to the coefficient shown in Figures 9.2.8 (b) and 9.2.9 (b).

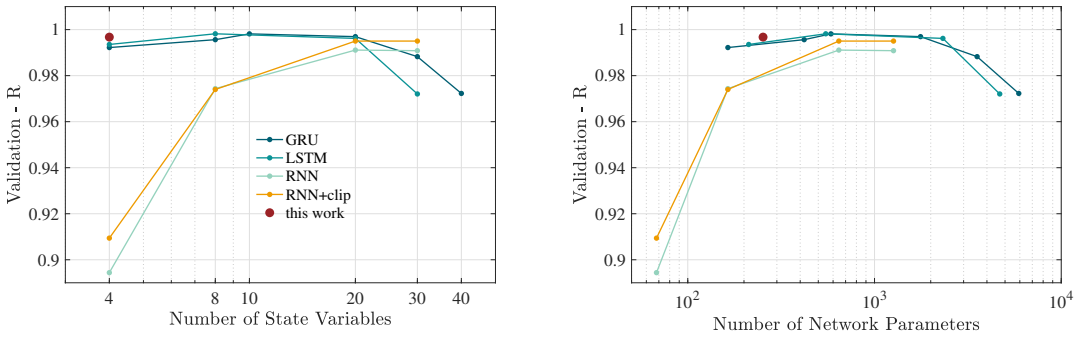


Figure 9.2.22: Validation accuracy obtained with different network architectures for plane strain perfect elastoplasticity. The correlation coefficient R^2 is similar to the coefficient shown in Figures 9.2.8 (b) and 9.2.9 (b).

9.3 Numerical Homogenisation

The training data utilised in this section is generated using numerical homogenisation RVE, which involves employing computational methods to simulate the behavior of a RVE. This simulation facilitates the analysis of the macroscopic properties of solid porous materials (*i.e.* plane strain case).

9.3.1 Example 6: Material Homogenisation - Plane Strain RVE

The performed material homogenisation in small strain analysis is comprehensively explained in [80]. The objective of this example is to employ the proposed methodology to represent the complex micro structure of a material, where the stress responses on macro level are computed for several macro strain data sequences. In the following, the case of plane strain perfect elastoplasticity is considered, (*i.e.* $H = 0$). Similar to Section 9.2.2, in the plane strain state, $\varepsilon_{yz} = \varepsilon_{zx} = \varepsilon_{zz} = 0$ and $\sigma_{yz} = \sigma_{zx} = 0$. Hence, three strain coefficients $\{\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{xy}\}$ induce a response consisting of four stress coefficients $\{\sigma_{xx}, \sigma_{yy}, \sigma_{xy}, \sigma_{zz}\}$.

It had been selected to input five internal state variables to the network-based stress update model, allowing for a representation of some form of internal plastic strains and to capture the pressure sensitivity behaviour of the material. As a result, the state network requires eleven input neurons and five output neurons. The hidden layer has 20 neurons ($11 \rightarrow 20 \rightarrow 5$). Eight input neurons and four output neurons ($8 \rightarrow 4$) are required for the response network. A hidden layer is not required for the response network, as it is not required for the problems discussed in Sections 9.1.2 to 9.2.2. As a consequence, the training procedure must establish 20 biases and 320 weights.

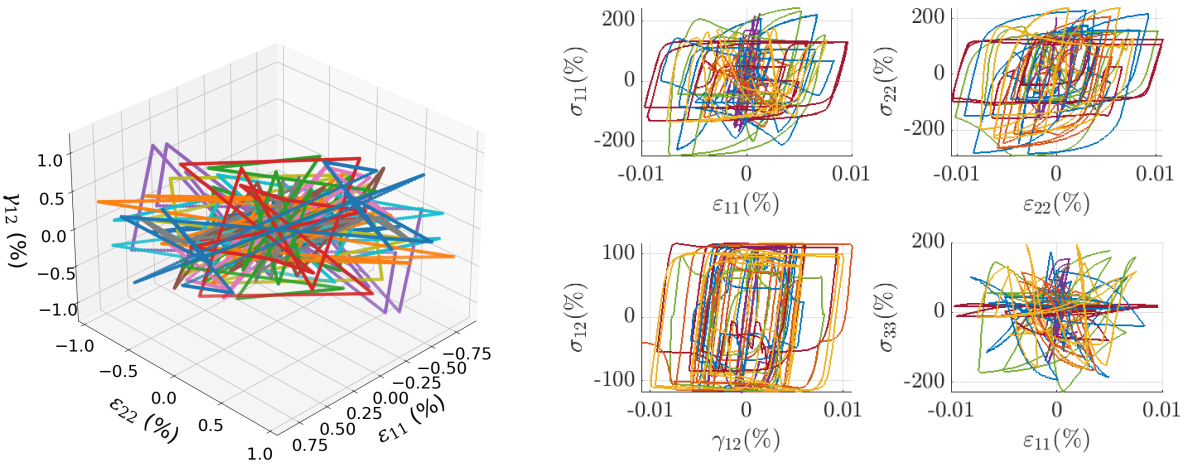
Training and Validation Data

The material parameters employed for the generation of the training and validation data are $E = 200GPa$, $\nu = 0.26$, $\sigma_y = 240MPa$ and $H = 0$ under plane strain condition with elastoplastic von Mises material characteristics. The linear boundary displacement is selected as the RVE boundary conditions. In addition, 2D second order quadratic meshes with 8 node elements are utilised. The number of gauss points $ngp = 8$ and the maximum number of iteration is $imax = 10$.

A random loading path is used to generate the data sequences with RVE that include a hole of $f = 10\%$ shown in Figure 4.4.2. The starting point in the random path is undeformed material state. In addition, the random loading paths are controlled by a defined value that represents a linear interpolation between

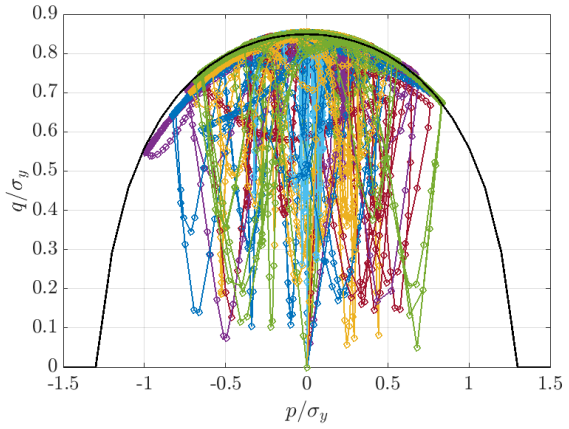
positions in the strain space which is set to be less than 0.3% of imposed volumetric strain *i.e.* $\varepsilon_V = \varepsilon_{xx} + \varepsilon_{yy}$ and do not exceed 3% in any strain component. Beyond these constraints, the choice of strain positions is arbitrary based on suitable interpolation to maintain moderate load step sizes and ensure that the strain data defines a clear deformation path.

Each individual sequence data consist of incremental strains $\boldsymbol{\varepsilon}_{ij}(t)$ and stresses $\boldsymbol{\sigma}_{ij}(t)$ for 350 loading time steps. A total of 32 data sequences are generated where 22 sequences are used for training and 10 sequences are used for validation. Figure 9.3.1 show the generated data sequences to train and validate the network.



(a) (b)

Figure 9.3.1: Different strain sequences (a) and associated stress-strain diagrams (b) for plane strain perfect elastoplasticity.



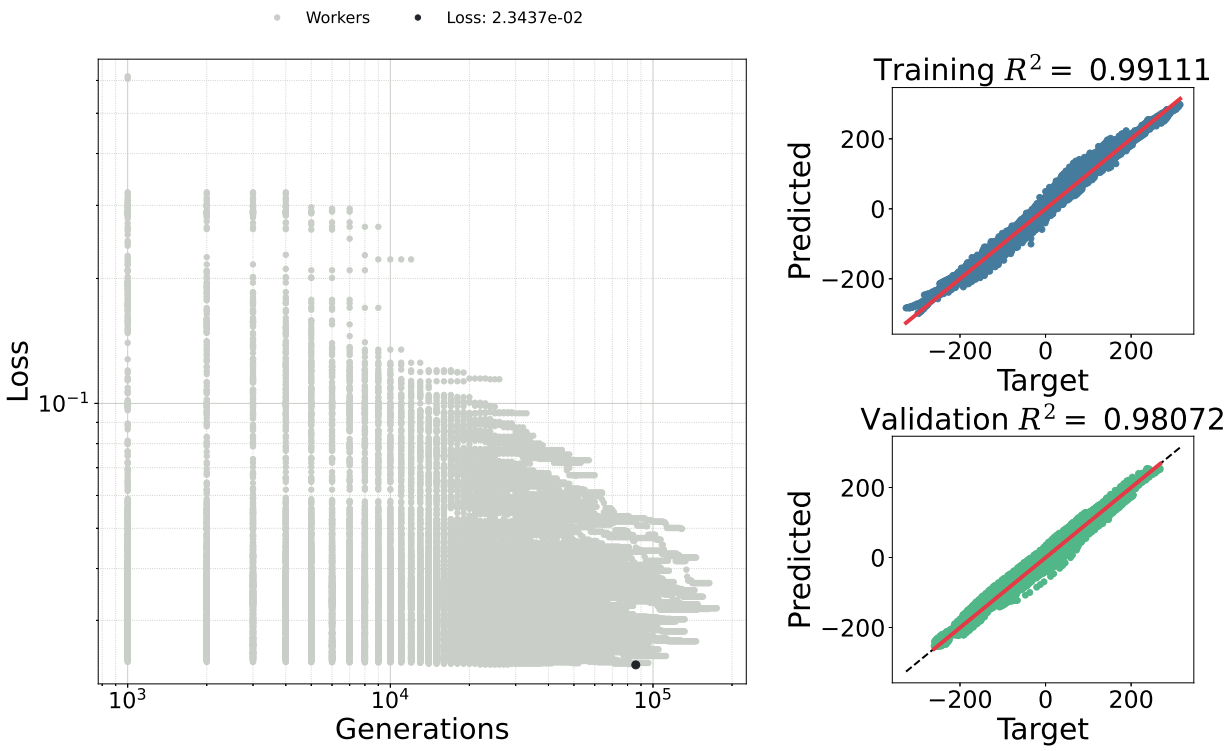
;

Figure 9.3.2: The training and validation data sequences in $\bar{p}$-$\bar{q}$ space.

The yield surface defines the boundary of the elastic domain in elastic-plastic materials (*i.e.* elastic deformations occur only inside the yield surface). In Figure 9.3.2, all the training and validation data is presented in terms of the normalised hydrostatic pressure p/σ_y and the norm of the normalised deviatoric stress q/σ_y. The black solid line corresponding to the Gurson analytical yield surface from Equation (4.4.8) using $q_1 = 1.5$, $q_2 = 1$, and $q_3 = q_1^2$.

Convergence

The results achieved in a total time duration of 3 days training on the cluster with CPU=101. Figure 9.3.3 (a) reports the workers training loss against the number of generations with the minimum loss achieved is 0.0234. In Figure 9.3.3 (b) all stress responses are displayed against their data counterparts. For the data used in training the accuracy is high and the correlation coefficient exceeds the value of 99.1%. For the validation data the correlation coefficient exceeds 98%.



(a) (b)

Figure 9.3.3: Training convergence for RVE plane strain perfect elastoplasticity: Supervised optimisation based on multiple instances of MOPSO (a) and stress responses versus stress data (b).

Results

The network stress response and the stress data for one sequence of training data is shown in Figure 9.3.4. Figures 9.3.5 and 9.3.6 present the stress-strain diagrams for all ten training data series. The associated $\bar{p}$-$\bar{q}$ diagrams are presented in Figures 9.3.7 and 9.3.8. The validation network stress response and the stress data for two sequences is shown in Figure 9.3.9. Figure 9.3.10 show the stress and strain validation diagrams. The corresponding validation $\bar{p}$-$\bar{q}$ diagrams are shown in Figure 9.3.11.

The responses obtained from the proposed neural network based stress update procedure are generally observed to agree accurately with the training data and represent the desired material behavior accurately in Figures 9.3.5 and 9.3.6. With regard to validation in Figure 9.3.10, the model performed poorer than the training results but the obtained accuracy is relatively good. The most notable deviation is noticed in the

norm of the deviatoric stresses, q, in Figures 9.3.7 and 9.3.11. This is due to the nonlinear dependency of q on the stress coefficients: Minimising the average of the approximation errors of the stress coefficients is unlikely to simultaneously minimise the approximation error in q. In addition to the added complexity due to the pressure sensitive behaviour in p. It is also suspected that the particular choices of data preprocessing and scaling prior to training has significant effects as described in Section 8.2.

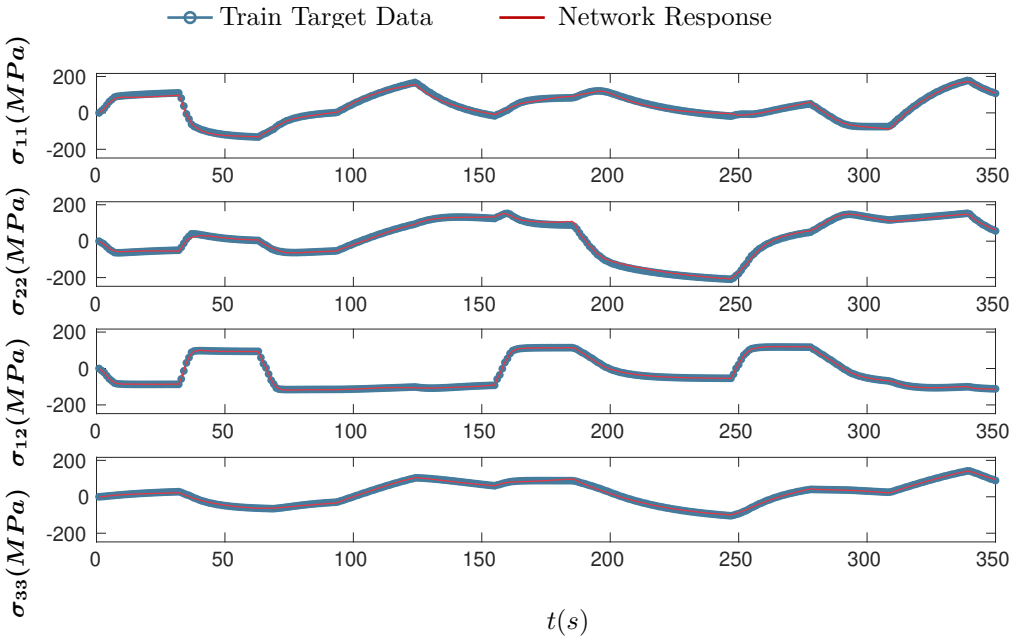


Figure 9.3.4: Stress sequences for training of RVE plane strain perfect elastoplasticity, Data Series 2.

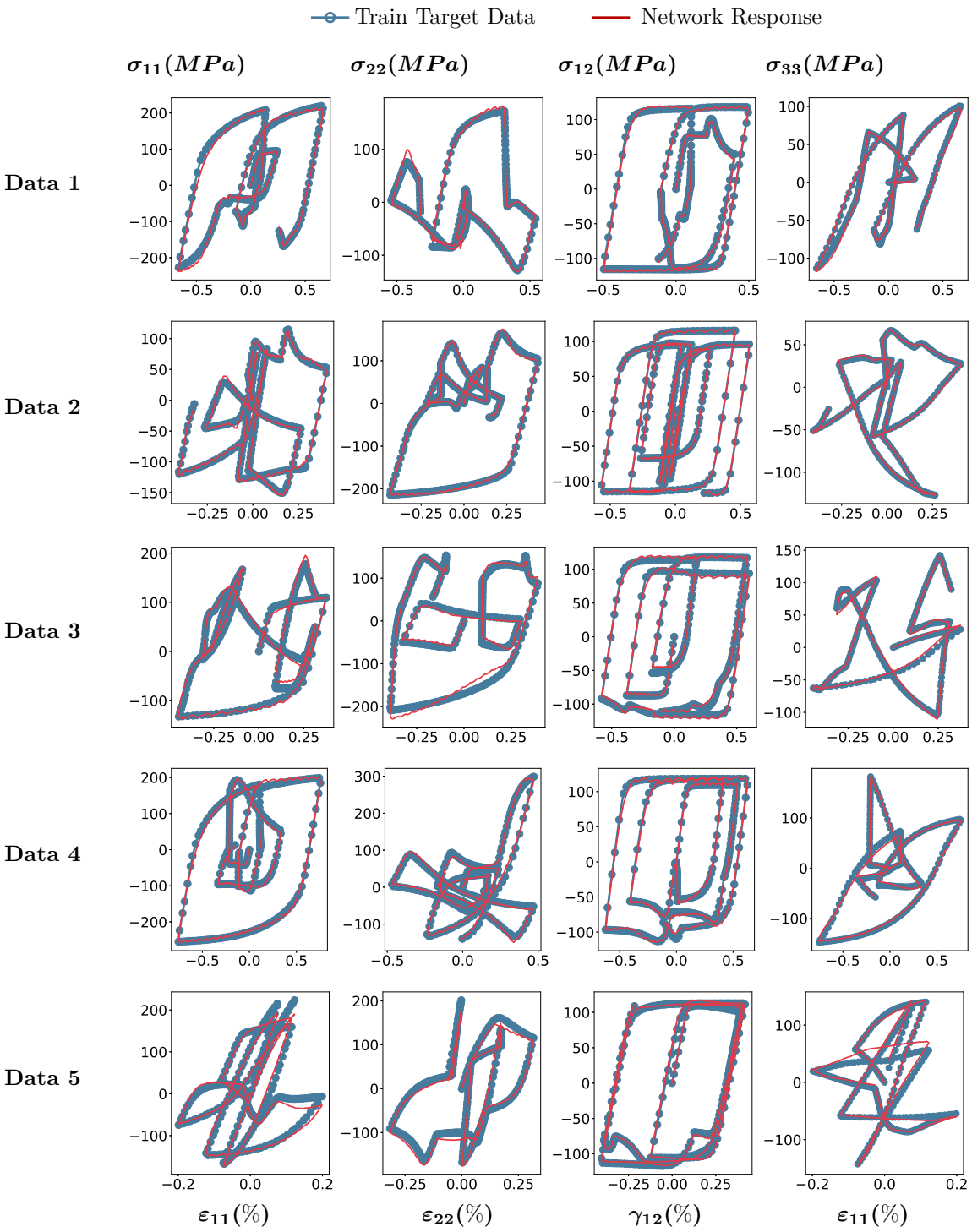


Figure 9.3.5: Stress-strain diagrams for training of plane strain perfect elastoplasticity, Data Series 1 to 5.

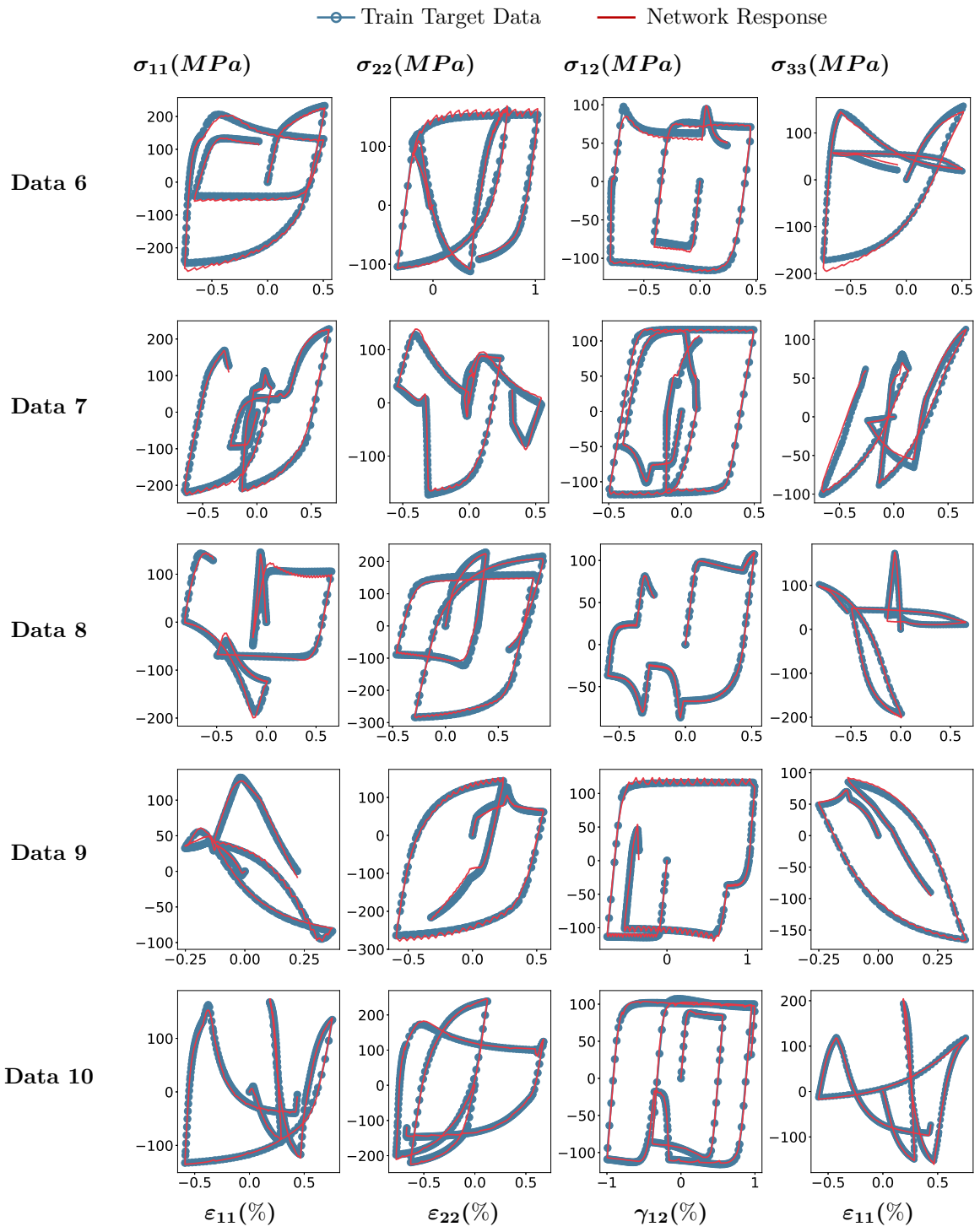


Figure 9.3.6: Stress-strain diagrams for training of plane strain perfect elastoplasticity, Data Series 1 to 5.

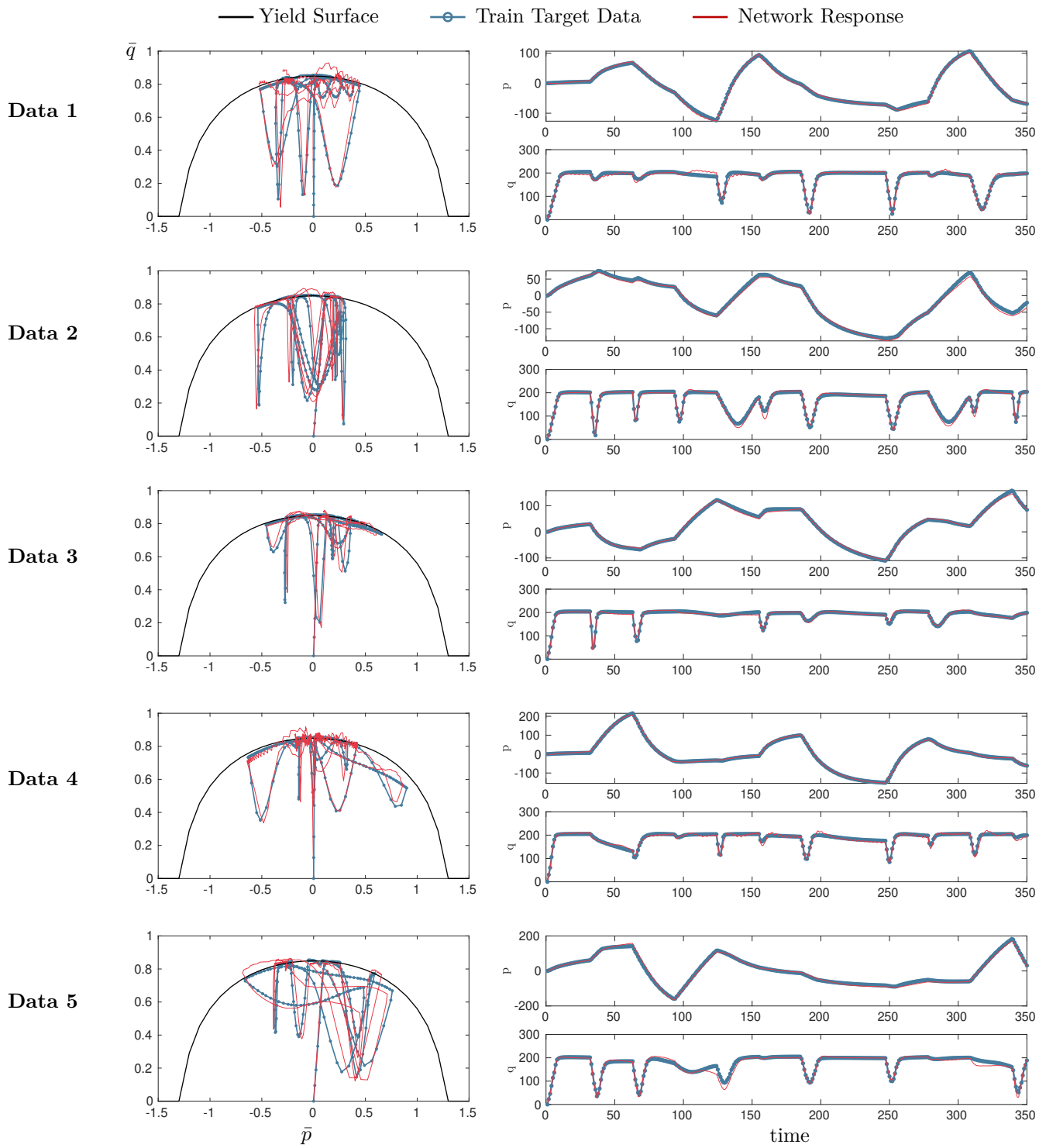


Figure 9.3.7: $\bar{p}$-$\bar{q}$ diagrams for training of plane strain perfect elastoplasticity, Data Sequences 1-5.

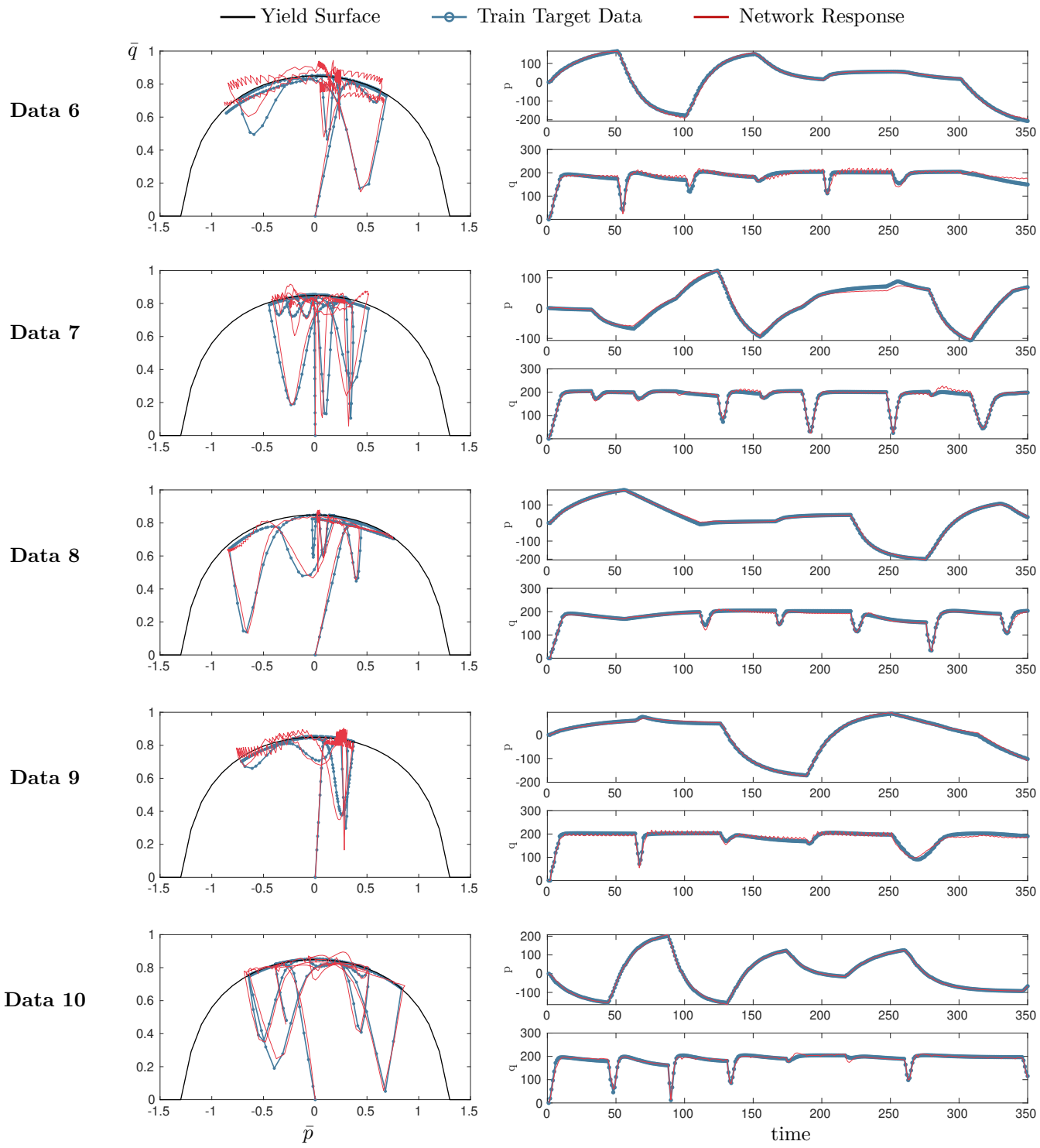


Figure 9.3.8: $\bar{p}$-$\bar{q}$ diagrams for training of plane strain perfect elastoplasticity, Data Sequences 5-10.

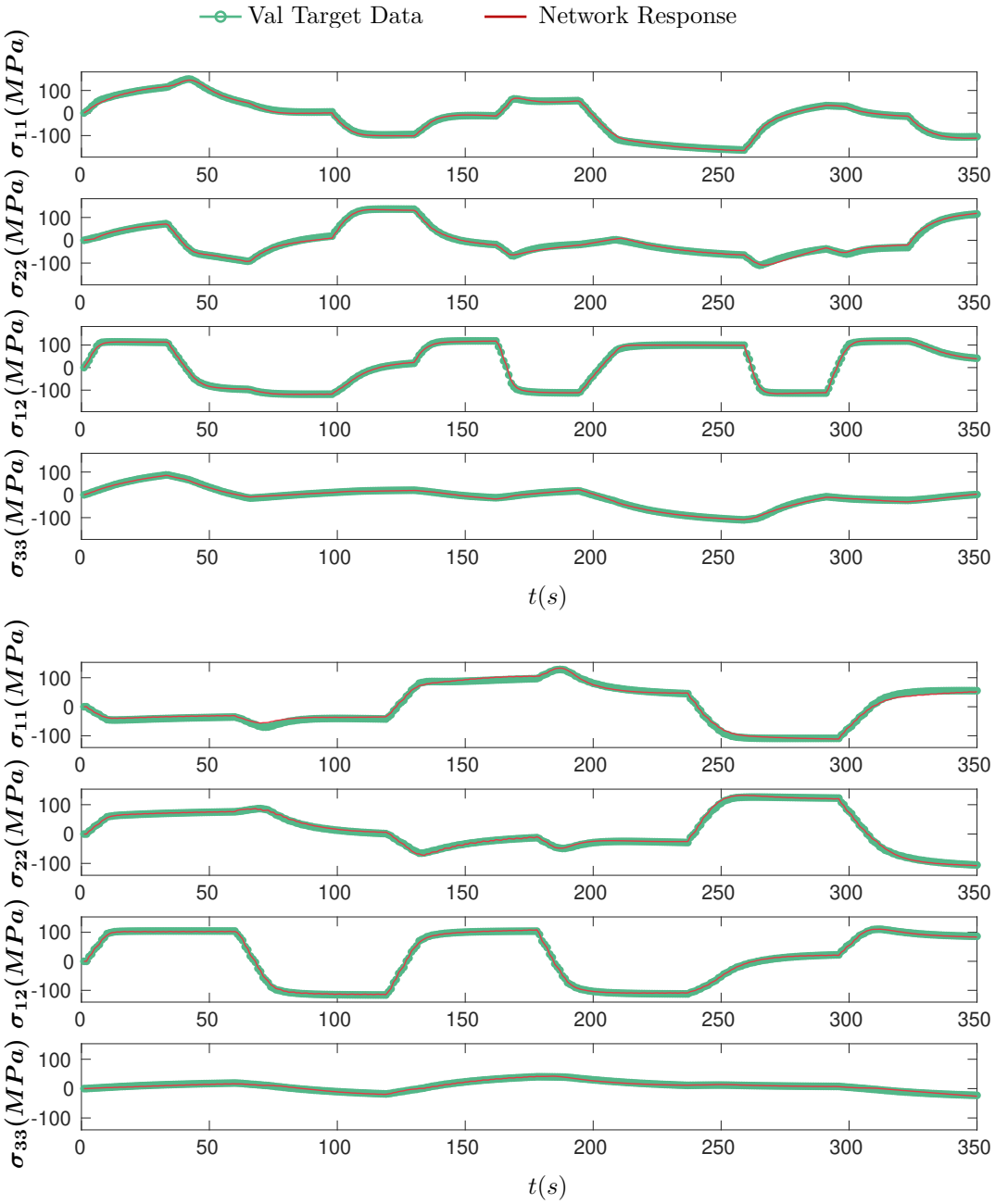


Figure 9.3.9: Stress sequences for training of plane strain perfect elastoplasticity, Data Series 1 and 2.

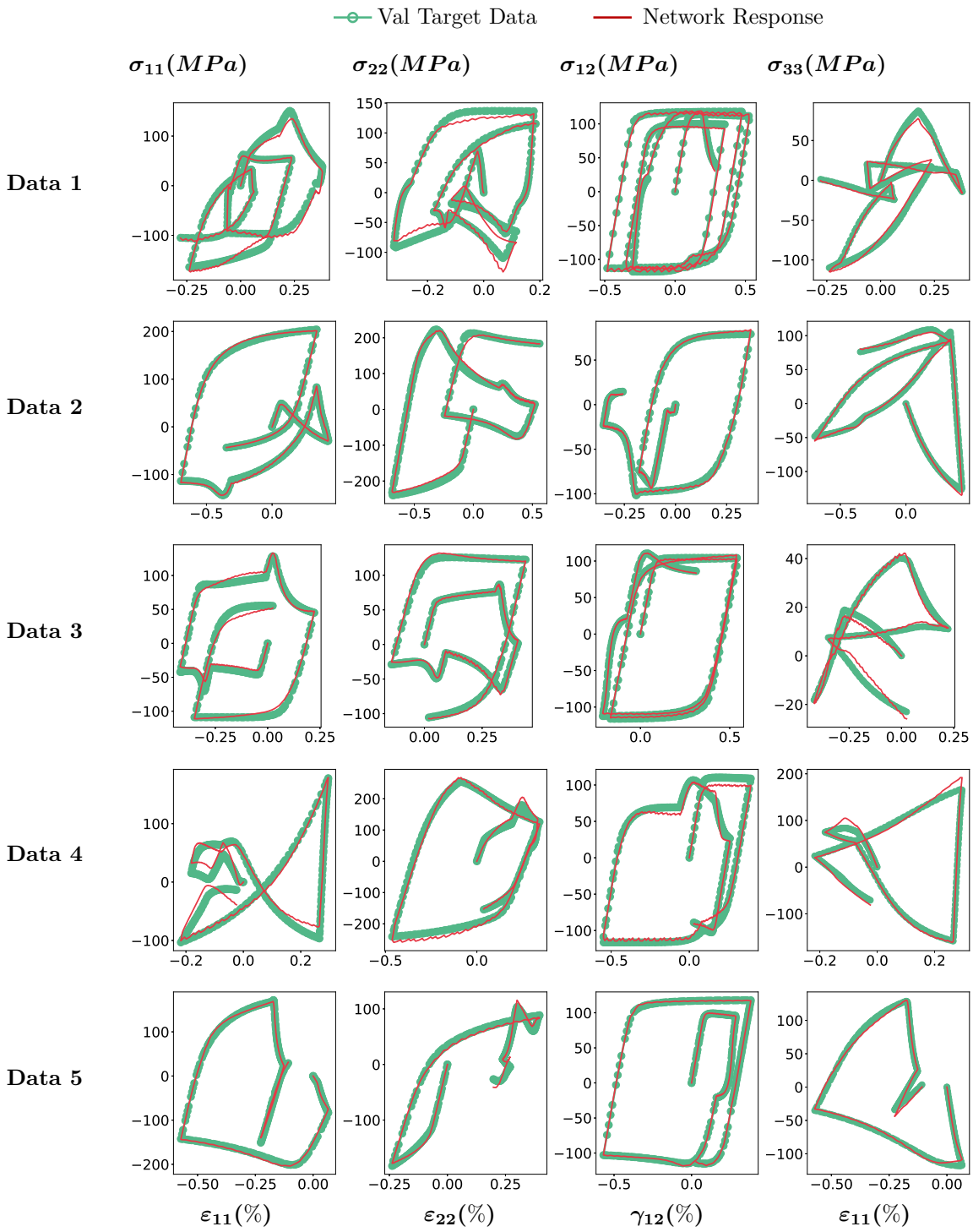


Figure 9.3.10: Stress-strain diagrams for validation of plane strain perfect elastoplasticity.

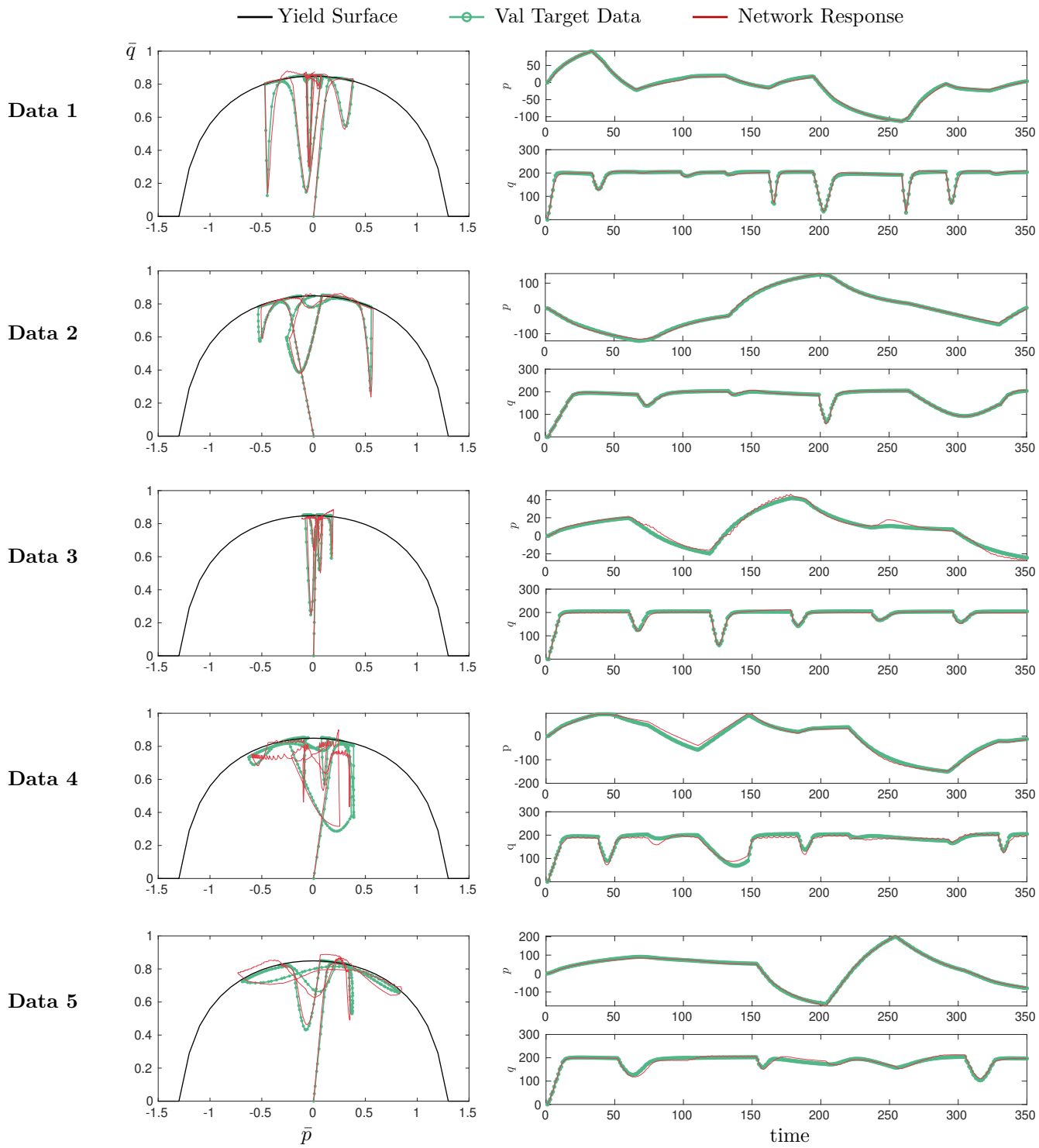


Figure 9.3.11: $\bar{p}$-$\bar{q}$ diagrams for validating of plane strain perfect elastoplasticity.

Thermodynamic Consistency

The dissipation criterion proposed in Equation (7.4.5) is evaluated and visualised for two data series in Figures 9.3.12 and 9.3.13. It can be observed that violation of the criterion is restricted to the load steps immediately following changes of loading direction. The largest negative value does not exceed 13.5% of the largest value of the dissipation D° computed for any of the dissipative load steps.

Data 2

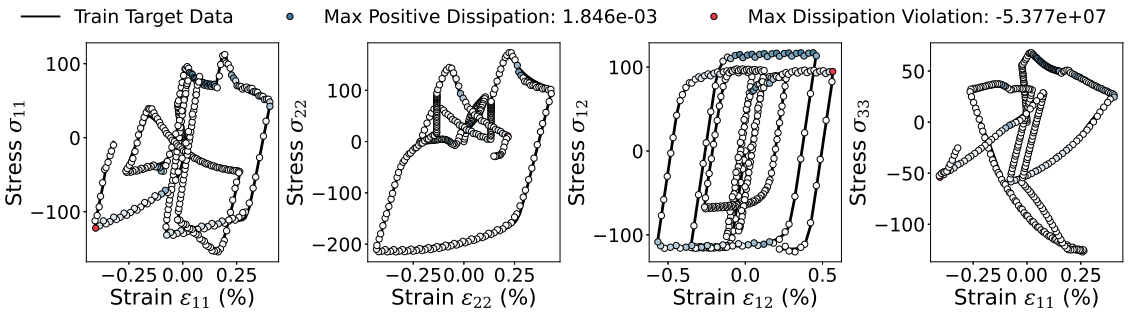


Figure 9.3.12: Stress-strain diagrams with dissipation criterion for Data 2.

Data 10

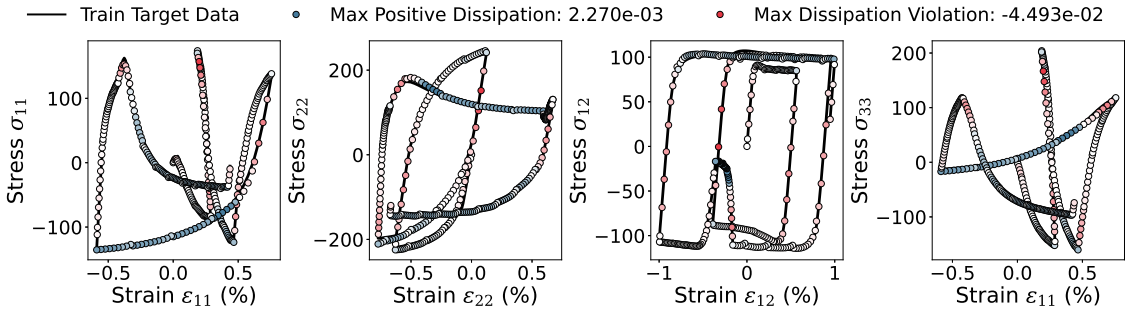


Figure 9.3.13: Stress-strain diagrams with dissipation criterion for Data 10.

Chapter 10

Conclusions

In conclusion, the work proposed a novel neural network based algorithmic constitutive model (state $\mathscr{F}$ and response $\mathscr{G}$ neural networks) with the capability of simulating the complex nonlinear relations of generic rate-independent inelastic solid material characteristics.

10.1 Achievements

Two neural network-based strategies have been proposed and compared to reproduce the constitutive relations of solid materials with high computational efficiency which are history based and internal variable strategies. The employed strategies can be trained on data sequences of stress and strain, generated by performing physical experiments or by material homogenisation. The data used for evaluating the proposed network performance were generated using the von Mises model, and implemented by the return mapping algorithm for elastic behavior and plastic deformation. The networks are formulated to identically represent the piecewise linear constitutive behaviour. Hence, the standard analytical expression used to describe the stress update for uniaxial elastoplasticity is recovered exactly by the models. For history-based with isotropic hardening an acceptable level of accuracy is obtained but inaccuracies are observed when the direction of the strain loading is changed. This indicates that the performance of internal variable-based strategy is better suited to capture the elastoplastic type material behavior.

Numerical examples related to material computational homogenisation have been considered. Two RVE meshes with same total volume but each mesh has different matrix and fiber element sizes and layer distributions. The first mesh consists of concentrated layers, while the second mesh contains distributed layers. The main objective of this example is to illustrate the effect of RVE distribution and density on determining the associated stresses in a plane strain case. The results indicates that increasing the RVE distribution and density results in better approximation. The computed error for the concentrated layers is 10.9%, whereas, for the distributed layers is 4.5%.

The manufactured training data are generated with random loading paths that are controlled with defined change of direction probability value and the minimum and maximum incremental size. The employed training methodology for the recurrent neural network architecture, which is based on gradient-free optimisation of the network associated weights and biases is explained and evaluated on number of numerical examples. These numerical examples including training the strategy to represent data of uniaxial elastoplasticity, uniaxial elastoplastic damage and plane strain elastoplasticity.

The proposed strategy was also tested on modelling the behaviour of porous solid materials. The data used for training were generated through numerical material homogenisation based on an RVE consisting of a von Mises elastoplastic matrix with a variable void volume fraction. The RVE numerical simulation was performed by enforcing the RVE boundary conditions and the stress responses on macro scale are computed for various macro strain data sequences to generate different loading paths.

In addition, a computationally efficient criterion based on two finite steps for testing thermodynamic consistency for trained stress update models is introduced and applied.

10.2 Results

The training and validation results determined by the proposed methodology have been compared against the associate von Mises constitutive model and have shown high accuracy predictions. The 2D plane strain elastoplasticity training proposed methodology is followed by a second training stage for fine tuning based on BPTT to improve the model validation predictions. Furthermore, the results are presented in terms of stress/strain relationship and $p - q$ diagram to ensure that the yield surface is accurately captured by the network.

Furthermore, the trained network based on the proposed strategy is compared to other generic recurrent neural network architecture (*i.e.* RNN, RNN+clip, GRU and LSTM). The obtained results deduced that the proposed methodology requires less complex and smaller network architectures, therefore it is more computational efficient. Moreover, all other methods considered require significantly more network parameters to achieve the same level of accuracy.

The stress update algorithm based on neural networks was then trained and validated, demonstrating high accuracy in describing the material stress/strain relationship. Additionally, the hydrostatic pressure p and deviatoric stress norm q showed good agreement with the corresponding Gurson model for porous elastoplastic material. Therefore, the neural network model developed through this training can be employed in finite element analysis to study real-world applications.

Based on the considered examples it is observed that, the largest positive values coincide with the load steps that induce large plastic deformation, while the largest negative values occur when the direction of the loading changes. The maximum violation of the dissipation criterion in the validation examples of uniaxial perfect elastoplasticity and uniaxial elastoplasticity with hardening lies within machine accuracy. Whereas, for the examples of uniaxial elastoplastic damage and for the 2D plane strain case, the largest negative value does not exceed 9.5% and 6%, respectively, of the largest value of the dissipation D^o computed for any of the dissipative load steps.

The given numerical examples proved that the proposed methodology is generic, trustworthy and comply with physical constraints.

10.3 Future Work

- Investigate the proposed strategy to model the complex nonlinear behaviour of solids and porous solid material in three dimensional space with numerical examples. In this case, the proposed network architecture can be modified and expand for general 3D considerations. Yet, the problem would get more complex in terms of training the algorithm, due to increasing:
  Inputs: 6 strain coefficients which are $\boldsymbol{\varepsilon} = \{\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \varepsilon_{xy}, \varepsilon_{yz}, \varepsilon_{zx}\}$.
  Outputs: 6 associated stress coefficients which are $\boldsymbol{\sigma} = \{\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{zx}\}$.
  Hence, more training data must be generated to cover the 3D space may cause computational cost to be expensive.

- Study the influence of different imposed boundary condition on the RVE such as periodic and traction boundary conditions on training the model. Moreover, generation and training with more complex RVE structure including multiple voids or higher void volume fractions.

- Integration of the proposed thermodynamic criterion into the training process may improve the model performance by discarding inconsistent network parameters at an early stage which would accelerate the training process.

- Integrate the trained model in finite element simulations on a Gauss point level to replace the standard algorithmic constitutive models library [73]. Hence, can be used to study real-world applications.

- Enhancing the training quality by improving the gradient-free optimisation methodology for computational efficiency. This will enable obtaining improved generic trained model that can be validated on unseen data.

- Evaluate the proposed strategy on training data generated with performing physical experiments and investigated the presence of noise in data and deal with cases of insufficient data.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M. et al. [2016], 'Tensorflow: Large-scale machine learning on heterogeneous distributed systems', *arXiv preprint arXiv:1603.04467* .

[2] Al-Haik, M., Hussaini, M. and Garmestani, H. [2006], 'Prediction of nonlinear viscoelastic behavior of polymeric composites using an artificial neural network', *International journal of plasticity* **22**(7), 1367–1392.

[3] Alhayki, R., Muttio, E. J., Dettmer, W. G. and Perić, D. [2022], 'On the performance of different architectures in modelling elasto-plasticity with neural network', *WCCM-APCOM 2022* **1700**.

[4] Anything, A. M. [2015], 'Dynamic memory networks for natural language processing', *Kumar et al. arXiv Pre-Print* **97**.

[5] Bengio, Y., Simard, P. and Frasconi, P. [1994], 'Learning long-term dependencies with gradient descent is difficult', *IEEE transactions on neural networks* **5**(2), 157–166.

[6] Benvidi, A., Abbasi, S., Gharaghani, S., Tezerjani, M. D. and Masoum, S. [2017], 'Spectrophotometric determination of synthetic colorants using pso–ga-ann', *Food chemistry* **220**, 377–384.

[7] Blanco, A., Delgado, M. and Pegalajar, M. C. [2001], 'A real-coded genetic algorithm for training recurrent neural networks', *Neural networks* **14**(1), 93–105.

[8] Blum, C. and Socha, K. [2005], Training feed-forward neural networks with ant colony optimization: An application to pattern classification, *in* 'Fifth International Conference on Hybrid Intelligent Systems (HIS'05)', IEEE, pp. 6–pp.

[9] Bock, F. E., Aydin, R. C., Cyron, C. J., Huber, N., Kalidindi, S. R. and Klusemann, B. [2019], 'A review of the application of machine learning and data mining approaches in continuum materials mechanics', *Frontiers in Materials* **6**, 110.

[10] Bonatti, C. and Mohr, D. [2021], 'One for all: Universal material model based on minimal state-space neural networks', *Science Advances* **7**(26), eabf3658.

[11] Bonatti, C. and Mohr, D. [2022], 'On the importance of self-consistency in recurrent neural network models representing elasto-plastic solids', *Journal of the Mechanics and Physics of Solids* **158**, 104697.

[12] Brough, D. B., Wheeler, D., Warren, J. A. and Kalidindi, S. R. [2017], 'Microstructure-based knowledge systems for capturing process-structure evolution linkages', *Current Opinion in Solid State and Materials Science* **21**(3), 129–140.

[13] Cazacu, O., Revil-Baudard, B., Lebensohn, R. A. and Gărăjeu, M. [2013], 'On the combined effect of pressure and third invariant on yielding of porous solids with von mises matrix', *Journal of Applied Mechanics* **80**(6).

[14] Che, Z., Purushotham, S., Cho, K., Sontag, D. and Liu, Y. [2018], 'Recurrent neural networks for multivariate time series with missing values', *Scientific reports* **8**(1), 1–12.

[15] Chen, G. [2021], 'Recurrent neural networks (rnns) learn the constitutive law of viscoelasticity', *Computational Mechanics* **67**(3), 1009–1019.

[16] Chen, Q., Jia, R. and Pang, S. [2021], 'Deep long short-term memory neural network for accelerated elastoplastic analysis of heterogeneous materials: An integrated data-driven surrogate approach', *Composite Structures* **264**, 113688.

[17] Cho, K., Van Merriënboer, B., Bahdanau, D. and Bengio, Y. [2014], 'On the properties of neural machine translation: Encoder-decoder approaches', *arXiv preprint arXiv:1409.1259* .

[18] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. [2014], 'Learning phrase representations using rnn encoder-decoder for statistical machine translation', *arXiv preprint arXiv:1406.1078* .

[19] Christensen, R. M. [2012], *Mechanics of composite materials*, Courier Corporation.

[20] Chu, S.-C., Roddick, J. F. and Pan, J.-S. [2005], 'A parallel particle swarm optimization algorithm with communication strategies', *J. Inf. Sci. Eng* **21**(4), 809–818.

[21] Ciregan, D., Meier, U. and Schmidhuber, J. [2012], Multi-column deep neural networks for image classification, *in* '2012 IEEE conference on computer vision and pattern recognition', IEEE, pp. 3642–3649.

[22] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P. [2011], 'Natural language processing (almost) from scratch', *Journal of machine learning research* **12**(ARTICLE), 2493–2537.

[23] de Souza Neto, E. A., Peric, D. and Owen, D. R. [2011], *Computational methods for plasticity: theory and applications*, John Wiley & Sons.

[24] de Souza Neto, E. and Feijóo, R. [2006], 'Variational foundations of multi-scale constitutive models of solid: small and large strain kinematical formulation', *LNCC Research & Development Report* **16**.

[25] Ding, S., Su, C. and Yu, J. [2011], 'An optimizing bp neural network algorithm based on genetic algorithm', *Artificial intelligence review* **36**, 153–162.

[26] Dokur, Z. and Ölmez, T. [2001], 'Ecg beat classification by a novel hybrid neural network', *Computer methods and programs in biomedicine* **66**(2-3), 167–181.

[27] Duong, S. C., Uezato, E., Kinjo, H. and Yamamoto, T. [2012], 'A hybrid evolutionary algorithm for recurrent neural network control of a three-dimensional tower crane', *Automation in Construction* **23**, 55–63.

[28] Eggersmann, R., Kirchdoerfer, T., Reese, S., Stainier, L. and Ortiz, M. [2019], 'Model-free data-driven inelasticity', *Computer Methods in Applied Mechanics and Engineering* **350**, 81–99.

[29] Fernández, M., Rezaei, S., Rezaei Mianroodi, J., Fritzen, F. and Reese, S. [2020], 'Application of artificial neural networks for the prediction of interface mechanics: a study on grain boundary constitutive behavior', *Advanced Modeling and Simulation in Engineering Sciences* **7**(1), 1–27.

[30] Frankel, A. L., Jones, R. E., Alleman, C. and Templeton, J. A. [2019], 'Predicting the mechanical response of oligocrystals with deep learning', *Computational Materials Science* **169**, 109099.

[31] Furukawa, T. and Hoffman, M. [2004], 'Accurate cyclic plastic analysis using a neural network material model', *Engineering Analysis with Boundary Elements* **28**(3), 195–204.

[32] Furukawa, T. and Yagawa, G. [1998], 'Implicit constitutive modelling for viscoplasticity using neural networks', *International Journal for Numerical Methods in Engineering* **43**(2), 195–219.

[33] Garro, B. A., Rodríguez, K. and Vázquez, R. A. [2016], 'Classification of dna microarrays using artificial neural networks and abc algorithm', *Applied Soft Computing* **38**, 548–560.

[34] Geers, M. G., Kouznetsova, V. G. and Brekelmans, W. [2010], 'Multi-scale computational homogenization: Trends and challenges', *Journal of computational and applied mathematics* **234**(7), 2175–2182.

[35] Gerbaud, P.-W., Néron, D. and Ladevèze, P. [2022], 'Data-driven elasto-(visco)-plasticity involving hidden state variables', *Computer Methods in Applied Mechanics and Engineering* **402**, 115394.

[36] Géron, A. [2019], *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*, " O'Reilly Media, Inc.".

[37] Gers, F. A., Schraudolph, N. N. and Schmidhuber, J. [2002], 'Learning precise timing with lstm recurrent networks', *Journal of machine learning research* **3**(Aug), 115–143.

[38] Ghaboussi, J., Garrett Jr, J. and Wu, X. [1991], 'Knowledge-based modeling of material behavior with neural networks', *Journal of engineering mechanics* **117**(1), 132–153.

[39] Giusti, S., Blanco, P., de Souza Neto, E. and Feijóo, R. [2009], 'An assessment of the gurson yield criterion by a computational multi-scale approach', *Engineering Computations* .

[40] Gorji, M. B., Mozaffar, M., Heidenreich, J. N., Cao, J. and Mohr, D. [2020], 'On the potential of recurrent neural networks for modeling path dependent plasticity', *Journal of the Mechanics and Physics of Solids* **143**, 103972.

[41] 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'.

[42] Heider, Y., Wang, K. and Sun, W. [2020], 'So (3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials', *Computer Methods in Applied Mechanics and Engineering* **363**, 112875.

[43] Hill, R. [1967], 'The essential structure of constitutive laws for metal composites and polycrystals', *Journal of the Mechanics and Physics of Solids* **15**(2), 79–95.

[44] Hinton, M. and Kaddour, A. [2013], 'Triaxial test results for fibre-reinforced composites: The second world-wide failure exercise benchmark data', *Journal of Composite Materials* **47**(6-7), 653–678.

[45] Hinton, M., Kaddour, A. S. and Soden, P. D. [2004], *Failure criteria in fibre reinforced polymer composites: the world-wide failure exercise*, Elsevier.

[46] Hochreiter, S. and Schmidhuber, J. [1997], 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

[47] Holzapfel, A. G. [2000], 'Nonlinear solid mechanics ii'.

[48] Holzapfel, G. A. [1996], 'On large strain viscoelasticity: continuum formulation and finite element applications to elastomeric structures', *International Journal for Numerical Methods in Engineering* **39**(22), 3903–3926.

[49] Huang, D., Fuhg, J. N., Weißenfels, C. and Wriggers, P. [2020], 'A machine learning based plasticity model using proper orthogonal decomposition', *Computer Methods in Applied Mechanics and Engineering* **365**, 113008.

[50] Jobba, M., Mishra, R. and Niewczas, M. [2015], 'Flow stress and work-hardening behaviour of al–mg binary alloys', *International Journal of Plasticity* **65**, 43–60.

[51] Jordan, B., Gorji, M. B. and Mohr, D. [2020], 'Neural network model describing the temperature-and rate-dependent stress-strain response of polypropylene', *International Journal of Plasticity* **135**, 102811.

[52] Juang, C.-F. [2004], 'A hybrid of genetic algorithm and particle swarm optimization for recurrent network design', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **34**(2), 997–1006.

[53] Jung, S. and Ghaboussi, J. [2006], 'Neural network constitutive model for rate-dependent materials', *Computers & Structures* **84**(15-16), 955–963.

[54] Kalanta, S. [1995], 'The equilibrium finite elements in computation of elastic structures', *Statyba* **1**(1), 25–47.

[55] Kanouté, P., Boso, D., Chaboche, J.-L. and Schrefler, B. [2009], 'Multiscale methods for composites: a review', *Archives of Computational Methods in Engineering* **16**, 31–75.

[56] Karaboga, D. and Ozturk, C. [2009], 'Neural networks training by artificial bee colony algorithm on pattern classification', *Neural Network World* **19**(3), 279.

[57] Le, B., Yvonnet, J. and He, Q.-C. [2015], 'Computational homogenization of nonlinear elastic materials using neural networks', *International Journal for Numerical Methods in Engineering* **104**(12), 1061–1084.

[58] Lefik, M., Boso, D. and Schrefler, B. [2009], 'Artificial neural networks in numerical modelling of composites', *Computer Methods in Applied Mechanics and Engineering* **198**(21-26), 1785–1804.

[59] Lefik, M. and Schrefler, B. A. [2003], 'Artificial neural network as an incremental non-linear constitutive model for a finite element code', *Computer methods in applied mechanics and engineering* **192**(28-30), 3265–3283.

[60] Lemaitre, J. [1996], Analysis of crack initiation in structures, *in* 'A Course on Damage Mechanics', Springer, pp. 154–221.

[61] Liu, X., Tao, F., Du, H., Yu, W. and Xu, K. [2020], 'Learning nonlinear constitutive laws using neural network models based on indirectly measurable data', *Journal of Applied Mechanics* **87**(8), 081003.

[62] Liu, Z., Liu, A., Wang, C. and Niu, Z. [2004], 'Evolving neural network using real coded genetic algorithm (ga) for multispectral image classification', *Future Generation Computer Systems* **20**(7), 1119–1129.

[63] Liu, Z., Wu, C. and Koishi, M. [2019], 'A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials', *Computer Methods in Applied Mechanics and Engineering* **345**, 1138–1168.

[64] Lu, S., Lu, Z., Phillips, P., Wang, S., Wu, J. and Zhang, Y. [2016], Fruit classification by hpa-slfn, *in* '2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)', IEEE, pp. 1–5.

[65] Mandel, J. [1966], Contribution théorique à l'étude de l'écrouissage et des lois de l'écoulement plastique, *in* 'Applied Mechanics: Proceedings of the Eleventh International Congress of Applied Mechanics Munich (Germany) 1964', Springer, pp. 502–509.

[66] Masi, F., Stefanou, I., Vannucci, P. and Maffi-Berthier, V. [2021], 'Thermodynamics-based artificial neural networks for constitutive modeling', *Journal of the Mechanics and Physics of Solids* **147**, 104277.

[67] Mavrovouniotis, M. and Yang, S. [2013], Evolving neural networks using ant colony optimization with pheromone trail limits, *in* '2013 13th UK Workshop on Computational Intelligence (UKCI)', IEEE, pp. 16–23.

[68] Menezes Jr, J. M. P. and Barreto, G. A. [2008], 'Long-term time series prediction with the narx network: An empirical evaluation', *Neurocomputing* **71**(16-18), 3335–3343.

[69] Michel, J.-C., Moulinec, H. and Suquet, P. [1999], 'Effective properties of composite materials with periodic microstructure: a computational approach', *Computer methods in applied mechanics and engineering* **172**(1-4), 109–143.

[70] Mitchell, M. [2019], *Artificial intelligence: A guide for thinking humans*, Penguin UK.

[71] Monchiet, V., Charkaluk, E. and Kondo, D. [2011], 'A micromechanics-based modification of the gurson criterion by using eshelby-like velocity fields', *European Journal of Mechanics-A/Solids* **30**(6), 940–949.

[72] Mozaffar, M., Bostanabad, R., Chen, W., Ehmann, K., Cao, J. and Bessa, M. [2019], 'Deep learning predicts path-dependent plasticity', *Proceedings of the National Academy of Sciences* **116**(52), 26414–26420.

[73] Muttio EJ, Alhayki R, D. W. and Peric D, F. L. [2024], 'A neural network-based surrogate model for inelastic solid materials simulations'.

[74] Nemat-Nasser, S. and Hori, M. [2013], *Micromechanics: overall properties of heterogeneous materials*, Elsevier.

[75] Nguyen, L. T. K. and Keip, M.-A. [2018], 'A data-driven approach to nonlinear elasticity', *Computers & Structures* **194**, 97–115.

[76] Nik, A. A., Nejad, F. M. and Zakeri, H. [2016], 'Hybrid pso and ga approach for optimizing surveyed asphalt pavement inspection units in massive network', *Automation in Construction* **71**, 325–345.

[77] Olorunda, O. and Engelbrecht, A. P. [2008], Measuring exploration/exploitation in particle swarms using swarm diversity, *in* '2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)', IEEE, pp. 1128–1134.

[78] Pascanu, R., Mikolov, T. and Bengio, Y. [2013], On the difficulty of training recurrent neural networks, *in* 'International conference on machine learning', PMLR, pp. 1310–1318.

[79] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. et al. [2019], 'Pytorch: An imperative style, high-performance deep learning library', *Advances in neural information processing systems* **32**.

[80] Perić, D., de Souza Neto, E., Feijóo, R., Partovi, M. and Molina, A. C. [2011], 'On micro-to-macro transitions for multi-scale analysis of non-linear heterogeneous materials: unified variational basis and finite element implementation', *International Journal for Numerical Methods in Engineering* **87**(1-5), 149–170.

[81] Ranjbar, M. and Perić, D. [2021], 'A virtual testing strategy to determine effective yield criteria for porous pressure sensitive solids', *International Journal of Rock Mechanics and Mining Sciences* **138**, 104625.

[82] Rehmer, A. and Kroll, A. [2020], 'On the vanishing and exploding gradient problem in gated recurrent units', *IFAC-PapersOnLine* **53**(2), 1243–1248.

[83] Reyes-Sierra, M., Coello, C. C. et al. [2006], 'Multi-objective particle swarm optimizers: A survey of the state-of-the-art', *International journal of computational intelligence research* **2**(3), 287–308.

[84] Robinson, J., Sinton, S. and Rahmat-Samii, Y. [2002], Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna, *in* 'IEEE Antennas and Propagation Society International Symposium (IEEE Cat. No. 02CH37313)', Vol. 1, IEEE, pp. 314–317.

[85] Rumelhart, D. E., Hinton, G. E. and Williams, R. J. [1985], Learning internal representations by error propagation, Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

[86] Samek, W., Binder, A., Montavon, G., Lapuschkin, S. and Müller, K.-R. [2016], 'Evaluating the visualization of what a deep neural network has learned', *IEEE transactions on neural networks and learning systems* **28**(11), 2660–2673.

[87] Schutte, J. F., Reinbolt, J. A., Fregly, B. J., Haftka, R. T. and George, A. D. [2004], 'Parallel global optimization with the particle swarm algorithm', *International journal for numerical methods in engineering* **61**(13), 2296–2315.

[88] Settgast, C., Abendroth, M. and Kuna, M. [2019], 'Constitutive modeling of plastic deformation behavior of open-cell foam structures using neural networks', *Mechanics of Materials* **131**, 1–10.

[89] Shang, L., Lu, Z. and Li, H. [2015], 'Neural responding machine for short-text conversation', *arXiv preprint arXiv:1503.02364* .

[90] Shen, Y., Chandrashekhara, K., Breig, W. and Oliver, L. [2005], 'Finite element analysis of v-ribbed belts using neural network based hyperelastic material model', *International Journal of Non-Linear Mechanics* **40**(6), 875–890.

[91] Simo, J. C. and Hughes, T. J. [2006], *Computational inelasticity*, Vol. 7, Springer Science & Business Media.

[92] Simo, J. C. and Taylor, R. L. [1985], 'Consistent tangent operators for rate-independent elastoplasticity', *Computer methods in applied mechanics and engineering* **48**(1), 101–118.

[93] Simo, J. and Govindjee, S. [1991], 'Non-linear b-stability and symmetry preserving return mapping algorithms for plasticity and viscoplasticity', *International Journal for Numerical Methods in Engineering* **31**(1), 151–176.

[94] Simo, J. and Hughes, T. [1998], 'Objective integration algorithms for rate formulations of elastoplasticity', *Computational inelasticity* pp. 276–299.

[95] Singh, P. and Kottath, R. [2021], 'An ensemble approach to meta-heuristic algorithms: comparative analysis and its applications', *Computers & Industrial Engineering* **162**, 107739.

[96] Sola, J. and Sevilla, J. [1997], 'Importance of input data normalization for the application of neural networks to complex industrial problems', *IEEE Transactions on nuclear science* **44**(3), 1464–1468.

[97] Somer, D. D., Peric, D., de Souza Neto, E. A. and Dettmer, W. G. [2015], 'Yield surfaces of heterogeneous media with debonded inclusions', *Engineering Computations* .

[98] Souza Neto, E., Peric, D. and Owen, D. [2002], 'Computational plasticity: small and large strain finite element analysis of elastic and inelastic solids', *Classroom notes University College of Swansea, Wales* .

[99] Such, F. P., Madhavan, V., Conti, E., Lehman, J., Stanley, K. O. and Clune, J. [2017], 'Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning', *arXiv preprint arXiv:1712.06567* .

[100] Taylor, G. I. [1938], 'Plastic strain in metals.', *our. Inst. Metals.* **62**, 307–324.

[101] Teranishi, M. [2022], 'Neural network constitutive model for uniaxial cyclic plasticity based on return mapping algorithm', *Mechanics Research Communications* **119**, 103815.

[102] Tvergaard, V. [1981], 'Influence of voids on shear band instabilities under plane strain conditions', *International Journal of fracture* **17**(4), 389–407.

[103] Valian, E., Mohanna, S. and Tavakoli, S. [2011], 'Improved cuckoo search algorithm for feedforward neural network training', *International Journal of Artificial Intelligence & Applications* **2**(3), 36–43.

[104] Venini, P. and Morana, P. [2001], 'An adaptive wavelet-galerkin method for an elastic-plastic-damage constitutive model: 1d problem', *Computer methods in applied mechanics and engineering* **190**(42), 5619–5638.

[105] Vlassis, N. N. and Sun, W. [2021], 'Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening', *Computer Methods in Applied Mechanics and Engineering* **377**, 113695.

[106] Waintraub, M., Schirru, R. and Pereira, C. M. [2009], 'Multiprocessor modeling of parallel particle swarm optimization applied to nuclear engineering problems', *Progress in Nuclear Energy* **51**(6-7), 680–688.

[107] Wang, K. and Sun, W. [2019], 'Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning', *Computer Methods in Applied Mechanics and Engineering* **346**, 216–241.

[108] Wansasueb, K., Bureerat, S. and Kumar, S. [2021], 'Ensemble of four metaheuristic using a weighted sum technique for aircraft wing design', *Engineering and Applied Science Research* **48**(4), 385–396.

[109] Wu, L., Kilingar, N. G., Noels, L. et al. [2020], 'A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths', *Computer Methods in Applied Mechanics and Engineering* **369**, 113234.

[110] Yang, Z., Yabansu, Y. C., Al-Bahrani, R., Liao, W.-k., Choudhary, A. N., Kalidindi, S. R. and Agrawal, A. [2018], 'Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets', *Computational Materials Science* **151**, 278–287.

[111] Yin, D., Boehlert, C., Long, L., Huang, G., Zhou, H., Zheng, J. and Wang, Q. [2021], 'Tension-compression asymmetry and the underlying slip/twinning activity in extruded mg–y sheets', *International Journal of Plasticity* **136**, 102878.

[112] Yun, G. J., Ghaboussi, J. and Elnashai, A. S. [2008], 'A new neural network-based model for hysteretic behavior of materials', *International Journal for Numerical Methods in Engineering* **73**(4), 447–469.

[113] Yvonnet, J. [2019], *Computational homogenization of heterogeneous materials with finite elements*, Vol. 258, Springer.

[114] Zhang, A. and Mohr, D. [2020], 'Using neural networks to represent von mises plasticity with isotropic hardening', *International Journal of Plasticity* **132**, 102732.

[115] Zhang, C., Shao, H. and Li, Y. [2000], Particle swarm optimisation for evolving artificial neural network, *in* 'Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0', Vol. 4, IEEE, pp. 2487–2490.

[116] Zhang, J.-R., Zhang, J., Lok, T.-M. and Lyu, M. R. [2007], 'A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training', *Applied mathematics and computation* **185**(2), 1026–1037.

[117] Zhang, Y.-D., Wu, L. and Wang, S. [2011], 'Magnetic resonance brain image classification by an improved artificial bee colony algorithm', *Progress In Electromagnetics Research* **116**, 65–79.

[118] Zhang, Y. and Ge, S. S. [2005], 'Design and analysis of a general recurrent neural network model for time-varying matrix inversion', *IEEE Transactions on Neural Networks* **16**(6), 1477–1490.

[119] Zhang, Y., Jiang, D. and Wang, J. [2002], 'A recurrent neural network for solving sylvester equation with time-varying coefficients', *IEEE Transactions on Neural Networks* **13**(5), 1053–1063.