Computational Mechanics
https://doi.org/10.1007/s00466-025-02613-5

ORIGINAL PAPER

Multiscale modeling of viscoelastic shell structures with artificial neural networks

Jeremy Geiger[1] · **Werner Wagner**[1] · **Steffen Freitag**[1]

Received: 28 October 2024 / Accepted: 8 February 2025

Abstract
For acquiring the effective response of structures with complex underlying microscopic properties, numerical homogenization schemes have widely been studied in the past decades. In this paper, an artificial neural network (ANN) is trained on effective viscoelastic strain–stress data, which is numerically acquired from a consistent homogenization scheme for shell representative volume elements (RVE). The ANN serves as a feasible surrogate model to overcome the bottleneck of the computationally expensive calculation of the coupled multiscale problem. We show that an ANN can be trained solely on uniaxial strain–stress data gathered from creep and relaxation tests, as well as cyclic loading scenarios on an RVE. Furthermore, the amount of data is reduced by including derivative information into the ANN training process, formally known as Sobolev training. Studies at the material point level reveal, that the ANN material model is capable of approximating arbitrary multiaxial stress–strain states, as well as unknown loading paths. Lastly, the material model is implemented into a finite element program, where the potential of the approach in comparison with multiscale and full-scale 3D solutions is analyzed within challenging numerical examples.

Keywords Multiscale modeling · Shell structures · Artificial neural networks · Viscoelasticity · Sobolev training · Finite element method

1 Introduction

The development of new high performance materials inevitably goes along with the need for efficient models to describe their material behavior. Full 3D solutions of engineering problems with complex underlying microstructures can quickly become computationally inefficient, especially when material or geometric non-linearities are considered. From early works on analytical homogenization procedures, see e.g. [15, 16, 36], which are oftentimes limited to simple microstructures and linear elastic constitutive laws, initial works on numerical schemes (FE^2) evolved [7, 24, 34, 35], becoming a central focus of micromechanics research in recent decades. For an in-depth review on the FE^2 method, the reader is referred to [40].

While the previously mentioned works focus on the coupling between continua on the micro- and macroscale, for certain problems it can be advantageous to follow a coupled approach where shell elements are used on the macroscale. Shell finite elements are not only superior in their numerical performance when it comes to thin, bending dominated problems, but also require less degrees of freedom compared to solid elements. The theoretical background for a numerical homogenization scheme for thin sheets can be found in [4, 8]. Follow up works present a consistently linearized variational formulation for the non-linear equations of the coupled boundary value problem [10], or the application to linear elastic material behavior with the consideration of geometrical non-linearities and buckling on the macroscale [5].

A possible practical engineering application for a multiscale approach are fiber reinforced composite materials. The fibers hereby mostly differ by many orders of magnitude compared to the overall engineering device of interest, making full 3D models unfeasible and motivating two-scale models. Furthermore, the fibers are usually embedded in

✉ Jeremy Geiger
jeremy.geiger@kit.edu

Werner Wagner
werner.wagner@kit.edu

Steffen Freitag
steffen.freitag@kit.edu

[1] Institute for Structural Analysis, Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, Karlsruhe 76131, Germany

some kind of polymer matrix, which can inhibit viscoelastic properties. In [25], a homogenization approach for obtaining the effective viscoelastic response of a composite structure is developed for the coupling of continua on the micro- and macroscale. Related works on viscoelastic homogenization include, e.g., the consideration of microcracks [41], an efficient methodology for computing the effective response in the time domain [44] and, quite recently, the application to composite laminates using shell elements on the macroscale [2].

Despite the fact that research has shown, that multiscale methods are capable of providing good results for the overall macroscopic response of heterogeneous structures, the computational bottleneck of solving the coupled global–local boundary value problem still remains an open issue. Due to the fundamental theoretical research done on the topic of artificial intelligence over the last few decades, e.g. the works of [33, 39], the methodology has also piqued interest in the field of computational mechanics, especially for material modeling. Since the pioneering works of [9, 14, 28], artificial neural networks (ANN) have widely been applied as material models. Hereby, one can distinguish between two kinds of applications: the ANN as a constitutive model trained directly on experimental (or synthetically experimental) data, replacing material models based on classic phenomenological approaches, or alternatively as a surrogate model, where the ANN is trained on numerically generated data from heterogeneous microstructures, gathered from simulations based on the previously mentioned homogenization schemes.

Within this work, the latter is of particular interest. Hereby, one generates an effective strain–stress data set on a representative volume element (RVE), trains a suitable ANN architecture under consideration of the underlying microscopic constitutive response and later simply evaluates the model for a given strain state within a finite element simulation. This has been done for approximating the response of reinforced concrete beams [45], for the modeling of bone structure [13], for composites [29], or nonlinear elastic microstructures [26]. Ongoing research includes the prospect of including thermodynamic consistency into the multiscale scheme, see e.g. [19, 22, 32]. To the best of the author's knowledge, so far there has not been any related work done on ANNs as a surrogate constitutive model for the coupling of shells and continua.

As the prospect of this work is the application of the ANN to model effective viscoelastic material behavior, the ANN has to be capable of capturing time and history dependency. Early works on ANNs for homogeneous viscoelastic materials include [1, 18]. In [3], recurrent neural networks are used to circumvent the need to define history variables and the authors in [38] present an approach with physics augmented neural networks, which meets thermodynamic consistency by construction. An overview of different ANN architectures to model 1D viscoelastic material behavior can be found in [37].

The prospect of this work is to develop a surrogate model for the micro to macro transition of inhomogeneous viscoelastic materials for the coupling of continua and shells on the two scales, respectively. The presented approach includes the following essential new features:

- Definition of a simple feed forward neural network architecture to model the effective viscoelastic constitutive relation for shells,
- Development of an efficient sampling strategy based on experimental setups for numerical generation of the training data,
- Enhancement of the training process by adding derivative information into the neural network training,
- Investigations on the behavior of the ANN model at the material point level,
- Application to challenging numerical examples within a finite element simulation.

In Sect. 2, a quick recap on the numerical homogenization for shell structures is given. While the current work aims on replacing the computationally expensive solution of the coupled problem, the consistent framework for extracting the effective properties from the microscale lays the foundation for the generation of the training data for the ANN. Sect. 3 defines the ANN architecture, while in Sect. 4 the generation of viscoelastic training data is discussed. Furthermore, investigations at the material point level for purely data-based training are done. In Sect. 5, additional information about the effective material matrix of the underlying microstructure is included by formulating appropriate constraints, leading to a more robust training process and reducing the amount of necessary data samples. Lastly, the feasibility of the surrogate model in a multiscale simulation is investigated in Sect. 6, including studies on the behavior for unknown loading paths and comparisons to classic full-scale as well as FE^2 solutions.

2 General framework for multiscale modeling of shell structures

2.1 Macroscopic shell formulation

The macroscopic boundary value problem is formulated within the three dimensional Euclidean space $\mathcal{B}$ occupied by a shell with reference surface Ω_0 and shell height h in the reference configuration. Ω_t refers to the reference surface in the current configuration due to loading conditions at time $t \in [0, T]$. A local convective coordinate system ξ^α is introduced, with $\xi^3 = z$ being the thickness direction. Stress boundary conditions $\bar{\mathbf{t}}$ are applied on Γ_σ and displacement boundary conditions on Γ_u, with $\Gamma = \Gamma_\sigma \cup \Gamma_u$ being the boundary of the reference surface Ω_0 and $\Gamma_\sigma \cap \Gamma_u = \emptyset$. The following kinematic assumptions are briefly derived according to [46]. With $\mathbf{X}$ and $\mathbf{x}$ referring to position vectors of material points in the reference and current configuration, the displacement field of the reference surface is introduced as

$$\mathbf{u} = \mathbf{x} - \mathbf{X}. \tag{1}$$

Taking into account rotational parameters $\boldsymbol{\omega}$ associated with the shell's local director field, the shell's degrees of freedom can be summarized in the vector $\mathbf{v}^T = [\mathbf{u}^T, \boldsymbol{\omega}^T]$. The general nonlinear Green-Lagrangian shell strains are arranged in the vector

$$\boldsymbol{\varepsilon}(\mathbf{v}) = [\varepsilon_{11}, \varepsilon_{22}, 2\varepsilon_{12}, \kappa_{11}, \kappa_{22}, 2\kappa_{12}, \gamma_1, \gamma_2]^T \tag{2}$$

with $\varepsilon_{\alpha\beta}$, $\kappa_{\alpha\beta}$ and γ_α referring to the membrane, bending and shear parts, respectively. Furthermore, we define the work conjugate 2. Piola-Kirchhoff shell stress resultants

$$\boldsymbol{\sigma} = [n_{11}, n_{22}, n_{12}, m_{11}, m_{22}, m_{12}, q_1, q_2]^T \tag{3}$$

and the material matrix

$$\mathbf{D} := \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{D}_m & \mathbf{D}_{mb} & \mathbf{0} \\ \mathbf{D}_{mb} & \mathbf{D}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_s \end{bmatrix} \tag{4}$$

without further assumptions about the constitutive relation. The indices m, b and s once again refer to the membrane, bending and shear parts. It is noted, that for the general anisotropic case, the material matrix can be fully populated, resulting in $\mathbf{D}_{ms}, \mathbf{D}_{bs} \neq \mathbf{0}$. The weak form of the nonlinear equilibrium equation reads

$$\begin{aligned} \delta\Pi^M(\mathbf{v}, \delta\mathbf{v}) &= \int_{\Omega_0} (\delta\boldsymbol{\varepsilon}^T \boldsymbol{\sigma} - \delta\mathbf{u}^T \bar{\mathbf{p}}) \mathrm{d}A - \int_{\Gamma_\sigma} \delta\mathbf{u}^T \bar{\mathbf{t}} \mathrm{d}s \\ &=: \mathbf{g}(\mathbf{v}, \delta\mathbf{v}) \overset{!}{=} 0, \end{aligned} \tag{5}$$

where the superscript $(\cdot)^M$ is introduced as a reference to the macroscopic problem, distinguishing from microscopic quantities, denoted with $(\cdot)^m$ in the following section, respectively.

2.2 Microscopic homogenization problem

In order to derive the macroscopic constitutive equation $\boldsymbol{\sigma} = f(\boldsymbol{\varepsilon})$, a second microscopic boundary value problem is introduced at each material point $\mathbf{X}$ of the macroscale with body $\mathcal{B}^m$ in the reference configuration, depicted in Fig. 1. Since the microstructure is assumed to be composed of different phases, $\boldsymbol{\sigma}$ can be seen as the average stress resultant and $\mathbf{D}$ as the effective material matrix, which one wants to compute from the micro problem. The microstructure is assumed to be periodic in the shell's in-plane direction and is described by a three-dimensional continuum, which will be referred to as a representative volume element (RVE). The RVE domain $\mathcal{B}^m$ is bounded within the range

$$-\frac{l_x}{2} \leq x \leq \frac{l_x}{2}, \quad -\frac{l_y}{2} \leq y \leq \frac{l_y}{2}, \quad h^- \leq z \leq h^+, \tag{6}$$

where x and y correspond to the in-plane coordinates and z to the thickness direction. If the midsurface is chosen as reference surface, it follows that $h^- = -h/2$ and $h^+ = +h/2$. The kinematics can be derived in a standard fashion considering basic continuum mechanic assumptions. Introducing a displacement field $\mathbf{u}^m$ on the microscale, the deformation gradient and Green-Lagrangian strains are defined as

$$\mathbf{F} = \mathbf{1} + \nabla_{\mathbf{X}^m} \mathbf{u}^m \quad \text{and} \quad \mathbf{E} = \frac{1}{2}(\mathbf{F}^\mathbf{T}\mathbf{F} - \mathbf{1}) \tag{7}$$

respectively, with ∇ denoting partial differentiation and $\mathbf{X}^m$ being the position vector of points in the macroscopic reference configuration. By taking advantage of the vector notation for the strains as well as for the 2. Piola Kirchhoff Stress $\mathbf{S}$

$$\begin{aligned} \mathbf{E} &= [E_{11}, E_{22}, E_{33}, 2E_{12}, 2E_{13}, 2E_{23}]^T, \\ \mathbf{S} &= [S_{11}, S_{22}, S_{33}, S_{12}, S_{13}, S_{23}]^T \end{aligned} \tag{8}$$

and in the absence of body forces, which are defined on the macroscale, the microscopic virtual work is

$$\delta\Pi^m(\mathbf{u}^m, \delta\mathbf{u}^m) = \int_{\mathcal{B}^m} \delta\mathbf{E}^T \mathbf{S} \, \mathrm{d}V. \tag{9}$$

Without further specification, we assume that the constitutive relation of different phases of the microstructure can be characterized by the definition

$$\mathbf{S} := \mathbf{S}(\mathbf{E}, \boldsymbol{\xi}), \quad \mathbf{C} := \frac{\mathrm{d}\mathbf{S}(\mathbf{E}, \boldsymbol{\xi})}{\mathrm{d}\mathbf{E}} \tag{10}$$

with $\boldsymbol{\xi}$ being a viscoelastic internal variable depending on $\mathbf{E}$, and $\mathbf{C}$ the material tangent.

As mentioned before, the aim of the microscopic problem is to derive the macroscopic quantities $\boldsymbol{\sigma}$ and $\mathbf{D}$ for given shell strains $\boldsymbol{\varepsilon}$. To do so, appropriate boundary conditions have to be applied for solving the microscopic problem. Therefore, the Hill-Mandel condition [16], which demands the equiva-

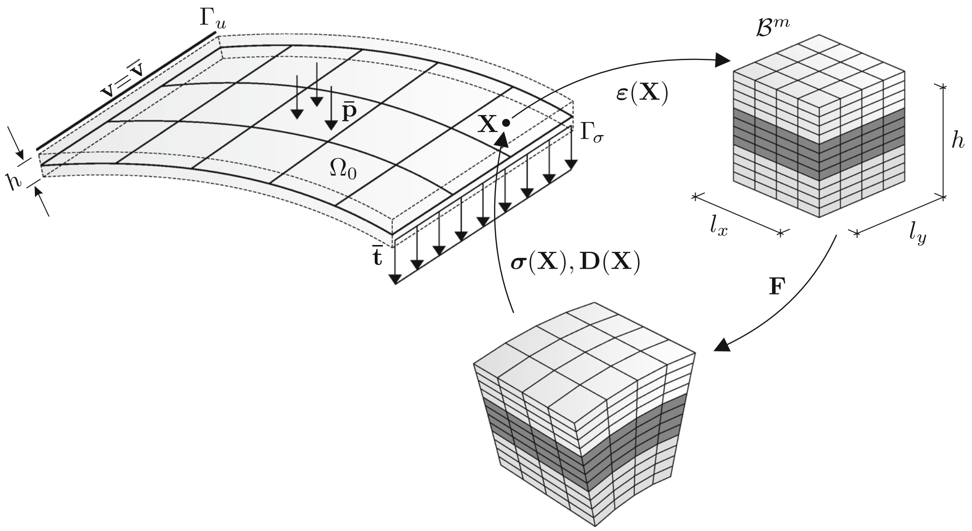


Fig. 1 Macroscopic boundary value problem and schematic homogenization procedure at each material point for shell RVEs

lence of virtual work on both scales, is introduced

$$\frac{1}{h}\delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma} = \frac{1}{V_0}\int\limits_{\mathcal{B}^m}\delta\mathbf{E}^T\mathbf{S}\,\mathrm{d}V\,. \tag{11}$$

$V_0 = l_x l_y h$ denotes the undeformed volume of the RVE, resulting in

$$\delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma} = \frac{1}{A_0}\int\limits_{\mathcal{B}^m}\delta\mathbf{E}^T\mathbf{S}\,\mathrm{d}V\,. \tag{12}$$

Eq. (12) can be seen as the adaptation of the general form of the Hill-Mandel condition for, in this case, the coupling of shells and continua proposed by [4], where the average only has to be taken over the undeformed reference area of the RVE $A_0 = l_x l_y$. Furthermore, it is written in terms of the 2. Piola-Kirchhoff stresses and Green-Lagrange strains in accordance with [10], which is necessary in the context of small strains but finite displacements. From Eq. (12), three types of admissible conditions on the boundary of the RVE $\partial\mathcal{B}^m$ can be extracted, see e.g. [34, 40]; linear displacement, uniform traction and periodic boundary conditions. While the former two yield upper and lower bounds of the effective material properties, the latter is known to be well suited for obtaining good approximations, even for non-periodic microstructures. In the following, periodic boundary conditions are briefly discussed with special focus on the difficulties that arise in the framework for shells.

We denote two opposing lateral surfaces of the RVE, corresponding to the surfaces perpendicular to the xy-plane, with $(\cdot)^+$ and $(\cdot)^-$, respectively. While the top and bottom surfaces have to remain free of traction according to the equilibrium assumptions made on the macroscale, the periodic displacements on the lateral surfaces are defined via

$$\mathbf{u}^+ = \mathbf{u}^- + \bar{\mathbf{E}}\,\Delta\mathbf{X}^m, \quad \Delta\mathbf{X}^m = \mathbf{X}^+ - \mathbf{X}^-, \tag{13}$$

with $\bar{\mathbf{E}}$ being the average strain tensor [11]. Considering the relations

$$\begin{aligned} \bar{E}_{11} &= \varepsilon_{11} + z\kappa_{11} \\ \bar{E}_{22} &= \varepsilon_{22} + z\kappa_{22} \\ \bar{E}_{33} &= 0 \\ \bar{E}_{12} &= \bar{E}_{21} = \varepsilon_{12} + z\kappa_{12} \\ 2\bar{E}_{13} &= 2\bar{E}_{31} = \gamma_1 \\ 2\bar{E}_{23} &= 2\bar{E}_{32} = \gamma_2 \end{aligned} \tag{14}$$

Eq. (13) can be expressed with the vector of shell strains

$$\mathbf{u}^+ = \mathbf{u}^- + \mathbf{A}(\Delta x, \Delta y, z)\,\boldsymbol{\varepsilon} \tag{15}$$

with

$$\mathbf{A} = \begin{bmatrix} \Delta x & 0 & \frac{1}{2}\Delta y & \Delta x\,z & 0 & \frac{1}{2}\Delta y\,z & 0 & 0 \\ 0 & \Delta y & \frac{1}{2}\Delta x & 0 & \Delta y\,z & \frac{1}{2}\Delta x\,z & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \Delta x & \Delta y \end{bmatrix}. \tag{16}$$

The relationship $\mathbf{u}^+ = \mathbf{u}^-$ is enforced with a master slave approach by defining appropriate constraints, whereas in-plane displacements are linked in a symmetric and out of plane displacements in an anti-symmetric matter regarding the in-plane coordinates x and y.

An issue that arises when dealing with periodic boundary conditions for shell RVEs is the length dependency of the homogenized shear stiffness, regarding the choice of l_x and l_y. This is due to the fact, that imposing a pure shear strain γ_α on the RVE does not result in pure shear stresses, but also yields an undesired linear momentum distribution.

An in-depth discussion for beams can be found in [21], where furthermore an approach is presented via a definition of constraints, enforcing vanishing linear bending moment distributions in an integral sense. We here take advantage of the method proposed in [11], where the constraints are formulated for shell structures. For example, in order to enforce a constant bending moment m_x on the RVE, the constraint reads

$$\int_x \int_y m_x \kappa_x \mathrm{d}y \mathrm{d}x = 0\,, \tag{17}$$

which, assuming a linear curvature κ_x, only vanishes if $m_x = \text{const.}$ throughout the reference surface of the RVE. The constraints, which are enforced by a Lagrange multiplier approach, circumvent the remedy of the length dependency of the effective shear stiffness and prevent pure rigid body deformation when imposing a pure shear strain. Within a numerical treatment, this leads to additional degrees of freedom on the microscale, the Lagrange parameters, which are summarized in the vector $\boldsymbol{\lambda}$. In the following sections, periodic boundary conditions with the additional constraints will be used on the RVE.

2.3 Numerical homogenization for shell RVEs

Having formulated the necessary equations for solving the boundary value problem, the aim is to extract the average stress field and effective material tangent from the micro problem numerically, following the approach in [11], which is referred to for details, e.g. on the numerical treatment of the Lagrange parameters. Taking advantage of the finite element method, the microscale is discretized into finite domains

$$\mathcal{B}^m \approx \mathcal{B}^h = \bigcup_{e=1}^{N} \mathcal{B}_e\,, \tag{18}$$

with N being the maximum number of elements used on the microscale. A standard displacement-based formulation, with 27-, or 64-node solid elements is used for the interpolation of the displacement field. We once again take advantage of Eq. (12) by linearizing both sides of the equation

$$\begin{aligned}&\delta\boldsymbol{\varepsilon}^T(\mathbf{D}\Delta\boldsymbol{\varepsilon}+\boldsymbol{\sigma}) = \\ &\quad \frac{1}{A_0}\int_{\mathcal{B}^m} \delta\mathbf{E}^T\mathbf{C}\Delta\mathbf{E} + \Delta\delta\mathbf{E}^T\mathbf{S} + \delta\mathbf{E}^T\mathbf{S}\,\mathrm{d}V\,.\end{aligned} \tag{19}$$

Inserting the ansatz for the displacements and virtual displacements into the right-hand side of Eq. (19) yields

$$\delta\boldsymbol{\varepsilon}^T(\mathbf{D}\Delta\boldsymbol{\varepsilon}+\boldsymbol{\sigma}) = \frac{1}{A_0}\sum_{e=1}^{N}\delta\mathbf{v}_e^T(\mathbf{k}_e\Delta\mathbf{v}_e+\mathbf{f}_e)\,, \tag{20}$$

with $\mathbf{v}_e$ being the microscopic element nodal displacement vector, $\mathbf{k}_e$ the element tangent stiffness matrix, and $\mathbf{f}_e$ the element residual vector. $\mathbf{V}$, $\mathbf{K}$, and $\mathbf{F}$ will later on denote their assembled global counterparts, respectively. In the following, the aim is to reformulate the right hand side of Eq. (20) such that the desired macroscopic quantities $\boldsymbol{\sigma}$ and $\mathbf{D}$ can be determined by comparison of coefficients. Therefore, the element displacement vector is divided into contributions resulting from the prescribed shell strains of the macroscale $\boldsymbol{\varepsilon}$ and the generalized displacement vector $\mathbf{V}$

$$\mathbf{v}_e = \mathbf{A}_e\boldsymbol{\varepsilon} + \mathbf{a}_e\mathbf{V}\,. \tag{21}$$

Hereby, $\mathbf{a}_e$ denotes a standard assembly matrix and $\mathbf{A}_e$ is specified in [11], based on Eq. (16). For elements which are not located on the boundary of the RVE, the contribution $\mathbf{A}_e\boldsymbol{\varepsilon}$ does not exist. Analogously, the linearized and virtual quantities follow as

$$\delta\mathbf{v}_e = [\mathbf{A}_e, \mathbf{a}_e]\begin{bmatrix}\delta\boldsymbol{\varepsilon}\\ \delta\mathbf{V}\end{bmatrix}, \quad \Delta\mathbf{v}_e = [\mathbf{A}_e, \mathbf{a}_e]\begin{bmatrix}\Delta\boldsymbol{\varepsilon}\\ \Delta\mathbf{V}\end{bmatrix}. \tag{22}$$

Inserting the relations (22) into the right hand side of Eq. (20) yields

$$\begin{aligned}&\delta\boldsymbol{\varepsilon}^T(\mathbf{D}\Delta\boldsymbol{\varepsilon}+\boldsymbol{\sigma}) = \\ &\frac{1}{A_0}\begin{bmatrix}\delta\mathbf{V}\\ \delta\boldsymbol{\varepsilon}\end{bmatrix}^T\left\{\begin{bmatrix}\mathbf{K}_{11} & \mathbf{K}_{12}\\ \mathbf{K}_{21} & \mathbf{K}_{22}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{V}\\ \Delta\boldsymbol{\varepsilon}\end{bmatrix}\begin{bmatrix}\mathbf{F}_1\\ \mathbf{F}_2\end{bmatrix}\right\},\end{aligned} \tag{23}$$

with the modified submatrices

$$\begin{aligned}\mathbf{K}_{11} &= \sum_{e=1}^{N}\mathbf{a}_e^T\mathbf{k}_e\mathbf{a}_e\,, \quad \mathbf{K}_{12} = \mathbf{K}_{21}^T,\\ \mathbf{K}_{21} &= \sum_{e=1}^{N}\mathbf{a}_e^T\mathbf{k}_e\mathbf{A}_e\,, \quad \mathbf{K}_{22} = \sum_{e=1}^{N}\mathbf{A}_e^T\mathbf{k}_e\mathbf{A}_e\end{aligned} \tag{24}$$

and residuals

$$\mathbf{F}_1 = \sum_{e=1}^{N}\mathbf{a}_e^T\mathbf{f}_e\,, \quad \mathbf{F}_2 = \sum_{e=1}^{N}\mathbf{A}_e^T\mathbf{f}_e\,. \tag{25}$$

Through static condensation, $\Delta\mathbf{V}$ can be eliminated from the system of equations using

$$\Delta\mathbf{V} = -\mathbf{K}_{11}^{-1}(\mathbf{F}_1 + \mathbf{K}_{12}\Delta\boldsymbol{\varepsilon})\,, \tag{26}$$

which reduces Eq. (23) to

$$\begin{aligned}\delta\boldsymbol{\varepsilon}^T(\mathbf{D}\Delta\boldsymbol{\varepsilon}+\boldsymbol{\sigma}) = \frac{1}{A_0}\Big[&(\mathbf{K}_{22} - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{K}_{12})\,\Delta\boldsymbol{\varepsilon}\\ &+\mathbf{F}_2 - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{F}_1\Big].\end{aligned} \tag{27}$$

From Eq. (27), one can extract the averaged stress resultants and effective material matrix by comparison of coefficients from the microscale

$$
\begin{aligned}
\boldsymbol{\sigma} &= \frac{1}{A_0}(\mathbf{F}_2 - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{F}_1),\\
\mathbf{D} &= \frac{1}{A_0}(\mathbf{K}_{22} - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{K}_{12})\,.
\end{aligned}
\tag{28}
$$

Within a multiscale framework, often referred to as FE^2, the above derived homogenization process is usually performed at each macroscopic integration point, leading to a coupled boundary value problem. The shell strains are necessary for solving the microscopic boundary value problem yielding the effective quantities, which then enter the macroscopic element tangent stiffness matrix and residual. Since the micro problems are decoupled, they can be solved individually in parallel. The whole system is solved iteratively with a Newton scheme, either simultaneously or within a nested iteration [10].

In the linear case, only a single calculation has to be done on one RVE within a preprocessing, yielding the effective material matrix $\mathbf{D}$, with which one can calculate the averaged stress resultants $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon}$, without having to solve the micro problem again. Nevertheless, the method encounters its bottleneck as soon as path dependent material laws, e.g., elastoplasticity or viscoelasticity, are considered. Depending on the complexity of the microstructure, the computational effort for solving the micro problems at each integration point usually outweighs the numerical advantages of shell elements when compared to a classic single scale approach using standard solid elements.

In the following section, an alternative approach is therefore introduced, combining the advantages of shell elements with a feasible way of obtaining the homogenized quantities for viscoelastic material behavior by using an artificial neural network.

3 Feed forward neural network model for viscoelastic materials

In order to surpass the time-consuming process of solving the microscopic boundary value problem at each macroscopic integration point, the next section is dedicated to the derivation of a suitable surrogate model. Here, an artificial neural network will be used for the approximation of the effective viscoelastic material behavior of the multi-phased microstructure. The key idea is to derive a mapping

$$
f(\mathbf{x}) : \mathbf{x} \mapsto \boldsymbol{\sigma}\,, \tag{29}
$$

where one obtains the averaged shell stress resultants from a given vector $\mathbf{x}$, which needs to contain the necessary information for a distinct mapping of $f(\mathbf{x})$. While Sect. 3.1 and 3.2 focus on general ANN formalism concerning mapping and the training process, Sect. 3.3 will focus on the application to viscoelastic material modeling. All of the following formalisms have been implemented in Matlab [43].

3.1 General ANN definitions and mapping

We define an ANN as a mapping

$$
f(\mathbf{x};\mathbf{w}) : \mathbf{x} \mapsto \mathbf{z}\,, \tag{30}
$$

with the input vector $\mathbf{x} \in \mathbb{R}^{n_i}$, the output vector $\mathbf{z} \in \mathbb{R}^m$ and the vector $\mathbf{w} \in \mathbb{R}^{n_w}$ containing the weights, i.e., the free parameters of the function. n_i and m refers to the maximum number of input and output variables, respectively, and n_w denotes the maximum number of weights in the network. As ANN type, a standard feed forward neural network is used. For details on the mapping process, see appendix A in [48]. The goal is to determine a set of weights, such that a predefined error function $E(\mathbf{x}_k;\mathbf{w})$ is minimized:

$$
\min \quad E(\mathbf{x}_k;\mathbf{w}) \quad \text{w.r.t.} \quad \mathbf{w}\,, \tag{31}
$$

where $\mathbf{x}_k$ denotes the input vector of a given data set containing $k = 1,\ldots,P$ data samples taken from a sampling space Ω. The corresponding target values of the data set for a given input vector $\mathbf{x}_k$ is referred to as $\mathbf{t}_k$. The generation of data samples will be discussed in Sect. 4. As the objective error function, we define

$$
E(\mathbf{x}_k;\mathbf{w}) = \frac{1}{2P}\sum_{k=1}^{P}\sum_{j=1}^{m}(z_{kj}(\mathbf{x}_k;\mathbf{w}) - t_{kj})^2\,, \tag{32}
$$

where one aims to minimize the difference between the output of the neural network and the target value from a given data set in a mean squared sense.

3.2 ANN training process

As Eq. (31) poses a classic minimization problem, one can resort to a wide range of first and second order methods, which deal with the issue of finding the minima of a function defined in a high dimensional space. The probably most renown first order method is the gradient descent, which updates the weights of the ANN in each iteration step p by the rule

$$
\mathbf{w}_{p+1} = \mathbf{w}_p - \eta\,\mathbf{g}\,. \tag{33}
$$

Hereby

$$\mathbf{g} := \nabla_{\mathbf{w}} E(\mathbf{w}) \tag{34}$$

denotes the gradient of the error function $E(\mathbf{w})$, which is calculated via the backpropagation method [39], and η a is scalar valued factor controlling the step size within the update scheme, oftentimes also referred to as learning rate. An improved version of the steepest descent algorithm is the Adam optimizer [20], which is suited for networks with a large number of degrees of freedom, so called deep neural networks. On the contrary, for smaller to moderate sized networks, second order schemes such as the Levenberg-Marquardt method [12] have shown to be more efficient, following the update rule

$$\mathbf{w}_{p+1} = \mathbf{w}_p + (\mathbf{H} + \mu \mathbf{I})^{-1}\, \mathbf{g}\ . \tag{35}$$

Here, μ is a damping parameter and

$$\mathbf{H} := \nabla^2_{\mathbf{w}} E(\mathbf{w}) \tag{36}$$

denotes the Hessian of the error function, which within the method is approximated by using only first order derivatives, see [30] and [31]. Regardless of the choice of the optimization scheme, the finding of a satisfying minimum is referred to as neural network training. Therefore, the data set P is usually split into two subsets

$$P = P^{\text{Train}} \cup P^{\text{Test}}, \quad P^{\text{Train}} \cap P^{\text{Test}} = \emptyset\,, \tag{37}$$

where P^{Train} is taken for solving the optimization problem and P^{Test} is used for monitoring the generalization abilities of the network when it is applied to unknown data. Equivalently, the error function given by Eq. (32) can be defined with respect to the training E^{Train} and testing E^{Test} samples, respectively. The calculation of the error function for the training set and the computation of the weight update is referred to as a training epoch. In general, neural network training consists of multiple epochs until a user defined termination criterion is reached. Later, either a number of maximum training epochs, or if the average change of the normalized training error over five epochs is smaller than 10^{-6}, will be specified as termination criterions.

3.3 ANN details for viscoelastic material models

The key feature in deriving a material model when using neural networks in general is the definition of suitable input and output vectors. In case of path dependency, appropriate internal variables have to be identified in order to approximate the material behavior, since a network architecture without a priori defined recurrence is used. For an in-depth discussion of input variables for rate independent plasticity, especially concerning history variables, the reader is referred to [49, 50].

Similarly to classic rheological models, it is crucial to identify the underlying effects when observing the material specific deformation behavior and to define the input variables accordingly. For material modeling, we consider ANN input vectors of the generic type

$$\mathbf{x} = \begin{bmatrix} \hat{\boldsymbol{\varepsilon}} \\ \mathbf{h} \\ T \\ \mathbf{p} \end{bmatrix} \tag{38}$$

with the generalized input variables:

- $\hat{\boldsymbol{\varepsilon}}$ refers to strain type variables, e.g., $\boldsymbol{\varepsilon}_{n+1}$, $\boldsymbol{\varepsilon}_n$ or the current strain increment $\Delta\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n$, where n is the previous and $n+1$ the current time-, or loading-step, respectively.
- $\mathbf{h}$ is the vector containing history variables, e.g., the last stress state $\boldsymbol{\sigma}_n$ or the maximum strain norm reached $||\boldsymbol{\varepsilon}||_{max}$.
- T are scalar valued time variables, e.g., the current time t_{n+1} or time step $\Delta t_{n+1} = t_{n+1} - t_n$.
- $\mathbf{p}$ are other, non-strain related parameters of the RVE, e.g., geometric features or material parameters of a certain phase.

While we assume that the variables $\hat{\boldsymbol{\varepsilon}}$, $\mathbf{h}$ and T are sufficient, if defined properly, to capture the effective material behavior, the convenience when using ANNs as a surrogate model within a multiscale scheme is clearly displayed when it comes to the non-strain parameters $\mathbf{p}$. Here, one could define arbitrary features of the RVE, e.g., the volume fraction of a phase in an RVE, the size of an inclusion or the height of a certain layer as additional input parameters. This, of course, also increases the dimensionality and complexity of the material function to be approximated, necessitating larger ANN architectures and additional training data.

As viscoelastic materials exhibit path and strain rate dependency, the variables $\hat{\boldsymbol{\varepsilon}}$, $\mathbf{h}$ and T always have to be considered, while $\mathbf{p}$ is optional. We therefore define the input vector of the ANN

$$\mathbf{x} := \begin{bmatrix} \Delta\boldsymbol{\varepsilon}_{n+1} \\ \boldsymbol{\varepsilon}_n \\ \boldsymbol{\sigma}_n \\ \Delta t_{n+1} \end{bmatrix}, \tag{39}$$

where the vector of shell stress resultants $\boldsymbol{\sigma}_n$ of the last equilibrium point is chosen as history variable. Other types of input variable combinations have been studied with the con-

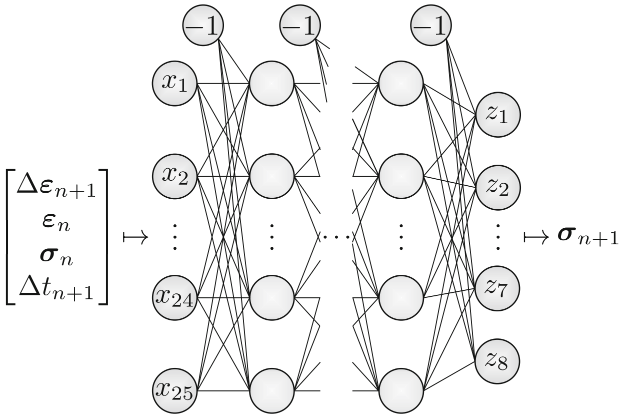


Fig. 2 Graphical illustration of the fully connected feed forward neural network with input and output variables used for modeling viscoelastic material behavior

clusion that Eq. (39) yields the best results. In the general three-dimensional case for shell elements and without additional parameters $\mathbf{p}$, the input vector consists of 25 input variables, considering the definitions (2) and (3).

The output vector on the other hand is straightforward, as we define the current stress as the desired output of the neural network, which consists of eight components

$$\mathbf{z} = \boldsymbol{\sigma}_{n+1} \,. \tag{40}$$

A possible alternative approach would be the definition of the current stress increment as output, which will not be further discussed here. Consequently, the ANN material matrix $\mathbf{D}^{ANN}$ can be derived by

$$\mathbf{D}^{ANN} = \frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}} = \frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\Delta\boldsymbol{\varepsilon}_{n+1}} \underbrace{\frac{\partial\Delta\boldsymbol{\varepsilon}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}}}_{\mathbf{1}} = \frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\Delta\boldsymbol{\varepsilon}_{n+1}} \,, \tag{41}$$

which requires the computation of derivative components $\partial z_j/\partial x_i$, see appendix B of [48]. Having defined the mapping, in the next section the generation of data samples for the training of the ANN is discussed.

4 Viscoelastic ANN training

Besides having to choose suitable input and output variables and an optimal topology of neurons in between, another essential feature for constructing a neural network material model is having an appropriate data set, containing the necessary information for the ANN to extract the main features characterizing the material behavior. When dealing with experimental data, this can pose a serious drawback, as data is often limited and multiaxial material tests are mostly difficult to realize. For a discussion on these issues and possible solutions in the context of ANN material modeling, the reader is referred to [48] and [49]. In the context of multiscale modeling though, unlimited data is available, since the material tests are performed synthetically within a numerical framework.[1] Nevertheless, for the sake of feasibility, the aim is to generate as little samples as possible, but as much as necessary.

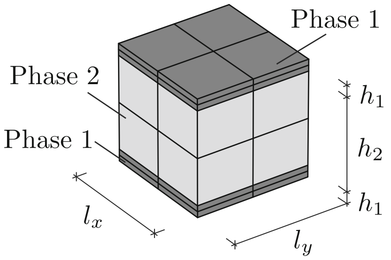


Fig. 3 Layered shell RVE with phase 1 (elastic material) and phase 2 (viscoelastic material)

The general procedure for developing an ANN material model for the effective constitutive behavior of a shell RVE with viscoelastic properties will be discussed in the following for a layered RVE consisting of two phases as shown in Fig. 3. It can, of course, be applied to arbitrary microstructures as will be shown in Sect. 6. Without further specification, for now, the RVE consists of two phases, an elastic bottom and top layer referred to as Phase 1 with associated height h_1, and a viscoelastic core labeled Phase 2 and height h_2, respectively. We assume the material behavior for both phases to be linear and isotropic. As one of the phases inhibits viscoelastic properties, the overall effective material behavior will be time and history dependent as well.

The following steps have to be undertaken for the exemplary shell RVE in order to develop a surrogate model for the effective material behavior:

1. Generate explicit strain (-stress) time paths for identifying the viscoelastic properties,
2. Choose appropriate sampling space limits for the relevant input variables,
3. Perform synthetic material tests on the RVE to obtain the effective quantities and transform samples according to the definition of the ANN input vector,
4. Train- and validate the ANN material model.

---

[1] On a side note, the whole viscoelastic ANN framework described within this work can of course also be applied to experimental data.

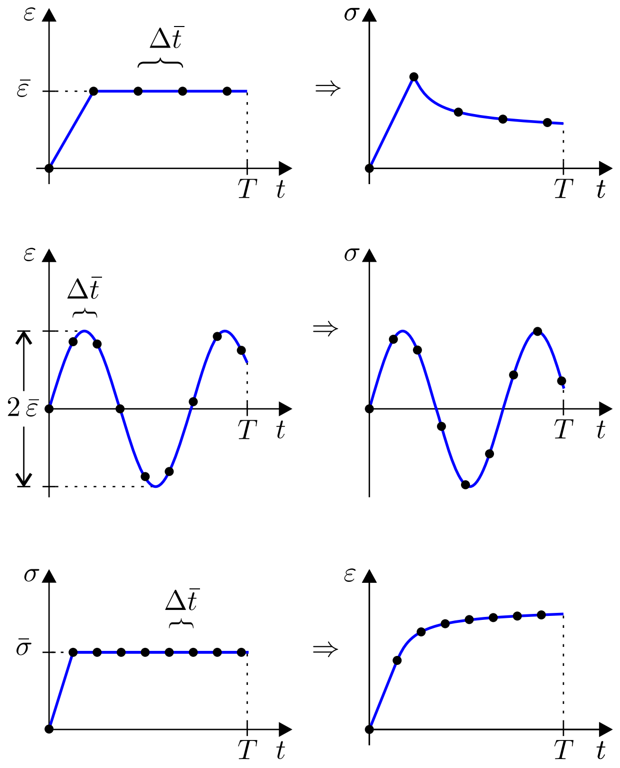


Fig. 4 Explicit time paths for viscoelastic ANN sampling (from top to bottom): relaxation test, cyclic loading and creep test

4.1 Generation of explicit time paths

In contrast to classical sampling approaches for feed forward ANNs, it would not be feasible to take the input variables in the defined form (39), define boundaries of the input space Ω and sample, e.g. randomly, within the space. As viscoelastic materials exhibit path dependency, it is necessary to generate the samples from a given path. Therefore, explicit strain-time paths

$$\{\boldsymbol{\varepsilon}_0, t_0\} \rightarrow \{\boldsymbol{\varepsilon}_1, t_1\} \rightarrow \ldots \rightarrow \{\boldsymbol{\varepsilon}_n, t_n\} \rightarrow \ldots \rightarrow \{\boldsymbol{\varepsilon}_T, t_T\} \tag{42}$$

or, alternatively in the case of stress-controlled sampling, stress-time paths

$$\{\boldsymbol{\sigma}_0, t_0\} \rightarrow \{\boldsymbol{\sigma}_1, t_1\} \rightarrow \ldots \rightarrow \{\boldsymbol{\sigma}_n, t_n\} \rightarrow \ldots \rightarrow \{\boldsymbol{\sigma}_T, t_T\} \tag{43}$$

are defined. The basic solution, being the effective shell stress resultant in the case of Eq. (42) and the average shell strains in the case of Eq. (43), is calculated on the RVE as described in Sect. 2.3 and the samples are transformed to the incremental form (39), including the history variables. As explicit time paths, we consider three different types of paths, depicted for the one-dimensional case in Fig. 4.

The paths are chosen in accordance with loading conditions that are known from experimental setups to identify the viscoelastic properties. The paths are:

- Relaxation test: $\varepsilon(t > 0) = \bar{\varepsilon}$
- Cyclic loading: $\varepsilon(t) = \bar{\varepsilon}\sin(\Omega t)$
- Creep test: $\sigma(t > 0) = \bar{\sigma}$

The paths can either be described as uniaxial tests on the RVE, to which we will restrict the training paths, but can of course also be done multiaxially, which is important for the test data, as the ANN is supposed to be applied as a general 3D material model. The synthetic material tests will be described in detail in Sect. 4.3.

From the neural network perspective, the relaxation test provides the important information that zero strain rates $\dot{\varepsilon} = \Delta\varepsilon/\Delta t = 0$ can result in a change of stress over time $\dot{\sigma} = \Delta\sigma/\Delta t \neq 0$. The cyclic loading is suited for covering a wide range of strain increments, whereas the relaxation test only contains a single strain increment unequal zero, if the loading is done within a single time step. Lastly, the creep test yields the input combination, for which the network is supposed to produce a constant stress response.

4.2 Choice of appropriate sampling space limits

For each of the previously defined paths, the variables noted with $\bar{\varepsilon}$ and $\bar{\sigma}$, as well as the test duration T, must be prescribed. While the former two are input variables of the ANN, the latter is not, but the time increment Δt used to discretize the path is. Starting with the sampling space limits for the strains, generally the assumption

$$||\mathbf{E}|| < 0.03 \tag{44}$$

holds for small strains. A specialty for the case when shell strains are used as input, is the fact that the curvatures $\kappa_{\alpha\beta}$ depend on the geometry of the RVE. Keeping in mind the relation from Eq. (14) and furthermore only using uniaxial training paths, one can restrict the curvatures to

$$|\kappa_{\alpha\beta}| < \frac{0.03}{z_{max}}\,, \tag{45}$$

where $z_{max} = h/2$ depends on the height of the RVE. For the lower and upper limits of the time increments, $\Delta\bar{t}_{min}$ and $\Delta\bar{t}_{max}$, it is on the other hand more difficult to determine suitable boundaries. As the feed forward ANN usually does not inhibit any reasonable extrapolation capabilities, the surrogate model will only yield results for time increments chosen within the training space. Therefore, currently

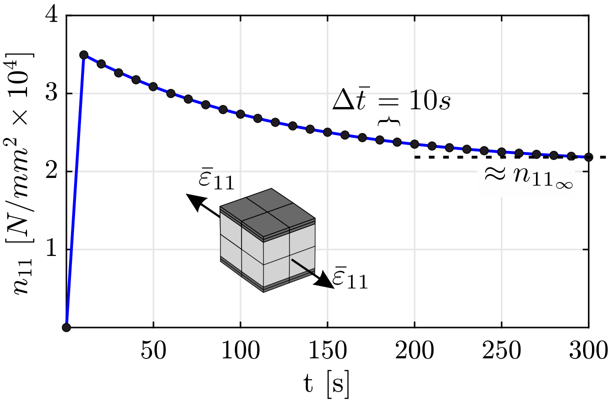


Fig. 5 Exemplary relaxation response of a layered RVE to a macroscopically prescribed shell strain and adequate choice of equidistant time steps until the equilibrium stress is reached

the relaxation behavior of one uniaxial relaxation test, e.g. prescribed ε_{11}, on the RVE is observed, until a converged solution with respect to the resulting equilibrium shell stress resultant, here n_{11_∞}, is reached. The time needed for reaching the equilibrium stress can afterwards be discretized into adequate time steps and subsequently, upper and lower bounds are defined. It is noted, that the efficiency of the sampling strategy could be further improved by adaptively choosing the time step size, e.g., by using larger time steps within regions of the time domain, where a lower resolution of the signal is sufficient.

As exemplary depicted in Fig. 5, for a prescribed constant strain $\bar{\varepsilon}_{11}$ the equilibrium stress is reached after $\approx\ 300\,s$. Consequently, within numerical simulations where the material model will be later used, a suitable time step could be $\Delta\bar{t} = 10\,s$ and the boundaries could be set to $\Delta\bar{t} \in [5s, 15s]$. Furthermore, the period for reaching the equilibrium stress can also be used to define the duration of the test T. Likewise, one needs to define appropriate stresses $\bar{\sigma}$ for the creep tests, ensuring the resulting strains remain within the limits defined in Eq. (44).

As it would not be feasible to prescribe each strain path individually, the paths are defined in a bunch. For example, a bunch n_{paths} of uniaxial relaxation tests with a prescribed membrane strain component $\bar{\varepsilon}_{11}$ is defined as

$$n_{paths},\ \Delta\bar{t}_{min},\ \Delta\bar{t}_{max},\ T_{min},\ T_{max},\ \varepsilon_{min},\ \varepsilon_{min},\ \bar{\varepsilon}_{11} \tag{46}$$

where the limits of the values denoted with $(\ldots)_{min}$ and $(\ldots)_{max}$ are chosen as discussed previously. In between the limits, the values are sampled via the Latin hypercube method.

4.3 Synthetic material tests and sample transformation

After having discussed the choice of explicit time paths, the remaining open question is in what way to prescribe the strain components. For example, one could prescribe one strain component and set the others to zero. This on the other hand would not cover a wide range of different input combinations. Therefore, in analogy to experimental testing, the paths are applied via synthetic material tests, meaning uni-, or multiaxial tension and compression tests, where within a Newton iteration scheme, the nonzero strain components are calculated. Since we currently limit ourselves to linear constitutive laws on the RVE, the solution within the Newton iteration is always acquired within one iteration step. The loading can either be done strain-controlled for the relaxation test and cyclic loading, or stress controlled for creep tests. For the exemplary shell RVE, this procedure is depicted in Fig. 6 for uniaxial strain-controlled sampling. As the RVE is symmetrically layered with the midsurface as chosen reference surface, the effective material matrix consequently reads

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_s \end{bmatrix}, \tag{47}$$

indicating that there is no coupling between the membrane and bending components. The prescribed strain components yield reaction forces unequal zero in the corresponding direction, whereas the other strain components are calculated from the assumption that the remaining reaction forces vanish.

Having described all necessary steps to generate the samples, lastly the samples have to be transformed to the incremental form (39). After each converged iteration step, the strain and time increments, as well as the ANN history variables, are calculated.

4.4 Application to a viscoelastic RVE

4.4.1 Geometric and material parameters of the RVE

In the following, the previously described sampling procedure is applied to the layered shell RVE to train a neural network surrogate model. For the numerical homogenization procedure, each layer is discretized with two 27-node solid elements based on a displacement formulation. Periodic boundary conditions are applied for imposing the average shell strain field as described in Sect. 2.2, and five Lagrange parameters are defined as additional degrees of freedom for each element. For further details on the number of Lagrange parameters used, the reader is referred to [11]. As mentioned, the top and bottom layer are chosen to be linear elastic while the core is described with a linear viscoelastic constitutive

Table 1 Material parameters for each phase and geometry of the layered shell RVE

Phase i	Initial modulus $E_0\ [N/mm^2]$	Poisson ratio $\nu_0\ [-]$	Shear parameter $\mu_1\ [-]$	Relaxation time $\tau_1\ [s]$	Height $h_i\ [mm]$
1	6850	0.2	–	–	2
2	2600	0.3	0.9	33.33	16

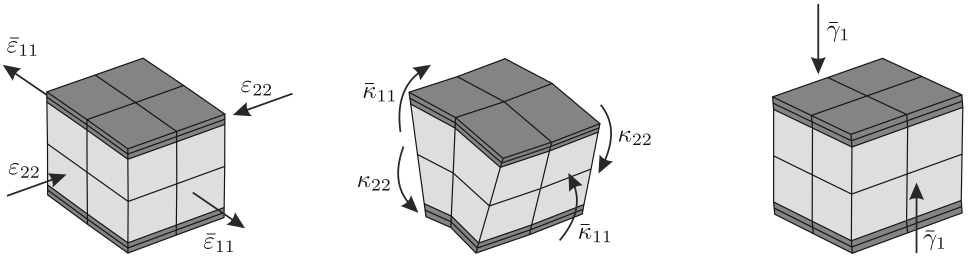


Fig. 6 Uniaxial synthetic material testing of a layered RVE; one shell strain component denoted with $(\bar{.})$ is prescribed, whereas the remaining components, which depend on the underlying microstructure, are calculated from the uniaxial equilibrium assumption

Table 2 Overview of ANNs and their underlying training data

Name	Topology	No. of uniaxial paths per strain component			Resulting
		Relaxation tests	Cyclic loading	Creep tests	Data points
ANN_5	25-60-8	5	5	5	2601
ANN_{10}	25-60-60-8	10	10	10	5415
ANN_{20}	25-80-80-8	20	20	20	10682

relation. The geometrical and material parameters of the two phases are given in Table 1. For modeling the viscoelastic phase, a generalized three-dimensional Maxwell model with

$$E_0 = E_\infty + \sum_{n}^{M} E_n, \quad \tau_n = \frac{\eta_n}{E_n} \tag{48}$$

is used, see Fig. 7. M denotes the number of Maxwell elements connected in parallel. In the following example, a simple model with only one Maxwell element will be used, i.e., $M = 1$. For details on the theoretical background and numerical implementation, the reader is referred to [17]. In accordance with observations made for many materials, we assume that only the deviatoric strain components yield a viscoelastic response, while the volumetric strain components are fully elastic. Therefore, the normalized function, with

$$G_0 = \frac{E_0}{2(1+\nu_0)}\,, \tag{49}$$

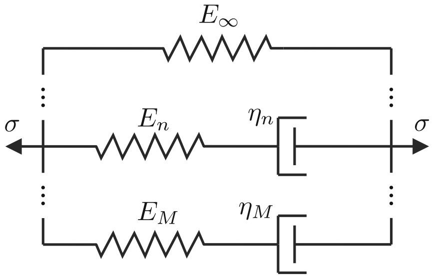


Fig. 7 Schematic illustration of the generalized Maxwell model: one spring element in parallel with an arbitrary number of Maxwell elements

is written in terms of the normalized shear parameters μ_n of each Maxwell element

$$\mu_n = \frac{G_n}{G_0}, \quad n = 1, \ldots, M\,. \tag{50}$$

Table 3 Overview of generated test paths on the RVE; for the sake of feasibility units are neglected

No	Test type	Test method	Prescribed component	Component values	Δt	T
1	Cyclic	Uniaxial	$\bar{\varepsilon}_{11}$	0.01	8	300
2	Creep	Uniaxial	$\bar{m}_{11}$	100	8	300
3	Relaxation	Uniaxial	$\bar{q}_1$	0.01	8	300
4	Cyclic	Biaxial	$\bar{\varepsilon}_{11}$,$\bar{\varepsilon}_{22}$	0.01, 0.01	8	300
5	Creep	Biaxial	$\bar{m}_{11}$,$\bar{m}_{22}$	100, 100	8	300
6	Relaxation	Biaxial	$\bar{q}_1$, $\bar{q}_2$	0.01, 0.01	8	300

4.4.2 Training of the ANN surrogate model

To generate training data for the ANN, only uniaxial strain-controlled relaxation, strain-controlled cyclic loading and stress-controlled creep tests are performed on the previously described RVE. The prescribed membrane and shear strains are restricted to $\varepsilon_{\alpha\beta}, \gamma_\alpha \in [-0.03, 0.03]$. Keeping in mind relation (45), the curvatures can be considered small for values less than $\pm 3 \cdot 10^{-3}/mm$, but will actually be restricted to $\kappa_{\alpha\beta} \in [-3.0 \cdot 10^{-4}/mm, 3.0 \cdot 10^{-4}/mm]$ to not yield unnecessary large bending moments compared to the membrane shell stress resultants. For the stress controlled creep tests, the following input space for the shell stress resultants is chosen: $n_{11}, n_{22} \in [-100\,N/mm, 100\,N/mm]$, $n_{12}, q_1, q_2 \in [-50\,N/mm, 50\,N/mm]$ and $m_{11}, m_{22}, m_{12} \in [-2000\,Nmm/mm, 2000\,Nmm/mm]$. The boundaries for the increments are set to $\Delta t \in [5\,s, 15\,s]$ and all tests are performed over a time span of $T = 300\,s$. Within the randomly chosen time span, the cyclic loading is done for a minimum of one, but at maximum for two full periods to eliminate unnecessary sample repetitions.

Three different ANNs will be investigated in the following, which differ in topology and in the number of used training samples. The ANN is referred to as $\text{ANN}_{n_{paths}}$, with n_{paths} being the number of uniaxial relaxation, cyclic and creep paths randomly generated for each strain component. The ANNs are trained multiple times on data sets with $n_{paths} = 5, 10, 20$. An overview of the ANNs and the underlying data can be found in Table 2. All ANNs are trained with the Levenberg-Marquard method with an initial damping parameter $\mu = 0.0001$ until convergence in the error function is reached or until a maximum of 500 epochs, which, due to the nature of the second-order scheme, is completely sufficient to train feasible networks.

To investigate the generalization capabilities of the trained networks, in addition to the training data, further tests on the RVE are performed to generate test data. Each network receives three uniaxial as well as three biaxial time paths, where the choice of the prescribed component as well as the test type is arbitrary. The capability of capturing general strain–stress states will be investigated in Sect. 6 within the numerical examples. The data for the test paths is summarized in Table 3. It is noted, that the uni- and biaxial creep tests with path No. 2 and 5 are not performed stress-controlled, but strain-controlled with the resulting strains that are sampled from the RVE. Therefore, the ANN has to yield the prescribed constant stress, which furthermore means, that no Newton iteration is necessary. Within this approach, the loss function does not have to be modified, which is defined in the output values of the ANN. Alternatively, one could perform stress controlled tests on the ANN and introduce a second loss term, which is defined as the mean squared difference between the creep strains from the RVE and the strains calculated within a Newton scheme with the ANN.

4.4.3 Results and discussion

The performances of the three purely data based ANNs on the chosen test paths is depicted in Fig. 8. As one would expect, all ANNs perform well on the known uniaxial tests, which are part of the training data. For the cyclic loading and relaxation test, ANN_5 is already sufficient to achieve good accuracy with respect to the reference data. Nevertheless, for satisfyingly reproducing the uniaxial pseudo creep test, more data is necessary as only ANN_{20} can replicate the desired constant response.

In the focus of interest though is especially the ability of the ANNs to capture the biaxial behavior, which is only indirectly known from the training data. For example, a uniaxial tension test with prescribed component $\bar{\varepsilon}_{11}$ on the RVE results in a contraction strain ε_{22} for the layered RVE. A biaxial tension test, as performed in test path No. 4, is therefore within the boundaries of the input space, but component wise in a region where no training data is available. Furthermore, for the chosen microstructure, the shear component of the material matrix $\mathbf{D}_s$ is only populated on the main diagonal, which decouples the shear stresses q_1 and q_2. Therefore, the biaxial relaxation test in test path No. 6 is completely unknown to the ANN for the given input combination. Nevertheless, as one can clearly see from the results in Fig. 8, it is possible to not only qualitatively capture the biaxial response with the ANN when trained only with uniaxial data, but to achieve very good results when using enough samples. It should be noted though, that multiple training runs were performed, whereas only the best ANNs are shown. The varying results for different training runs will be discussed in Sect. 5.4. Even with enough data, only about one out of five ANNs yielded good training results and low test errors.

As it is not guaranteed that purely data-based training is a well posed optimization problem, in the following section the training is further enhanced by adding constraints. The motivation is not only to stabilize the training process, but also to reduce the number of necessary data samples, e.g., for the ANN_{20}, 10862 effective stress responses were calculated on the RVE. Consequently, the computation time for more complex microstructures increases, and therefore the time for gathering the training data, which can be drastically reduced by including additional physical information.

5 Training enhancement via derivative constraints

When recapping the derivation of the numerical homogenization scheme for shell RVEs in Sect. 2.3, one should notice that

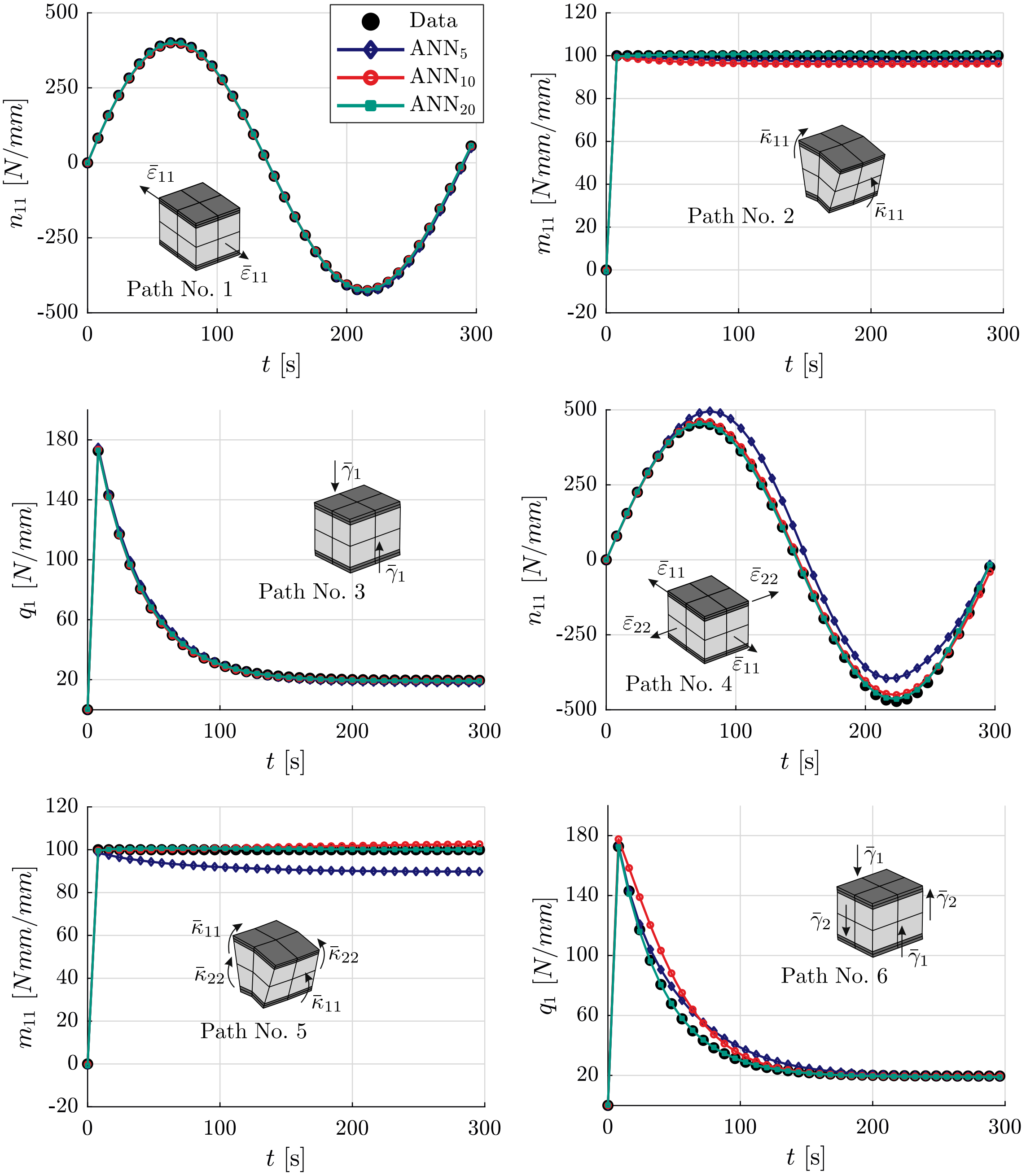


Fig. 8 Results of the three purely data based ANNs for uniaxial and biaxial material tests. Uniaxial material tests are depicted in path No. 1-3, while path No. 4-6 show the results for biaxial testing

until now, only the shell stress resultants were used for the training process of the neural network. So far, the information of the homogenized material matrix **D** has been neglected, which is referred to as purely data-based training, as the error function is only defined in the primary outputs of the neural network. The following section is dedicated to enhancing the training process by including derivative information, which is formally known as Sobolev training, first introduced for deep neural networks in [6]. Therefore, we briefly motivate constraint neural network training in general, before specifically discussing Sobolev training and lastly demonstrating its effectiveness at the material point level.

5.1 ANN training as a constraint optimization problem

To include constraint information into the neural network training process, we follow the approach already discussed in [48], where the framework was introduced in the context of hyperelastic material modeling, with application to experimental data. We continue to use the same ANN feed forward architecture, but enhance the error function (32) via

$$E(\mathbf{w}) = E^{Data}(\mathbf{x}_k; \mathbf{w}) + E^C(\mathbf{x}_k^C; \mathbf{w})\,. \tag{51}$$

Here, $E^{Data}(\mathbf{x}_k; \mathbf{w})$ refers to the conventional neural network error term between the ANN output and the target values, and $E^C(\mathbf{x}_k^C; \mathbf{w})$ is an additional error term, which is evaluated on a separate set of constraint samples $\mathbf{x}_k^C \in P^C$. Depending on the underlying problem, two distinctions for the constraint samples $\mathbf{x}_k^C$ can be made

$$\mathbf{x}_k^C \in P^C, \quad P^C \subseteq P\,, \tag{52}$$

or

$$\mathbf{x}_k^C \in P^C, \quad P^C \cup P = \varnothing\,. \tag{53}$$

Eq. (53) is especially useful when dealing with sparse data, e.g., in the context of ANN material modeling with underlying experimental input–output measurements. The constraints can be formulated on a separate subset of constraint samples, ideally in regions of the input space with missing data, to enforce physical restrictions, stabilize the training process or improve the ANN generalization abilities. In the context of acquiring data via a numerical scheme on the other hand, it would not be feasible to define a separate subset, so consequently, the set of constraint samples is always a real subset of the set of data samples P.

In order to apply the same methods for the ANN training as described in Sect. 3.2 by minimizing the overall error in a mean squared sense with respect to both terms, we introduce the equality equation

$$\mathbf{h}(\mathbf{x}_k^C; \mathbf{w}) = \mathbf{0}, \quad \mathbf{h} \in \mathbb{R}^{n_{eq}}\,, \tag{54}$$

where n_{eq} refers to the number of equality constraints. The constraint is enforced via a classical penalty approach

$$E^C(\mathbf{w}; \mathbf{x}_k^C) = \frac{\bar{\lambda}}{2}(\mathbf{h}(\mathbf{x}_k^C; \mathbf{w}))^2 \tag{55}$$

leading to an unconstrained optimization problem. $\bar{\lambda}$ denotes the scalar penalty parameter, not to be confused with the vector of Lagrange parameters $\boldsymbol{\lambda}$ introduced in Sect. 2.2. For further details and an in-depth discussion on constraint neural network training, the reader is referred to [48]. For the neural network training, depending on the chosen training method, either the gradient of the enhanced error function

$$\nabla_{\mathbf{w}} E(\mathbf{w}) = \nabla_{\mathbf{w}} E^{Data}(\mathbf{x}_k; \mathbf{w}) + \nabla_{\mathbf{w}} E^C(\mathbf{x}_k^C; \mathbf{w}) \tag{56}$$

or, for second order schemes, the Hessian[2]

$$\nabla^2_{\mathbf{w}} E(\mathbf{w}) = \nabla^2_{\mathbf{w}} E^{Data}(\mathbf{x}_k; \mathbf{w}) + \nabla^2_{\mathbf{w}} E^C(\mathbf{x}_k^C; \mathbf{w}) \tag{57}$$

has to be calculated.

5.2 Formulation of the derivative constraint

To include the derivative information of the underlying problem into the neural network training, the equality constraint (54) takes the following form

$$h_{k,ji} = \frac{\partial z_j(\mathbf{x}_k^C; \mathbf{w})}{\partial x_{\Delta\varepsilon_i}} - D_{ji}(\mathbf{x}_k^C), \quad \begin{cases} j = 1, \ldots, 8 \\ i = 1, \ldots, 8\,. \end{cases} \tag{58}$$

Each input sample $\mathbf{x}_k^C$ therefore yields 64 additional equality constraints. It is well known, that a transformation of the input and output vectors of the ANN from a physical to a normalized training space, e.g. setting the boundaries between $[-1, 1]$, improves the convergence behavior, see e.g. [27]. Therefore, a linear transformation of the form

$$\hat{x}_i = (x_i - m_{xi}/s_{xi}), \quad i = 1, \ldots, n_i \tag{59}$$

$$\hat{z}_j = (z_j - m_{zj}/s_{zj}), \quad j = 1, \ldots, m \tag{60}$$

is used here. Hence, the relationship between the physical and transformed network derivatives

$$\frac{\partial z_j}{\partial x_i} = \left(\frac{s_{zj}}{s_{xi}}\right)\frac{\partial \hat{z}_j}{\partial \hat{x}_i} \tag{61}$$

can be inserted in Eq. (58) yielding

$$\hat{h}_{k,ji} = \left(\frac{s_{zj}}{s_{xi}}\right)\frac{\partial \hat{z}_j(\mathbf{x}_k^C; \mathbf{w})}{\partial \hat{x}_{\Delta\varepsilon_i}} - D_{ji}(\mathbf{x}_k^C), \tag{62}$$

which can be rewritten as

$$\hat{h}_{k,ji} = \frac{\partial \hat{z}_j(\mathbf{x}_k^C; \mathbf{w})}{\partial \hat{x}_{\Delta\varepsilon_i}} - \underbrace{\left(\frac{s_{xi}}{s_{zj}}\right) D_{ji}(\mathbf{x}_k^C)}_{\hat{D}_{ji}}\,. \tag{63}$$

The main essential arising from Eq. (63) is the fact, that the constraint can be enforced in the normalized training space, thus, the remedy of unbalanced error terms, as discussed, e.g., for the symmetry constraint of the material tangent in [48], does not occur. As a result, the penalty parameter is typically set to $\bar{\lambda} = 1$, which equally balances the influence

---

[2] Usually, the second order derivatives $\partial^2 E/\partial w_{ij}^2$ are not explicitly calculated but approximated for neural network training, e.g. $\mathbf{H} = \mathbf{J}^T\mathbf{J}$ within the Levenberg-Marquardt scheme, $\mathbf{J}$ denoting the Jacobian.

of data and constraints within the training process. With Eq. (63), the normalized constrained error term reads

$$\hat{E}^C(\mathbf{x}_k^C;\mathbf{w}) = \frac{\bar{\lambda}}{2P^C}\sum_{k=1}^{P^C}\sum_{j=1}^{m}\sum_{i=1}^{n_i}\hat{h}_{k,ji}^2. \tag{64}$$

Consequently, for the quantities in Eq. (56) and Eq. (57), one needs to derive the derivative components $\partial z_j/(\partial x_i \partial w_{rs})$. An analytical algorithmic derivation of the derivative components is given in Appendix B of [48]. A short remark on the use of Sobolev constraints for the ANN training with the Levenberg-Marquardt method is given in Appendix A of this paper.

5.3 Remarks on the physical interpretation of derivative constraints

For a purely hyperelastic material, enforcing the derivative constraints can be interpreted as weakly enforcing conservation of the free energy ψ [48], which is equivalent to the major symmetry of the material tangent

$$\frac{\partial^2\psi}{\partial\mathbf{E}^2} = \mathbf{C} = \mathbf{C}^T. \tag{65}$$

For elastic materials, the rate of dissipation $\dot{\mathcal{D}}$ is always equal to zero. For isothermal viscoelastic materials though, an important requirement is the fulfillment of the Clausius-Duhem inequality

$$\dot{\mathcal{D}} = \mathbf{S} : \dot{\mathbf{E}} - \dot{\psi} \geq= 0. \tag{66}$$

Therefore, for rate dependent materials, the enforcement of the derivative entries as done in Sect. 5.2 is not sufficient for thermodynamic consistency and therefore poses a drawback of the ANN material model developed here. It does though weakly enforce the positive definiteness of the material tangent and allows for the use of symmetric finite element formulations. Approaches, where physical restrictions for inelastic materials are incorporated into the ANN material model can be found, e.g., in [49], where additional constraints are formulated for rate independent plasticity, or [38] where the neural network model is formulated based on two independent potentials and is therefore thermodynamically consistent by construction. These approaches could be readily incorporated within future works.

5.4 ANN material tests with Sobolev constraints

In the following, the response of the neural network trained with Sobolev constraints to the synthetic material tests is analyzed. For this study, the performance of the only on data trained neural network ANN_{20} from Sect. 4.4 and a neural network ANN_{cons}, which is trained on less data but with additional constraints, is compared. For the training of the ANN_{cons}, only five creep, five relaxation and five cyclic loading paths for each strain component are used, which corresponds to the training data of the ANN_5 from Sect. 4.4.

As already mentioned in the previous section, due to the enforcement of the constraints in the normalized training space, for the training of the ANN_{cons} the penalty factor is always set to $\bar{\lambda} = 1$, equally balancing both error terms in the total loss function. Studies on the number of necessary samples $\mathbf{x}_\mathbf{k}^\mathbf{C}$, where the constraints are enforced, show, that adding more than 1000 randomly chosen samples from the entire dataset P does not significantly improve the training process. Conversely, no loss in accuracy is observed when additional samples are included. The constraint is therefore ideally suited for practical applications, since the choice of the magnitude of the penalty parameter is technically obsolete, and the number of constraints simply has to be chosen high enough.

Both networks have the same topology 25-80-80-8. Additionally, with the intention of undermining the improved robustness of the training process when adding constraint samples, not the results of the best trained networks are evaluated, but the average performance over five training cycles on the same data set. The only difference (randomness) within the training process lies in the weight initialization. Averaged quantities will be marked with an overline, e.g. $\overline{\text{ANN}_{\text{cons}}}$. Additionally, the average results ± the standard deviation, s_{20} and s_{cons}, for the five training cycles are computed.

The results of both ANNs on the same testing paths as in the previous study from Table 2 are shown in Fig. 9. Since uniaxial paths are always consistently well approximated regardless of the training cycle, only the responses of the ANNs for the biaxial data will be discussed. Continuous lines show the average response over five training runs, while dashed lines indicate the mean values ± the standard deviations.

When looking at the results for the purely data based neural networks in Fig. 9, one can clearly conclude that, depending on the initial weights, different, and not always optimal, minima are reached within the optimization process. Especially for the biaxial tension test, one can observe significant variations in the results. Furthermore, the average results do not lie on the reference solution, indicating that the optimization problem is not well posed when only using uniaxial training data. It is noted, that for fully converged mean values more training runs would be necessary, which has not been done here, since the overall aim is to develop a well posed problem, which always results in a robust training process. The neural network with additional constraints, on the other hand, does not only yield average results which are in good agreement with the reference solution, but also

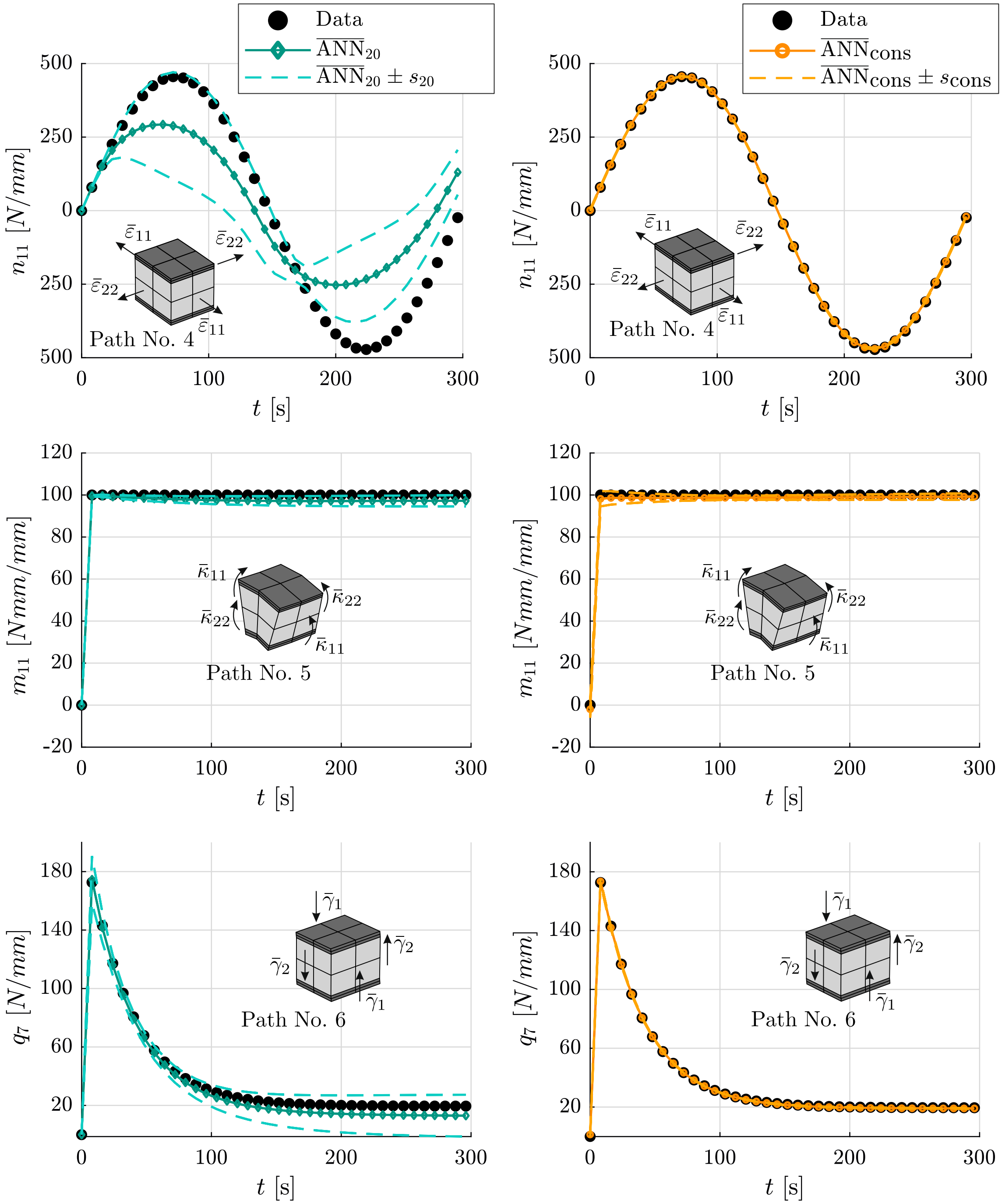


Fig. 9 Results of the purely data based (left column) and Sobolev (right column) trained ANNs for biaxial synthetic material tests. Solid lines show the average results over five training runs, while the dashed lines indicate the mean values ± the standard deviation

displays just low variations in the response over multiple training cycles. As a conclusion, Sobolev training does not only reduce the number of necessary data samples, but also seems to lead to a much better posed optimization problem and, consequently, a more robust training process.

In the following section, the performance of the data-based and constraint enhanced neural network material models will be analyzed within numerical examples. Arbitrary stress–strain states, as well as unknown loading cases, will be applied to fully test their generalization capabilities within structural finite element analysis.

6 Numerical examples

So far, all tests were done only at the material point level. As the scope of this work is to develop an FE^2 surrogate model for challenging finite element simulations, two numerical examples will be investigated. The first example is a layered cylindrical shell under different loading conditions, where the previously introduced RVE will be used. It serves the purpose of emphasizing the feasibility of the neural network material model within a numerical scheme, while not necessarily being the ideal example for a multiscale approach, since here a 3D solution with solid elements is computationally more efficient. The second example is a more complex RVE in order to demonstrate the potential of the method for microstructures with increased computational time. Results will be generated and compared with full 3D solutions, a coupled two scale shell formulation (FE^2) and a shell element with corresponding ANN surrogate models trained on an RVE with the according microstructure. Therefore, the ANN material model is implemented into an extended version of the finite element program FEAP [42]. The transfer between the ANN training in Matlab and FEAP is done via an input text file, which contains all the relevant information to be read, such as how to construct the input vector of the ANN and which history variables need to be processed. Shell elements on the macroscale are based on a three-field variational formulation as originally described in [46]. Details on the current implementation can be found in [47].

6.1 Layered cylindrical shell

For the first example, a layered cylindrical shell as depicted in Fig. 10 is investigated. Taking advantage of the system's symmetry, only one quarter of the structure is modeled by applying appropriate boundary conditions. The cylinder is loaded at the top with a constantly distributed line load $q(t)$, where two different loading patterns are considered. The shell is loaded in the midsurface while here, for the 3D model, the load is equally distributed on the upper and lower surface. The quantity of interest is the displacement $w(t)$ at the corner node i in the midsurface. The shell consists of three layers with the same geometrical and material parameters as the RVE discussed in Sect. 4, see Fig. 3 and Table 1. The quarter of the system is meshed with 16 × 16 shell elements in the circumferential and longitudinal direction with 2 × 2 integration points in each element.

For the 3D solution, $16 \times 16 \times 3$ 27-node solid elements are used, with one element for each layer over the height of the system, which is sufficient for reaching converged solutions. In contrast to the previous studies, where only uni- and biaxial strain–stress states where considered, for this example arbitrary multiaxial states are possible. Furthermore, besides the time paths used for generating the training data for the ANN - relaxation, creep and cyclic loading - unknown loading scenarios, specifically a creep test with posterior unloading and a sawtooth loading pattern, are considered.

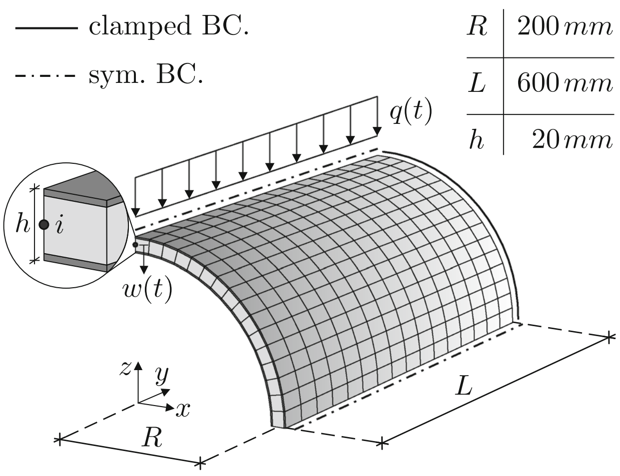


Fig. 10 Layered cylindrical shell with initial geometry, loading and boundary conditions (BC.). Due to the symmetry, only one fourth of the system is modeled

6.1.1 Creep test with unloading

The first loading scenario is a creep test, where an initial load q_0 is applied in the first time step and held constant until $t = 350\,s$, followed by a complete unloading process, see also Fig. 11

$$
\begin{aligned}
q(t) &= q_0 = 10\,N/mm, && t \in [10\,s, 350\,s] \\
q(t) &= 0, && t \in (350\,s, 700\,s]\,.
\end{aligned}
$$

So far, no stress-controlled creep tests where done, as during the tests of the ANNs on the material point level, the strains of a synthetic creep test were the input of the ANN and the desired output was a constant stress. Furthermore, unloading was not included in the training data, but of course has to be adequately approximated by the neural network. Lastly, all explicit time paths used to generate the data were limited to $T = 300\,s$. As only the time increment Δt is part of the input vector, the duration of the test is in general irrelevant for the ANN, which should be visible in the results. Both, the purely data based ANN$_{20}$ and the ANN$_{\text{cons}}$ will be compared to the 3D and FE^2 solutions. All results are generated with a time increment $\Delta t = 10\,s$.

The results for the displacement $w(t)$ for all four models are depicted in Fig. 11. Firstly, the 3D and the multiscale shell solution are found to be in good agreement. Negligible differences in the displacement during the constant loading process are more than likely attributable to local effects resulting from how the load is applied. Furthermore, both ANN models are in good agreement with the FE^2 solution, indicating, that the

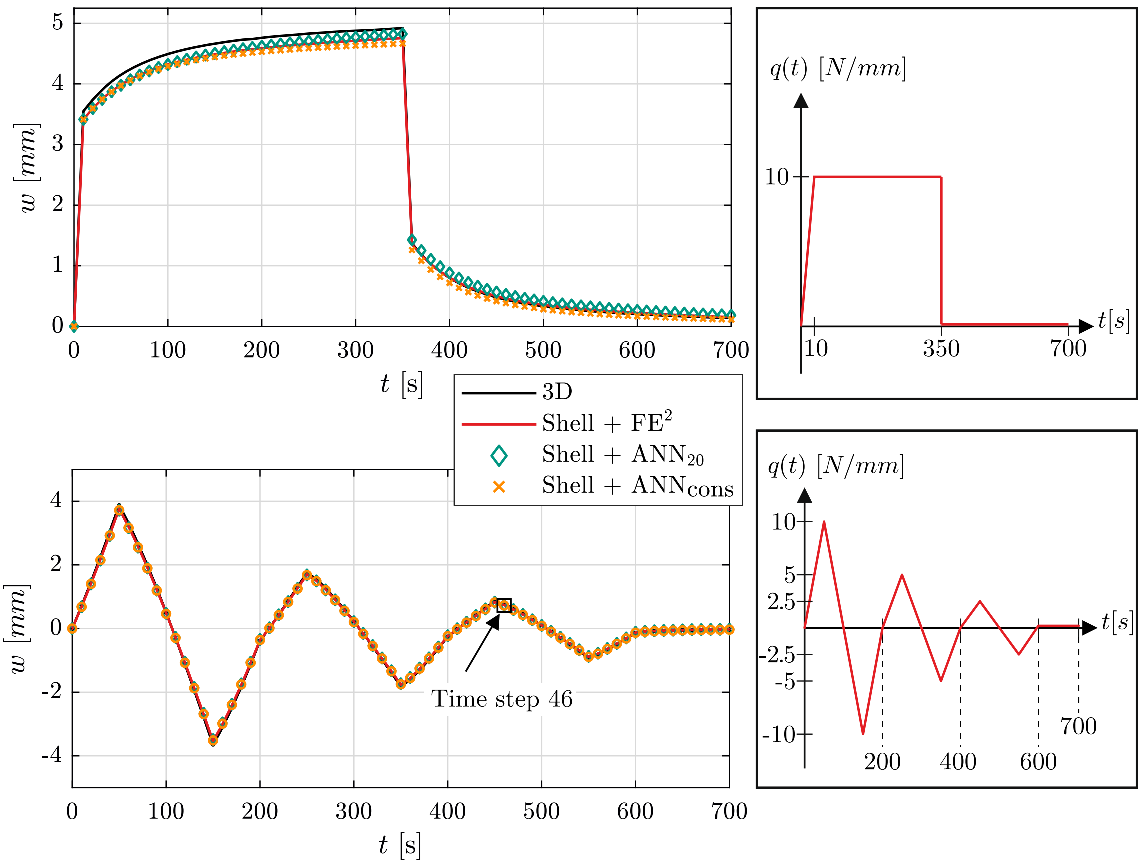


Fig. 11 Loading and corner displacement $w(t)$ with respect to time for a creep test (top) with posterior unloading and a sawtooth loading (bottom) on the cylindrical shell. Results are shown for a 3D solution, as well as a multiscale shell formulation and a shell with two ANN surrogate models (purely data-based and data-based with additional Sobolev constraints)

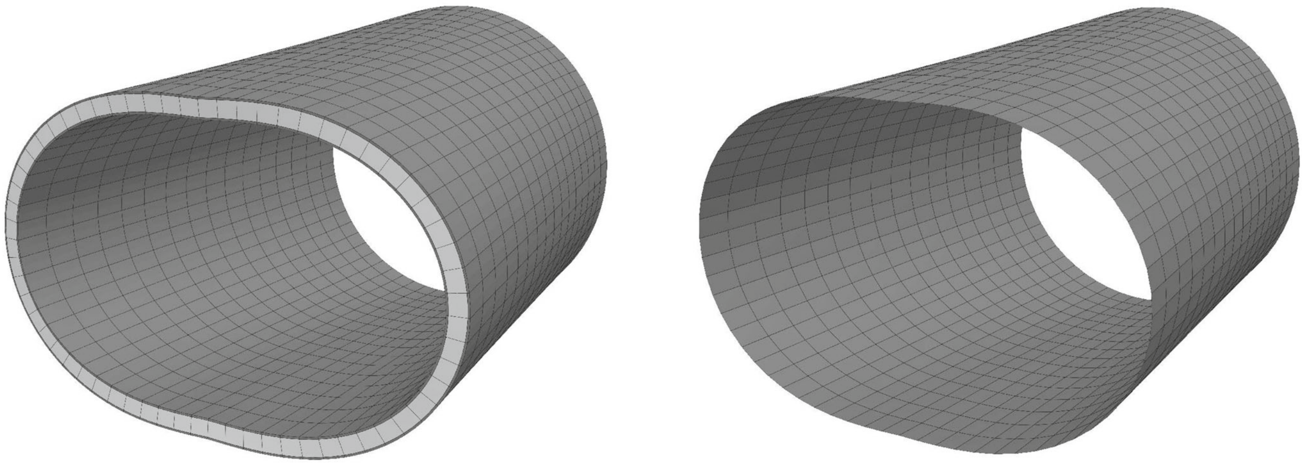
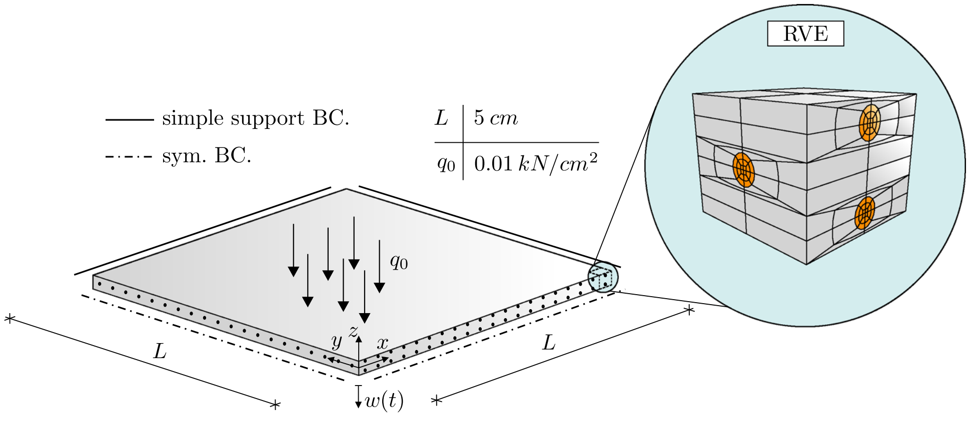


Fig. 12 Complete deformed meshes at $t = 350\,s$ for the creep test. Full-scale 3D solution (left) and the shell + ANN model (right) scaled by a tenfold

Fig. 13 Fiber reinforced plate with initial geometry, loading and boundary conditions (BC) as well as the underlying microstructure (RVE). Due to symmetry, only one quarter of the system is modeled



purely data based, as well constrained enhanced ANN, are well suited for structural finite element simulations. Additionally, one can observe, that even though unloading is not part of the bunch of generated training paths, the neural network is capable of adequately approximating this unknown loading case. Lastly, extrapolating the duration of the test does, as expected, not cause any problems for the ANNs. The deformed meshes of the entire system for the 3D and shell + ANN model are shown in Fig. 12 at $t = 35\,s$, right before unloading.

As already mentioned, from a purely computational aspect, the system does not necessarily have to be modeled with a multiscale approach. A full 3D solid element model is a completely feasible solution for this problem, since the microstructure is not very complex. Alternatively, a layered shell formulation could be used as proposed in [23], which is not considered here. Nevertheless, the problem reveals the potential of a shell with a neural network surrogate model. The computation of the entire displacement-time curve takes over one hour when using a shell with an RVE at each integration point, in the computer environment used. With a neural network surrogate model, the computation time is reduced to less than one minute, which is similar to the computation time of the 3D model.

Two further aspects have to be considered here: first, the analytical material model only consists of one Maxwell element in parallel, making it a simple viscoelastic model with only six history variables at each integration point. Second, the analytical model is completely linear, while the ANN is by construction non-linear, since the material matrix depends on the current strain increment $\mathbf{D}_{\text{ANN}} = \mathbf{D}_{\text{ANN}}(\Delta\boldsymbol{\varepsilon})$. Therefore, approximately four to five iteration steps are necessary for each loading step. Due to the reduced number of degrees of freedoms though, the computation time is still within the range of the 3D model. A short discussion on the convergence behavior will be done in Sect. 6.1.2.

The full potential of the ANN can therefore be exploited for geometric and material non-linear problems and more complex microstructures. While the former will not be thematised within this work, the latter will be discussed in Sect. 6.2.

Table 4 Convergence of the global residual vector for the two ANN material models at time step 46, see Fig. 11

Iteration Step	Norm of residual vector $\|\|\mathbf{g}\|\|$	
	ANN$_{20}$	ANN$_{\text{cons}}$
1	2.67E+03	1.25E+05
2	1.36E+00	1.78E+00
3	5.00E−04	8.48E−04
4	2.27E−07	5.36E−07
5	3.80E−10	6.93E−10
6	2.99E−10	4.36E−10

6.1.2 Sawtooth loading

To test the ANN with a completely unknown loading path, a sawtooth pattern is applied to the cylindrical shell. Here, $q(t)$ is three times linearly in- and decreased over the duration of $T = 200\,s$ of each cycle, reaching a maximum of $\pm 10 N/mm$ in the first, $\pm 5 N/mm$ in the second and $\pm 2.5 N/mm$ in the third one. The shell is completely unloaded in the last $100\,s$. The loading pattern and results for the displacement are shown in Fig. 11. Once again, all models produce almost identical solutions, which here are in even better agreement compared to the creep test. It is assumed that incrementally increasing and decreasing the load results in less pronounced local effects. The example shows, that even though only creep and relaxation tests, as well as cyclic loading with constant amplitudes are included in the training data, good approximations are reached for arbitrary paths.

Besides the results solely for the displacement, the iteration behavior for both ANN material models is shown in Table 4. The shell element used for the calculations is based on a symmetric finite element formulation, meaning sym-

h	l_e	r
$0.2\ cm$	$0.25\ cm$	$0.0255\ cm$

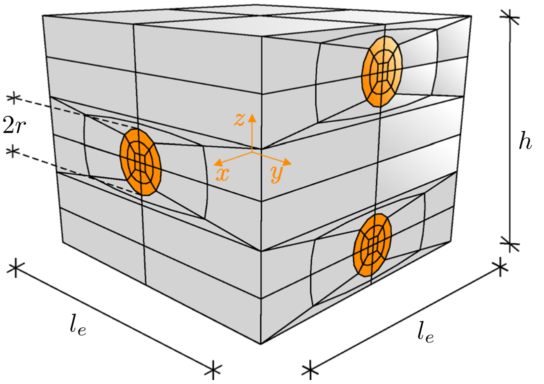


Fig. 14 Geometry of the RVE for the fiber reinforced plate

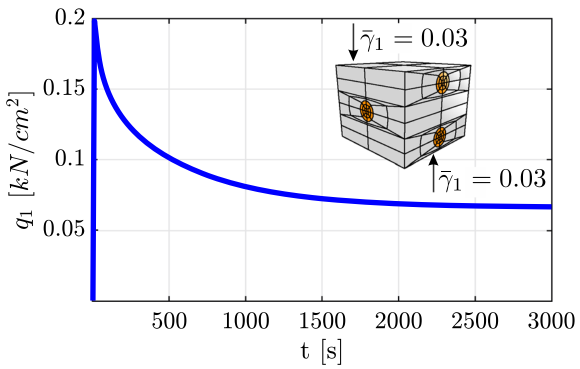


Fig. 15 Relaxation test for the shear stress resultant on the fiber reinforced RVE

metry of the material matrix is assumed. Therefore, $\mathbf{D}$ is symmetrized according to

$$\mathbf{D}_{\text{ANN,sym}} = \frac{1}{2}(\mathbf{D}_{\text{ANN}}^{T} + \mathbf{D}_{\text{ANN}})\,. \tag{67}$$

For both ANNs, the global iteration behavior within the Newton scheme is documented at time step 46, with the norm of the residual vector $||\mathbf{g}||$ being the quantity of interest. Interestingly, both ANNs show quadratic convergence, which indicates, that $\mathbf{D}$ is symmetric in the first place. This allows the conclusion, that even without constraints, a well-trained ANN with a sufficient amount of data also produces a symmetric material matrix. With reference to Sect. 5.4 though, one has to keep in mind, that the constraints dramatically reduce the number of necessary RVE training samples and lead to a much more stable training process.

6.2 Fiber reinforced plate

In the following numerical example, the ANN is applied to a more complex microstructure, where the full potential of the surrogate model regarding the computational cost is demonstrated. The example is based on structures made of fiber reinforced polymers, where the fibers inhibit elastic and the surrounding polymer matrix time-, and oftentimes temperature dependent, viscoelastic material behavior. Furthermore, these structures are mostly thin and subjected to bending dominated deformation, where shell elements are well suited on the macroscale. The shell will be subjected to two different loading scenarios to compare the ANN results with a multiscale and 3D solution, and to further investigate the approximation capability of the ANN across different time steps.

For the system, a quadratic plate under a surface load q_0 is considered, see Fig. 13. Once again, the systems symmetry is taken advantage of by setting appropriate boundary conditions on the symmetry axes. The remaining edges are simply supported, with $w(z) = 0$ for the 3D model. The microstructure consists of a fiber composite laminate with a 90°/0°/90° stacking sequence, where 0° refers to the direction of the $x-$axis. The material parameters for both phases can be found in Table 5. The parameters for the elastic phase correspond to values which fit the typical range for glass fibers. The fibers have a volume fraction of $\mathrm{v}_F = 13\%$. For the matrix, a viscoelastic polymer is chosen, which is typically used in 3D printing processes. From underlying experimental data[3] the material parameters have been fitted using a non-linear regression to a generalized Maxwell model, leading to four Maxwell terms.

The geometric details for the RVE are depicted in Fig. 14. It consists of 216 27-node solid elements resulting in 3175 nodes. Furthermore, periodic boundary conditions with nine additional Lagrange parameters are set, see [11]. Once again, a 3D model of the system serves as a reference solution, where 27-node solid elements are used. The full-scale model consists of twenty RVE-discretizations in $x-$ as well as the $y-$direction resulting in 86400 finite elements and approximately 2.1 million degrees of freedom. Resulting from a convergence study, for the 2D solution a mesh with 12 × 12 elements and 2 × 2 integration points in each element is used, which leads to 766 degrees of freedom on the macroscale. Two loading scenarios are applied to the plate. The quantity of interest is the vertical displacement in the center, where the middle node over the height is used as a reference point for the 3D solution.

For the training of the ANN, the same steps as described in Sect. 4 are undertaken. The sampling is limited to five

---

[3] Data was made available to us by the Chair of Structural Analysis, University of the Bundeswehr Munich.

Table 5 Material parameters for the elastic fiber and viscoelastic matrix of the reinforced plate

	Initial modulus $E_0\,[kN/cm^2]$	Poisson ratio $\nu_0\,[-]$	Maxwell term i	Shear parameter $\mu_i\,[-]$	Relaxation time $\tau_i\,[s]$
Fiber	7300	0.18	–	–	–
Matrix	125	0.3	1	0.40	565
			2	0.17	83
			3	0.10	21
			4	0.03	4

Table 6 Input space limits for the eight shell stress resultants determined from relaxation tests on the RVE

Shell stress resultant		Limits of input space
n_{11}	$\in$	$[-4, 4]\,kN/cm$
n_{22}	$\in$	$[-2, 2]\,kN/cm$
n_{12}	$\in$	$[-0.3, 0.3]\,kN/cm$
m_{11}	$\in$	$[-0.17, 0.17]\,kNcm/cm$
m_{22}	$\in$	$[-0.02, 0.02]\,kNcm/cm$
m_{12}	$\in$	$[-0.01, 0.01]\,kNcm/cm$
q_1	$\in$	$[-0.20, 0.20]\,kN/cm$
q_2	$\in$	$[-0.20.0.20, 4]\,kN/cm$

Table 7 Initial and final displacement results for the creep test

	$w(t = 75\,s)\,[cm]$	$w(t = 3000\,s)\,[cm]$
3D	7.84E-02	9.26E-02
Shell + FE^2	7.51E-02	9.27E-02
Shell + ANN	7.84E-02	9.08E-02

Table 8 Relative computation times for one time-step until convergence is reached

	ANN	FE^2	3D
$t_{ref} = t/t_{ANN}$	1	≈ 100	≈ 450

uniaxial cyclic loading, relaxation and creep paths for each strain component. For the membrane and shear strains, the sampling space is restricted to $[-0.03, 0.03]$ and with relationship from Eq. (45), the limits of the curvatures are $[-0.3/cm, 0.3/cm]$, z_{max} here being $0.1\,cm$. Furthermore, from relaxation tests on the RVE, the limits for the prescribed stress resultants for the creep tests and a suitable interval for the time step and test duration are determined. An exemplary test is shown in Fig. 15 for the shear stress resultant q_1. An induced strain of $\gamma_1 = 0.03$ yields a maximum initial stress resultant of $q_1 \approx 0.2\,kN/cm^2$ and reaches the equilibrium stress after about $3000\,s$. An analogous approach for the remaining strain components yields the limits of the eight shell stress resultants given in Table 6. Due to the anisotropy of the RVE in the local 1/2 direction, in contrary to the previous example, the limits differ in the corresponding directions.

The ANN topology resembles the one of the previous example, consisting of two hidden layers with 80 neurons in each layer. All tests were done over the duration of $T = 3000\,s$ and the time increments are sampled between $[50\,s, 150\,s]$, resulting in 4173 data samples. Additionally, 1000 constraint samples, enforced with a penalty factor of $\bar{\lambda} = 1$ are used. For this example, no additional purely data-based ANN, i.e., without constraints, is trained.

6.2.1 Creep Test

The first considered loading scenario is a creep test, where the surface load q_0 is applied within the first time increment at $t = 75\,s$ and held constant until $t = 3000\,s$. The results for the center displacement with respect to time $w(t)$ are shown in Fig. 16 as well as in Tab. 7. First, the solutions for the FE^2 approach and the 3D model are in good agreement. The calculations with the ANN are done with a time increment $\Delta t = 75\,s$, which yields good results. Interestingly, the ANN solution is in better agreement with the 3D solution for the initial displacement and slightly underestimates the final displacement at $t = 3000\,s$. Nonetheless the error between the ANN and the remaining models for the final displacement is $\approx 2\%$, which is more than feasible for a high dimensional surrogate model based on approximation methods.

When it comes to computational times, one should be cautious with quantitative statements, since, of course, this is highly dependent on the underlying computational capacity and would require to obtain the maximum computational efficiency for each model. Fair comparisons are furthermore difficult, since each method requires individual pre- and post-processing steps, besides the actual calculation. For example, on the one hand, the data generation and training time for the ANN must be considered, but on the other hand, the meshing processes of the full-scale 3D solution are also time-consuming. When simply comparing computation times for the three models until convergence is reached for one time step, the ANN shell model is in the range of a second, the FE^2 approach within a couple of minutes and the 3D model in the range of several minutes, in the computer environment used. This undermines the great potential of the proposed ANN

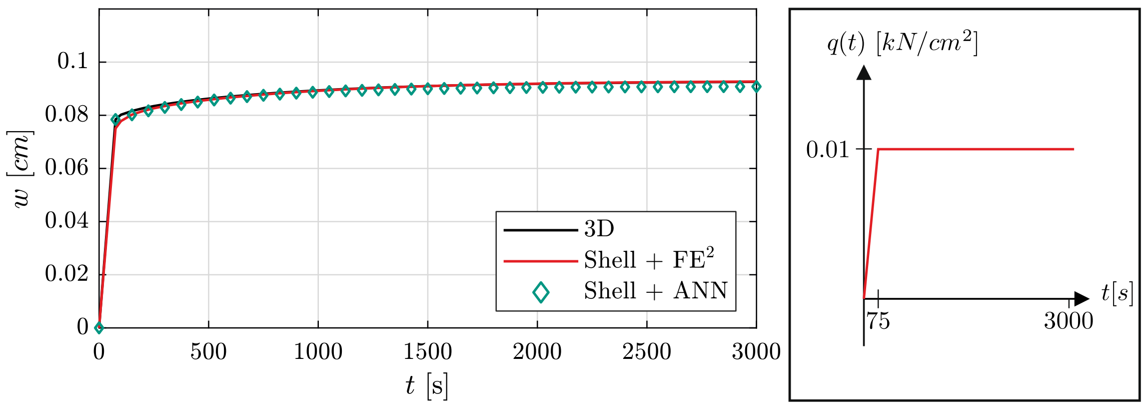


Fig. 16 Center displacement with respect to time for the fiber reinforced plate under constant loading (creep test)

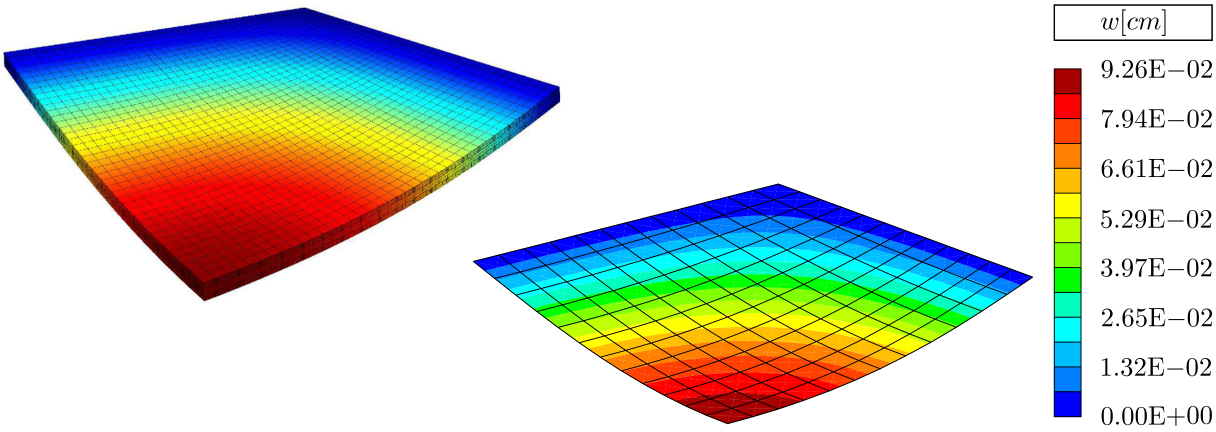


Fig. 17 Deformed mesh of the 3D and 2D shell + ANN solutions at $t = 3000\,s$ for the creep test, scaled by a tenfold

approach for complex and history dependent microstructures. The relative times for all three models are given in Table 8. The quarter of the deformed 3D and 2D meshes are depicted in Fig. 17.

6.2.2 Cyclic loading

Lastly, the behavior of the ANN for different time steps is investigated. Therefore, a cyclic load

$$q(t) = q_0 \sin\left(\frac{4\pi}{3000}t\right), \quad t \in [0\,s, 3000\,s]$$

is applied to the plate, leading to two complete loading and unloading cycles. The results for the center displacement $w(t)$ with respect to time are shown in Fig. 18 for the FE^2 shell model as well as for the shell + ANN material model. The calculations with the ANN were done with time step sizes $\Delta t \in \{50, 75, 100, 125, 150\}$. All ANNs yield feasible approximations, which allows the conclusion that the model is stable for different time steps. As one would expect, $\Delta t = 50\,s$ shows the highest deviation from the reference solution, as it marks the lower boundary of the ANN input space. Calculations with $\Delta t = 150\,s$, which marks the upper bound of the input space, yield surprisingly good results.

7 Conclusion

In the present work, it has been shown, that artificial neural networks have the potential of serving as a feasible surrogate model in replacing time consuming full-scale, or multiscale models for investigating the behavior of thin macrostructures with viscoelastic microstructural properties. Based on a consistent homogenization scheme for the coupling of shells and continua, effective strain–stress data was generated, which was used for the training of a feed forward neural network. Therefore, suitable input and output variables were defined to capture the time and history dependency of the effective strain–stress response. An efficient sampling strategy was developed, involving simple uniaxial synthetic material tests on the RVE, where we have demonstrated, that this is sufficient for the ANN to approximate arbitrary multiaxial stress–strain states. Based on typical experimental setups for viscoelastic materials, cyclic loading scenarios, as well as

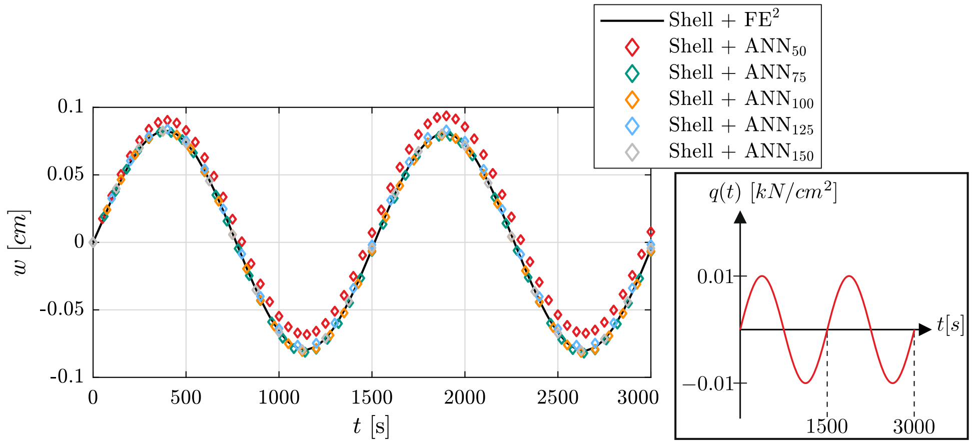


Fig. 18 Center displacement of the fiber reinforced plate with respect to time for a cyclic loading scenario. Results are shown for the shell + FE^2 model and the shell + ANN model by using different time step sizes

creep and relaxations tests were performed on the RVE. To further reduce the number of necessary data samples obtained from numerical simulations on the RVE, additional constraint samples are introduced, making use of already existing information about the homogenized material matrix. Studies at the material point level have revealed, that the constraints not only reduce the number of necessary data samples, but also lead to a more robust ANN training process.

The feasibility of the ANN surrogate model was successfully proven in challenging numerical examples, comparing results to full 3D and FE^2 solutions. Evidence was further provided, that not only known loading scenarios from the training paths were well approximated, but also unknown paths such as unloading and sawtooth loading. Lastly, the full potential of the method was exploited for a complex underlying microstructure, where the use of the ANN only requires a fracture of the computation time when compared to the previously mentioned approaches.

In our opinion, within the framework of replacing multiscale solutions, ANNs are well suited as material models. A great advantage is the fact of unlimited data and additional information about derivatives, which can be included in the training process by formulating constraints. Therefore, the extension to other history dependent material behaviors on the microscale is possible, e.g., plasticity, fracture, or even geometric non-linearities, to analyze microstructural instabilities. Finally, further improvements to the ANN architecture should be considered, e.g., by meeting thermodynamic consistency as proposed in [38].

Appendix A Considering Sobolev constraints in the Levenberg-Marquardt method

The following equations follow mainly the derivations in [51] and are adapted accordingly to the defined Sobolev constraints defined in Sect. 5. With the already given relations for the weight update within the Levenberg-Marquardt scheme in Eq. (35)

$$\mathbf{w}_{p+1} = \mathbf{w}_p + (\mathbf{H} + \mu \mathbf{I})^{-1}\, \mathbf{g}$$

and the definition of the constraint error term in Eq. (64)

$$\hat{E}^C(\mathbf{x}_k^C; \mathbf{w}) = \frac{\bar{\lambda}}{2P^C} \sum_{k=1}^{P^C} \sum_{j=1}^{m} \sum_{i=1}^{n_i} \hat{h}_{k,ji}^2$$

it is necessary to calculate the corresponding gradient

$$\hat{\mathbf{g}}^C := \nabla_{\mathbf{w}} \hat{E}^C(\mathbf{x}_k^C; \mathbf{w}) \tag{A1}$$

and the Hessian

$$\hat{\mathbf{H}}^C := \nabla_{\mathbf{w}}^2 \hat{E}^C(\mathbf{x}_k^C; \mathbf{w}) \tag{A2}$$

of the constrained error term. In index notation, the gradient can be written as

$$\begin{aligned} \hat{g}_r^C = \frac{\partial E^C(\mathbf{x}_k^C; \mathbf{w})}{\partial w_r} &= \frac{\partial}{\partial w_r} \left(\frac{\bar{\lambda}}{2P^C} \sum_{k=1}^{P^C} \sum_{j=1}^{m} \sum_{i=1}^{n_i} \hat{h}_{k,ji}^2 \right) \\ &= \frac{\bar{\lambda}}{P^C} \sum_{k=1}^{P^C} \sum_{j=1}^{m} \sum_{i=1}^{n_i} \frac{\partial \hat{h}_{k,ji}}{\partial w_r} \hat{h}_{k,ji}\, . \end{aligned} \tag{A3}$$

With the definition of the Jacobian

$$\hat{\mathbf{J}}^C = \begin{bmatrix} \dfrac{\partial \hat{h}_{1,11}}{\partial w_1} & \cdots & \dfrac{\partial \hat{h}_{1,11}}{\partial w_{n_w}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial \hat{h}_{P^C,n_i m}}{\partial w_1} & \cdots & \dfrac{\partial \hat{h}_{P^C,n_i m}}{\partial w_{n_w}} \end{bmatrix}, \tag{A4}$$

where the calculation of terms $\partial \hat{h}_{k,ji}/\partial w_r$ has already been discussed in Sect. 5.2, the gradient is expressed as

$$\hat{\mathbf{g}}^C = \frac{\bar{\lambda}}{P^C} \hat{\mathbf{J}}^{C^T} \hat{\mathbf{h}} \tag{A5}$$

with

$$\hat{\mathbf{h}} = \left[\hat{h}_{1,11}, \ldots, \hat{h}_{P^C,n_i m}\right]^T . \tag{A6}$$

Furthermore, for the calculation of the Hessian we can derive

$$\begin{aligned} \hat{H}^C_{rs} &= \frac{\partial E^C(\mathbf{x}^C_k; \mathbf{w})}{\partial w_r \partial w_s} \\ &= \frac{\partial}{\partial w_r} \left(\frac{\bar{\lambda}}{P^C} \sum_{k=1}^{P^C} \sum_{j=1}^{m} \sum_{i=1}^{n_i} \frac{\partial \hat{h}_{k,ji}}{\partial w_s} \hat{h}_{k,ji} \right) \\ &= \frac{\bar{\lambda}}{P^C} \sum_{k=1}^{P^C} \sum_{j=1}^{m} \sum_{i=1}^{n_i} \underbrace{\frac{\partial \hat{h}_{k,ji}}{\partial w_r \partial w_s}}_{\approx 0} \hat{h}_{k,ji} \\ &+ \frac{\partial \hat{h}_{k,ji}}{\partial w_r} \frac{\partial \hat{h}_{k,ji}}{\partial w_s} . \end{aligned} \tag{A7}$$

Taking advantage of the basic assumption of the Gauss-Newton method, where higher order derivatives are neglected as marked in Eq. (A7), what remains is

$$\hat{H}^C_{rs} = \frac{\bar{\lambda}}{P^C} \sum_{k=1}^{P^C} \sum_{j=1}^{m} \sum_{i=1}^{n_i} \frac{\partial \hat{h}_{k,ji}}{\partial w_r} \frac{\partial \hat{h}_{k,ji}}{\partial w_s} , \tag{A8}$$

which can be written in terms of the Jacobian and in vector–matrix notation as

$$\hat{\mathbf{H}}^C = \frac{\bar{\lambda}}{P^C} \hat{\mathbf{J}}^{C^T} \hat{\mathbf{J}}^C . \tag{A9}$$

Acknowledgements We thank Till Weithoff and Georgios Michaloudis from the Chair of Structural Analysis, Bundeswehr University Munich, for kindly providing experimental data from 3D printed materials, which was used for fitting the material parameters of the analytical viscoelastic material model in Sect. 6.2.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Al-Haik M, Hussaini M, Garmestani H (2006) Prediction of nonlinear viscoelastic behavior of polymeric composites using an artificial neural network. Int J Plast 22(7):1367–1392. https://doi.org/10.1016/j.ijplas.2005.09.002
2. An N, Jia Q, Jin H et al (2022) Multiscale modeling of viscoelastic behavior of unidirectional composite laminates and deployable structures. Mater Des 219:110754. https://doi.org/10.1016/j.matdes.2022.110754
3. Chen G (2021) Recurrent neural networks (rnns) learn the constitutive law of viscoelasticity. Comput Mech 67(3):1009–1019. https://doi.org/10.1007/s00466-021-01981-y
4. Coenen EWC, Kouznetsova VG, Geers MGD (2010) Computational homogenization for heterogeneous thin sheets. Int J Numer Meth Eng 83(8–9):1180–1205. https://doi.org/10.1002/nme.2833
5. Cong Y, Nezamabadi S, Zahrouni H et al (2015) Multiscale computational homogenization of heterogeneous shells at small strains with extensions to finite displacements and buckling. Int J Numer Meth Eng 104(4):235–259. https://doi.org/10.1002/nme.4927
6. Czarnecki OSJMSGPRW, M (2017) Sobolev training for neural networks. Advances in neural information processing systems, 30
7. Feyel F, Chaboche JL (2000) Fe2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials. Comput Methods Appl Mech Eng 183(3–4):309–330. https://doi.org/10.1016/S0045-7825(99)00224-8
8. Geers MGD, Coenen EWC, Kouznetsova VG (2007) Multi-scale computational homogenization of structured thin sheets. Modell Simul Mater Sci Eng 15(4):S393–S404. https://doi.org/10.1088/0965-0393/15/4/S06
9. Ghaboussi J, Wu X, Kaklauskas G (1999) Neural network material modelling. Statyba 5(4):250–257. https://doi.org/10.1080/13921525.1999.10531472
10. Gruttmann F, Wagner W (2013) A coupled two-scale shell model with applications to layered structures. Int J Numer Meth Eng 94(13):1233–1254. https://doi.org/10.1002/nme.4496
11. Gruttmann F, Wagner W (2024) A FE2 shell model with periodic boundary conditions for thin and thick shells. Int J Numer Meth Eng. https://doi.org/10.1002/nme.7433
12. Hagan M, Menhaj M (1994) Training feedforward networks with the marquardt algorithm. IEEE Trans Neural Netw 5(6):989–993. https://doi.org/10.1109/72.329697
13. Hambli R (2010) Application of neural networks and finite element computation for multiscale simulation of bone remodeling. J Biomech Eng 132(11):114502. https://doi.org/10.1115/1.4002536
14. Hashash YMA, Jung S, Ghaboussi J (2004) Numerical implementation of a neural network based material model in finite element analysis. Int J Numer Meth Eng 59(7):989–1005. https://doi.org/10.1002/nme.905
15. Hashin Z (1983) Analysis of composite materials-a survey. J Appl Mech 50(3):481–505. https://doi.org/10.1115/1.3167081
16. Hill R (1963) Elastic properties of reinforced solids: some theoretical principles. J Mech Phys Solids 11(5):357–372. https://doi.org/10.1016/0022-5096(63)90036-X
17. J. C. Simo TJRH, (1998) Computational inelasticity. Interdisciplinary Applied Mathematics. https://doi.org/10.1007/b98904
18. Jung S, Ghaboussi J (2006) Neural network constitutive model for rate-dependent materials. Comput Struct 84(15–16):955–963. https://doi.org/10.1016/j.compstruc.2006.02.015

19. Kalina KA, Linden L, Brummund J et al (2023) FE^{ANN}: an efficient data-driven multiscale approach based on physics-constrained neural networks and automated data mining. Comput Mech 71(5):827–851. https://doi.org/10.1007/s00466-022-02260-0
20. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization https://doi.org/10.48550/arXiv.1412.6980
21. Klarmann S, Gruttmann F, Klinkel S (2019) Homogenization assumptions for coupled multiscale analysis of structural elements: beam kinematics. Comput Mech 65(3):635–661. https://doi.org/10.1007/s00466-019-01787-z
22. Klein DK, Ortigosa R, Martínez-Frutos J et al (2022) Finite electro-elasticity with physics-augmented neural networks. Comput Methods Appl Mech Eng 400:115501. https://doi.org/10.1016/j.cma.2022.115501
23. Klinkel S, Gruttmann F, Wagner W (2007) A mixed shell formulation accounting for thickness strains and finite strain 3d material models. Int J Numer Meth Eng 74(6):945–970. https://doi.org/10.1002/nme.2199
24. Kouznetsova, VG (Varvara) (2002) Computational homogenization for the multi-scale analysis of multi-phase materials https://doi.org/Kouznetsova_2002
25. Kurnatowski B, Matzenmiller A (2008) Finite element analysis of viscoelastic composite structures based on a micromechanical material model. Comput Mater Sci 43(4):957–973. https://doi.org/10.1016/j.commatsci.2008.02.026
26. Le BA, Yvonnet J, He Q (2015) Computational homogenization of nonlinear elastic materials using neural networks. Int J Numer Meth Eng 104(12):1061–1084. https://doi.org/10.1002/nme.4953
27. LeCun Y, Bottou L, Orr GB, et al (1998) Efficient backprop pp 9–50. https://doi.org/10.1007/3-540-49430-8_2
28. Lefik M, Schrefler B (2003) Artificial neural network as an incremental non-linear constitutive model for a finite element code. Comput Methods Appl Mech Eng 192(28–30):3265–3283. https://doi.org/10.1016/S0045-7825(03)00350-5
29. Lefik M, Boso D, Schrefler B (2009) Artificial neural networks in numerical modelling of composites. Comput Methods Appl Mech Eng 198(21–26):1785–1804. https://doi.org/10.1016/j.cma.2008.12.036
30. Levenberg K (1944) A method for the solution of certain non-linear problems in least squares. Q Appl Math 2(2):164–168. https://doi.org/10.1090/QAM/10666
31. Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear parameters. J Soc Ind Appl Math 11(2):431–441. https://doi.org/10.1137/0111030
32. Masi F, Stefanou I (2022) Multiscale modeling of inelastic materials with thermodynamics-based artificial neural networks (tann). Comput Methods Appl Mech Eng 398:115190. https://doi.org/10.1016/j.cma.2022.115190
33. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5(4):115–133. https://doi.org/10.1007/BF02478259
34. Miehe C, Koch A (2002) Computational micro-to-macro transitions of discretized microstructures undergoing small strains. Arch Appl Mech (Ingenieur Archiv) 72(4–5):300–317. https://doi.org/10.1007/s00419-002-0212-2
35. Miehe C, Schröder J, Schotte J (1999) Computational homogenization analysis in finite plasticity simulation of texture development in polycrystalline materials. Comput Methods Appl Mech Eng 171(3–4):387–418. https://doi.org/10.1016/S0045-7825(98)00218-7
36. Nemat-Nasser S, Lori M, Datta SK (1996) Micromechanics: overall properties of heterogeneous materials. J Appl Mech 63(2):561–561. https://doi.org/10.1115/1.2788912
37. Rosenkranz M, Kalina KA, Brummund J et al (2023) A comparative study on different neural network architectures to model inelasticity. Int J Numer Meth Eng 124(21):4802–4840. https://doi.org/10.1002/nme.7319
38. Rosenkranz M, Kalina KA, Brummund J et al (2024) Viscoelasticty with physics-augmented neural networks: model formulation and training methods without prescribed internal variables. Comput Mech. https://doi.org/10.1007/s00466-024-02477-1
39. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536. https://doi.org/10.1038/323533a0
40. Saeb S, Steinmann P, Javili A (2016) Aspects of computational homogenization at finite deformations: a unifying review from reuss' to voigt's bound. Appl Mech Rev 68(5):050801. https://doi.org/10.1115/1.4034024
41. Souza FV, Allen DH (2009) Multiscale modeling of impact on heterogeneous viscoelastic solids containing evolving microcracks. Int J Numer Meth Eng 82(4):464–504. https://doi.org/10.1002/nme.2773
42. Taylor RL (2024) FEAP - finite element analysis program. University of California, Berkeley http://projects.ce.berkeley.edu/feap/
43. The MathWorks Inc. (2023) Matlab version: 9.14.0 (R2023a). Natick, Massachusetts, United States: The MathWorks Inc https://www.mathworks.com
44. Tran A, Yvonnet J, He QC et al (2011) A simple computational homogenization method for structures made of linear heterogeneous viscoelastic materials. Comput Methods Appl Mech Eng 200(45–46):2956–2970. https://doi.org/10.1016/j.cma.2011.06.012
45. Unger JF, Könke C (2009) Neural networks as material models within a multiscale approach. Comput Struct 87(19–20):1177–1186. https://doi.org/10.1016/j.compstruc.2008.12.003
46. Wagner W, Gruttmann F (2005) A robust non-linear mixed hybrid quadrilateral shell element. Int J Numer Meth Eng 64(5):635–666. https://doi.org/10.1002/nme.1387
47. Wagner W, Gruttmann F (2020) An improved quadrilateral shell element based on the Hu-Washizu functional. Adv Model Simul Eng Sci 7(1):28. https://doi.org/10.1186/s40323-020-00162-5
48. Weber P, Geiger J, Wagner W (2021) Constrained neural network training and its application to hyperelastic material modeling. Comput Mech 68(5):1179–1204. https://doi.org/10.1007/s00466-021-02064-8
49. Weber P, Wagner W, Freitag S (2023) Physically enhanced training for modeling rate-independent plasticity with feedforward neural networks. Comput Mech 72(4):827–857. https://doi.org/10.1007/s00466-023-02316-9
50. Weber PT (2023). Physikalisch konsistente Modellierung von elastischem und plastischem Materialverhalten mit künstlichen neuronalen Netzen. https://doi.org/10.5445/IR/1000164437
51. Yu H, Wilamowski B (2011) Levenberg-marquardt training pp 1–16. https://doi.org/10.1201/b10604-15