

Data-driven anisotropic finite viscoelasticity using neural ordinary differential equations

Vahidullah Taç^a, Manuel K. Rausch^b, Francisco Sahli Costabal^c, Adrian Buganza Tepole^{a,d,*}

^a Department of Mechanical Engineering, Purdue University, West Lafayette, IN, USA

^b Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX, USA

^c Department of Mechanical Engineering, Pontifica Universidad Catolica de Chile, Santiago, Chile

^d Weldon School of Biomedical Engineering, Purdue University, West Lafayette, IN, USA

Received 20 January 2023; received in revised form 31 March 2023; accepted 2 April 2023

Available online 21 April 2023

Abstract

We develop a fully data-driven model of anisotropic finite viscoelasticity using neural ordinary differential equations as building blocks. We replace the Helmholtz free energy function and the dissipation potential with data-driven functions that a priori satisfy physics-based constraints such as objectivity and the second law of thermodynamics. Our approach enables modeling viscoelastic behavior of materials under arbitrary loads in three-dimensions even with large deformations and large deviations from the thermodynamic equilibrium. The data-driven nature of the governing potentials endows the model with much needed flexibility in modeling the viscoelastic behavior of a wide class of materials. We train the model using stress–strain data from biological and synthetic materials including human brain tissue, blood clots, natural rubber and human myocardium and show that the data-driven method outperforms traditional, closed-form models of viscoelasticity.

© 2023 Elsevier B.V. All rights reserved.

Keywords: Viscoelasticity; Neural ordinary differential equations; Data-driven mechanics; Tissue mechanics; Nonlinear mechanics; Physics-informed machine learning

1. Introduction

The traditional approach to modeling the mechanics of materials has been the use of expert-constructed closed-form constitutive equations. However, there are a large number of such models and no consensus on the best choice for any material. For example, Dal et al. [1] list 44 different hyperelastic constitutive material models for elastomers alone and, similarly, there are several examples of viscoelastic models for rubbers and polymers [2]. Furthermore, closed-form models inherently restrict the type of behaviors that can be described, often rendering them incapable of accurately capturing the response of many materials. Both of these problems can be solved with the help of data-driven methods as has been demonstrated various times for the case of hyperelasticity [3–7], but remains an ongoing area of investigation for dissipative phenomena such as viscoelasticity.

* Corresponding author at: Department of Mechanical Engineering, Purdue University, West Lafayette, IN, USA.

E-mail address: abuganza@purdue.edu (A.B. Tepole).

The past few years have witnessed a rapid adoption of data-driven methods in constitutive modeling of materials. Initially some studies used feed-forward fully-connected neural networks (FFNNs) to predict the derivatives of strain energy with respect to invariants of deformation. Liu et al. [5] enforced convexity of strain energy with respect to the elements of the Green strain tensor by adding loss terms that ensure the Hessian matrix of the strain energy is positive semi-definite. Other studies [8,9] used the monotonicity of stress as an alternative criterion to enforce the same class of convexity. However, polyconvexity is the more physically relevant criterion. We have previously enforced polyconvexity of the strain energy through the loss function [4]. All of these studies enforced the desired constraint in the weak sense using specially designed loss functions that penalize deviations from convexity. Later, methods were developed using input convex neural networks (ICNNs) [10] and neural ordinary differential equations (NODEs) [11] to satisfy convexity conditions *a priori* [3,12–14]. A parallel approach to constitutive modeling of hyperelasticity has been the automated discovery of constitutive laws from a large catalog of existing closed-form expressions [15–18].

In comparison, adoption of data-driven methods in viscoelasticity has been slow. Some forms of viscoelasticity models were developed for specific use cases, but no general, physics-informed, fully data-driven model of finite viscoelasticity has been developed so far, to the best of our knowledge. For example, Salahshoor and Ortiz [19] took a model-free data-driven approach to viscoelasticity in the frequency domain to simulate wave propagation in viscoelastic solids. Wang et al. [20] recently proposed a method of numerically approximating the solution of time-dependent ODEs and PDEs over long temporal domains using Physics Informed Deep Operator Neural Networks (PI DeepONets). The DeepONets are trained with a distribution of initial conditions and predict the solution of the ODE/PDEs in a short time interval. This solution is then used as the initial conditions for the next time interval. The authors of [21] use artificial neural networks to approximate the relationship between stress and strain increments in the discretized form of a viscoelastic constitutive law. And in [22] the stress update algorithm is replaced entirely by recurrent neural networks (RNNs). However, the use of such approaches in large scale simulations would require additional safeguards because, in general, when the relationship between stress and strain is approximated using plain, unconstrained artificial neural networks (including the recurrent unit in an RNN) the positive dissipation of energy is not guaranteed, resulting in possible violations of the second law of Thermodynamics. An approach to achieve physics-informed viscoelasticity models is to represent the viscoelastic response as a Prony series with many terms and identify the most relevant parameters [23], or to do sparse systems identification on a large library of models [24].

The lethargy in adoption of data driven methods in viscoelasticity can largely be attributed to two major challenges in developing fully data-driven, flexible, but physically realistic models of viscoelasticity: (i) the difficulties associated with ensuring dissipation of energy is non-negative and, (ii) high computational costs associated with optimizing the parameters of such models.

In this study we develop a fully data-driven model of anisotropic finite viscoelasticity using neural ordinary differential equations (NODEs). We adopt the framework of variational finite viscoelasticity from [25] as the mathematical foundation of the model and use the method of [26] to extend it to the anisotropic case. We propose NODE-based functions for the governing potentials: Helmholtz free energy function, the *creep potential*, and a NODE-based evolution equation for the fiber stress. We show that some appropriate convexity conditions on the creep potential can guarantee that the dissipation of energy is positive. We leverage prior knowledge accumulated in the field of hyperelasticity to design automatically convex data-driven creep potential functions using NODEs. The method provided in [25] for solving the evolution equation involves the use of a predictor–corrector algorithm, which is computationally costly and unstable when combined with gradient-based optimization algorithms, which are the mainstays of data-driven methods. We reformulate the problem as a system of ordinary differential equations (ODEs) thereby bypassing the iterative approach of the predictor–corrector algorithm. The Helmholtz free energy function is polyconvex with respect to the deformation gradient by construction [3]. We train the model with experimental data from biological and synthetic materials including brain tissue samples, blood clots, natural rubber and human myocardium. A brief overview of this approach is shown in Fig. 1.

2. Materials and methods

2.1. Theory of finite viscoelasticity

We assume that the material of interest consists of an isotropic matrix and a finite number of embedded fibers that endow the material with different mechanical behavior in certain directions, resulting in an overall anisotropic

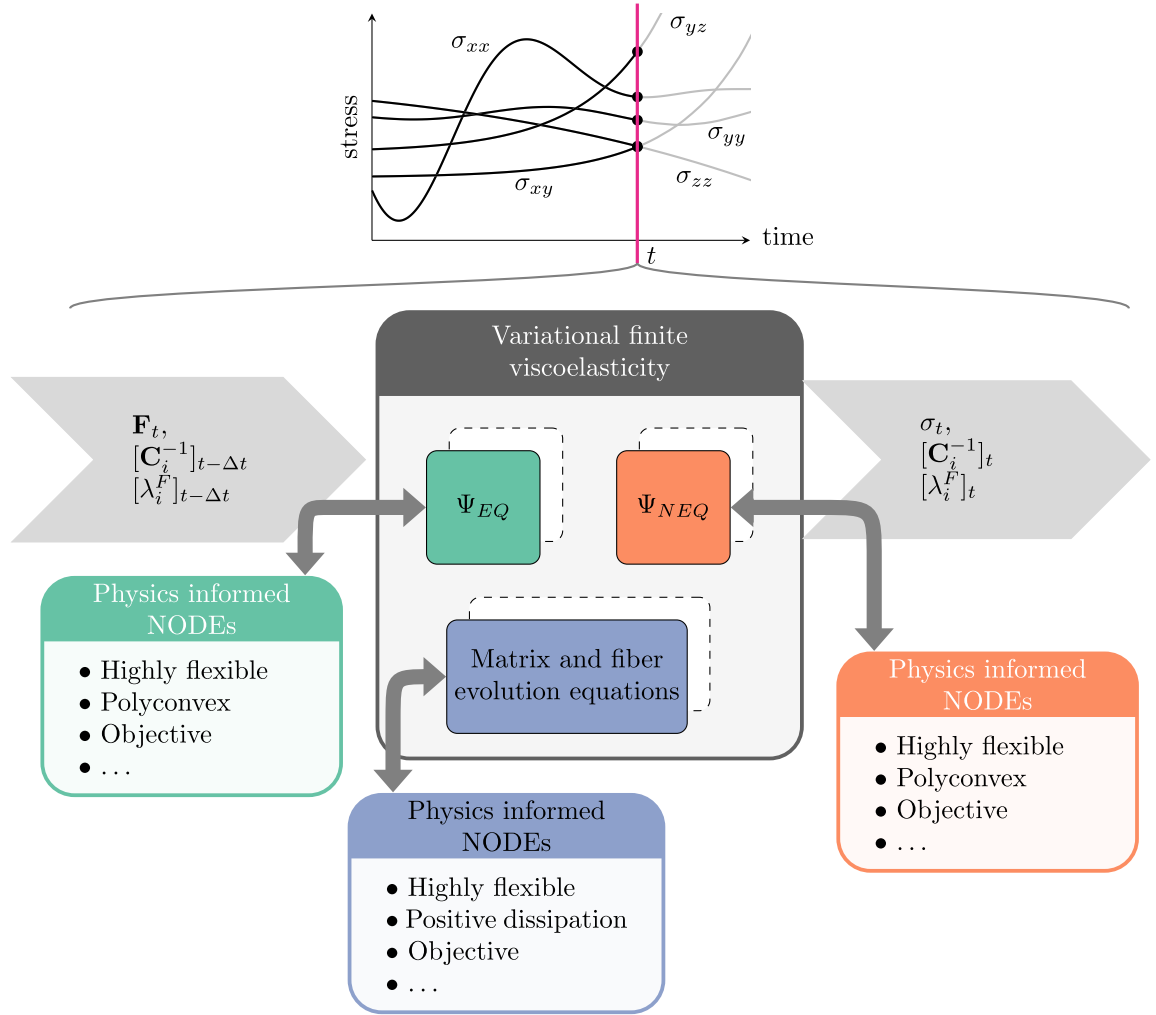


Fig. 1. Overview of the proposed method to construct data-driven constitutive models of finite viscoelasticity that automatically satisfy objectivity, polyconvexity and the dissipation inequality. The governing potentials Ψ_{EQ} , Ψ_{NEQ} and Φ are constructed using neural differential equations (NODEs) to satisfy physics-based constraints *a priori*. The constitutive model can be used to compute stress histories for arbitrary deformation gradients and rates of deformation.

behavior. To model this phenomena, we use the parallel multiplicative decomposition of the deformation gradient into elastic and inelastic parts (denoted by subscripts $(\cdot)_e$ and $(\cdot)_i$, respectively) for both the matrix and the fibers (denoted by superscripts $(\cdot)^M$ and $(\cdot)^F$, respectively)

$$\mathbf{F} = \mathbf{F}_e^M \mathbf{F}_i^M = \mathbf{F}_e^F \mathbf{F}_i^F \quad (1)$$

We additively split the Helmholtz free energy function into isotropic matrix and anisotropic fiber parts, both of which are further additively split into equilibrium (EQ) and non-equilibrium (NEQ) portions

$$\Psi = \Psi_{EQ}^M(\mathbf{C}) + \Psi_{EQ}^F(\mathbf{C}) + \Psi_{NEQ}^M(\mathbf{C}_e^M) + \Psi_{NEQ}^F(\mathbf{C}_e^F) \quad (2)$$

where $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ is the right Cauchy–Green deformation tensor and $\mathbf{C}_e^M = \mathbf{F}_e^{MT} \mathbf{F}_e^M$ and $\mathbf{C}_e^F = \mathbf{F}_e^{FT} \mathbf{F}_e^F$ are the corresponding elastic counterparts for matrix and fiber, respectively. The matrix free energy functions are isotropic by construction and only depend on the isotropic invariants of $\mathbf{C}$ and $\mathbf{C}_e$, whereas the fiber functions are anisotropic and depend explicitly on deformation pseudo-invariants associated with fiber directions $\mathbf{V}$ defined in the reference configuration.

2.2. Helmholtz free energy

The construction of the free energy function is based on the NODE approach from our previous publication [3].

2.2.1. Matrix phase

We express the two free energy functions Ψ_{EQ}^M and Ψ_{NEQ}^M as the sum of an isochoric and a volumetric part

$$\Psi_{EQ}^M(\mathbf{C}) = \Psi_{EQ}^{iso}(\bar{\mathbf{C}}) + \Psi_{EQ}^{vol}(J) \quad (7)$$

$$\Psi_{NEQ}^M(\mathbf{C}_e) = \Psi_{NEQ}^{iso}(\bar{\mathbf{C}}_e^M) + \Psi_{NEQ}^{vol}(J_e^M) \quad (8)$$

where $J = \det \mathbf{F}$, $J_e^M = \det \mathbf{F}_e^M$ and $\bar{\mathbf{C}}$ and $\bar{\mathbf{C}}_e^M$ are the isochoric parts of $\mathbf{C}$ and $\mathbf{C}_e^M$, respectively

$$\bar{\mathbf{C}} = J^{-2/3} \mathbf{C} \quad (9)$$

$$\bar{\mathbf{C}}_e^M = J_e^{-2/3} \mathbf{C}_e^M. \quad (10)$$

We require that the two free energy functions Ψ_{EQ}^M and Ψ_{NEQ}^M be objective and polyconvex in $\mathbf{F}$ and $\mathbf{F}_e^M$ respectively. To this end, we express the Ψ_{EQ}^{iso} and Ψ_{NEQ}^{iso} in terms of isotropic polyconvex invariants of $\bar{\mathbf{C}}$ and $\bar{\mathbf{C}}_e^M$ [27], respectively, and adopt an additive split into two independent functions

$$\Psi_{EQ}^{iso}(\mathbf{C}) = \Psi_{EQ}^{iso}(\bar{I}_1, \bar{I}_2) = \Psi_{EQ,1}(\bar{I}_1) + \Psi_{EQ,2}(\bar{I}_2) \quad (11)$$

$$\Psi_{NEQ}^{iso}(\mathbf{C}_e^M) = \Psi_{NEQ}^{iso}(\bar{I}_1^e, \bar{I}_2^e) = \Psi_{NEQ,1}(\bar{I}_1^e) + \Psi_{NEQ,2}(\bar{I}_2^e) \quad (12)$$

with

$$\bar{I}_1 = \text{tr}(\bar{\mathbf{C}}) \quad (13)$$

$$\bar{I}_2 = \bar{I}_2^{3/2} - 3\sqrt{3} \text{ with } \bar{I}_2 = \frac{1}{2} (\bar{I}_1^2 - \text{tr}(\bar{\mathbf{C}}^2)) \quad (14)$$

$$\bar{I}_1^e = \text{tr}(\bar{\mathbf{C}}_e^M) \quad (15)$$

$$\bar{I}_2^e = (\bar{I}_2^e)^{3/2} - 3\sqrt{3} \text{ with } \bar{I}_2^e = \frac{1}{2} ((\bar{I}_1^e)^2 - \text{tr}(\bar{\mathbf{C}}_e^M)^2). \quad (16)$$

Then, to guarantee polyconvexity of Ψ_{EQ} and Ψ_{NEQ} it is sufficient to require that the functions $\Psi_{EQ,1}$, $\Psi_{EQ,2}$, Ψ_{EQ}^{vol} , $\Psi_{NEQ,1}$, $\dots$ be convex and non-decreasing. Or equivalently, it is sufficient for $\partial \Psi_{EQ,1}/\partial \bar{I}_1$, $\partial \Psi_{EQ,2}/\partial \bar{I}_2$ be monotonic and non-negative. We use NODEs to approximate the functions $\partial \Psi_{EQ,1}/\partial \bar{I}_1$, $\partial \Psi_{EQ,2}/\partial \bar{I}_2$, such that they are monotonic and non-negative as explained in Section 2.6.

2.2.2. Fiber phase

The Helmholtz free energy function of the fiber phase can be expressed in terms of anisotropic pseudo-invariants of the deformation. In this study we use the following form for Ψ_{EQ}^F and Ψ_{NEQ}^F :

$$\Psi_{EQ}^F(\mathbf{C}) = \Psi_{EQ}^F(I_4) \quad (17)$$

$$\Psi_{NEQ}^F(\mathbf{C}_e^F) = \Psi_{NEQ}^F(I_{4e}^F) \quad (18)$$

with

$$I_4 = \mathbf{C} : \mathbf{M} \quad (19)$$

$$I_{4e}^F = \frac{\mathbf{C} : \mathbf{M}}{\mathbf{C}_i^F : \mathbf{M}} \quad (20)$$

where $\mathbf{M} = \mathbf{V} \otimes \mathbf{V}$ is the structure tensor defined in terms of material fiber direction vector $\mathbf{V}$. In general, the fiber direction vector can be an arbitrary vector in 3D space. However, for many biological tissues the vector can reasonably be assumed to lie in a plane, especially if plane stress or strain deformations are considered, resulting in a 2D vector of the form $\mathbf{V} = [\cos \theta \quad \sin \theta \quad 0]^T$ with θ the material fiber direction. This assumption simplifies the computations for the fiber phase since it does not contribute to the out-of-plane response. As in the matrix case, we use NODEs to approximate the derivative functions $\partial \Psi_{EQ}^F/\partial I_4$, $\partial \Psi_{NEQ}^F/\partial I_{4e}^F$.

2.3. Dissipation inequality

The constitutive equations have to satisfy the entropy inequality, which is equal to the *internal dissipation inequality* in the case of isothermal processes

$$\frac{1}{2} \mathbf{S} : \dot{\mathbf{C}} - \dot{\Psi} \geq 0 \quad (17)$$

with $\mathbf{S} = 2\partial\Psi/\partial\mathbf{C}$ the second Piola–Kirchhoff stress. It can be shown that substitution of (2) into (17) leads to [26]

$$-2 \frac{\partial\Psi}{\partial\mathbf{C}_i^M} : \frac{1}{2} \dot{\mathbf{C}}_i^M - 2 \frac{\partial\Psi}{\partial\mathbf{C}_i^F} : \frac{1}{2} \dot{\mathbf{C}}_i^F \geq 0. \quad (18)$$

The two terms in (18) correspond to the dissipation of energy in the matrix and the fiber. We require that the two terms be non-negative individually. This corresponds to the assumption that the dissipation mechanisms in the matrix and the fiber are independent of each other, resulting in two separate criteria

$$-2 \frac{\partial\Psi}{\partial\mathbf{C}_i^M} : \frac{1}{2} \dot{\mathbf{C}}_i^M \geq 0 \quad (19)$$

$$-2 \frac{\partial\Psi}{\partial\mathbf{C}_i^F} : \frac{1}{2} \dot{\mathbf{C}}_i^F \geq 0. \quad (20)$$

2.4. Evolution law for the matrix phase

The inequality of (19) can be more conveniently written in spatial form as [25]

$$-\boldsymbol{\tau}_{NEQ}^M : \frac{1}{2} (\mathcal{L}\mathbf{b}_e^M)(\mathbf{b}_e^M)^{-1} \geq 0 \quad (21)$$

where $\boldsymbol{\tau}_{NEQ}^M = 2\mathbf{F}_e^M(\partial\Psi_{NEQ}^M/\partial\mathbf{C}_e^M)\mathbf{F}_e^{MT}$ is the Kirchhoff stress of the non-equilibrium part of the deformation, $\mathcal{L}(\cdot)$ denotes the Lie derivative and $\mathbf{b}_e^M = \mathbf{F}_e^M\mathbf{F}_e^{MT}$ is the left Cauchy–Green deformation tensor of the elastic part of the matrix deformation. Eq. (21) implies the need of a constitutive equation that specifies the time evolution of $\mathbf{b}_e^M$. One option is to introduce a scalar valued function Φ , known as the *creep potential*, which defines the evolution equation for the matrix phase

$$-\frac{1}{2}(\mathcal{L}\mathbf{b}_e^M)\mathbf{b}_e^{M-1} = \frac{\partial\Phi}{\partial\boldsymbol{\tau}_{NEQ}^M}. \quad (22)$$

Substituting (22) to (21) and carrying out the double contraction results in

$$\sum_{i=1}^3 \sum_{j=1}^3 (\boldsymbol{\tau}_{NEQ}^M)_{ij} \frac{\partial\Phi}{\partial(\boldsymbol{\tau}_{NEQ}^M)_{ij}} \geq 0 \quad (23)$$

We require that this inequality be satisfied for each $(\boldsymbol{\tau}_{NEQ}^M)_{ij}$ independently, i.e.,

$$(\boldsymbol{\tau}_{NEQ}^M)_{ij} \frac{\partial\Phi}{\partial(\boldsymbol{\tau}_{NEQ}^M)_{ij}} \geq 0 \text{ (no sum)} \quad \forall i, j \in \{1, 2, 3\} \quad (24)$$

$$\implies \frac{\partial\Phi}{\partial(\boldsymbol{\tau}_{NEQ}^M)_{ij}} \geq 0 \text{ when } (\boldsymbol{\tau}_{NEQ}^M)_{ij} \geq 0 \text{ and} \quad (25)$$

$$\frac{\partial\Phi}{\partial(\boldsymbol{\tau}_{NEQ}^M)_{ij}} \leq 0 \text{ when } (\boldsymbol{\tau}_{NEQ}^M)_{ij} \leq 0 \quad (26)$$

Thus, it can be seen that to satisfy the dissipation inequality (17) it is sufficient to ensure

1. Φ is a convex function of $\boldsymbol{\tau}_{NEQ}^M$
2. $\arg \min_{\boldsymbol{\tau}_{NEQ}^M} \Phi = \mathbf{0}$ (i.e., Φ attains its global minimum at $\boldsymbol{\tau}_{NEQ}^M = \mathbf{0}$)

but the principle of objectivity of stress introduces a third requirement

3. Φ is an objective function of $\boldsymbol{\tau}_{NEQ}$, i.e. $\Phi(\mathbf{R}\boldsymbol{\tau}_{NEQ}^M\mathbf{R}^T) = \Phi(\boldsymbol{\tau}_{NEQ}^M)$ for all proper orthogonal tensors $\mathbf{R}$

2.4.1. Convex and objective creep potentials

We are interested in finding convex and objective functions $\Phi : \text{Sym}^{3 \times 3} \rightarrow \mathbb{R}$ such that $\arg \min_{\boldsymbol{\tau}_{NEQ}^M} \Phi = \mathbf{0}$. Here $\text{Sym}^{3 \times 3}$ denotes the space of symmetric 3×3 tensors. One approach to this problem is to make Φ a function of the invariants g_i of $\boldsymbol{\tau}_{NEQ}^M$, which in turn can be expressed in terms of its eigenvalues, τ_i .

$$\Phi(\boldsymbol{\tau}_{NEQ}^M) = \hat{\Phi}(g_1(\tau_1, \tau_2, \tau_3), g_2(\tau_1, \tau_2, \tau_3), \dots) . \quad (27)$$

Some examples of functions $g_i()$ include $\tau_1, \tau_2, \tau_1 + \tau_2, I_1^\tau, (I_1^\tau)^2, I_2^\tau, \dots$, where I_j^τ ($j = 1, 2, 3$) are the *principal* invariants of $\boldsymbol{\tau}_{NEQ}^M$.

From the fundamentals of convex functions of multiple variables [28] we know that a composition of functions $f = h(\mathbf{g}(\mathbf{x})) = h(g_1(\mathbf{x}), g_2(\mathbf{x}), \dots)$ is convex in $\mathbf{x}$ if

1. $g_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, 2, \dots, k$ are convex in each argument
2. $h : \mathbb{R}^k \rightarrow \mathbb{R}$ is convex and non-decreasing in each argument

This indicates that for Φ to be a convex function of $\boldsymbol{\tau}_{NEQ}^M$ we must ensure that $\hat{\Phi}$ is convex and non-decreasing and g_i are convex in $\boldsymbol{\tau}_{NEQ}^M$. Some suitable examples of such $g_i()$ functions are:

- The first principal invariant, $I_1^\tau = \text{tr}(\boldsymbol{\tau}_{NEQ}^M)$ and its square $(I_1^\tau)^2$
- $(I_1^\tau)^2 - \alpha I_2^\tau$ where $0 \leq \alpha \leq 3$ and $I_2^\tau = \text{tr}(\text{cof}(\boldsymbol{\tau}_{NEQ}^M))$ is the second principal invariant of $\boldsymbol{\tau}_{NEQ}^M$
- The largest eigenvalue τ_1 of $\boldsymbol{\tau}_{NEQ}^M$
- $-\tau_3$ where τ_3 is the smallest eigenvalue of $\boldsymbol{\tau}_{NEQ}^M$
- Sum of k largest eigenvalues

$I_1^\tau = (\boldsymbol{\tau}_{NEQ}^M)_{11} + (\boldsymbol{\tau}_{NEQ}^M)_{22} + (\boldsymbol{\tau}_{NEQ}^M)_{33}$ is linear in elements of $\boldsymbol{\tau}_{NEQ}^M$ and therefore its convexity is obvious. Proofs of convexity of $(I_1^\tau)^2$ and $(I_1^\tau)^2 - \alpha I_2^\tau$ are given in [Appendix A](#). For the proofs of the other functions given here, or more examples of convex functions the readers are referred to [28].

In this study we adopt an additive form for $\hat{\Phi}$ and use the first three of the functions listed above, i.e. $g_1 = I_1^\tau, g_2 = (I_1^\tau)^2$ and $g_3 = (I_1^\tau)^2 - 3I_2^\tau$. With this, (27) becomes

$$\Phi(\boldsymbol{\tau}_{NEQ}^M) = \hat{\Phi}_1(I_1^\tau) + \hat{\Phi}_2((I_1^\tau)^2) + \hat{\Phi}_3((I_1^\tau)^2 - 3I_2^\tau) . \quad (28)$$

This form of Φ is sufficient for most practical applications. In fact it can be shown that the creep potential used in [25] consists of just a linear combination of $g_2 = (I_1^\tau)^2$ and $g_3 = (I_1^\tau)^2 - 3I_2^\tau$, which is a special case of the class of creep potentials we can capture with NODEs.

Finally, for the requirement on the global minimum we have

$$\arg \min_{\boldsymbol{\tau}_{NEQ}^M} \Phi(\boldsymbol{\tau}_{NEQ}^M) = \arg \min_{\boldsymbol{\tau}_{NEQ}^M} \left(\hat{\Phi}_1(I_1^\tau) + \hat{\Phi}_2((I_1^\tau)^2) + \hat{\Phi}_3((I_1^\tau)^2 - 3I_2^\tau) \right) = \mathbf{0} \quad (29)$$

we require that the functions $\hat{\Phi}_i$ individually attain their minima at $\boldsymbol{\tau}_{NEQ}^M = \mathbf{0}$, i.e.

$$\arg \min_{\boldsymbol{\tau}_{NEQ}^M} \hat{\Phi}_1(I_1^\tau(\boldsymbol{\tau}_{NEQ}^M)) = \mathbf{0} \quad (30)$$

$$\arg \min_{\boldsymbol{\tau}_{NEQ}^M} \hat{\Phi}_2\left((I_1^\tau(\boldsymbol{\tau}_{NEQ}^M))^2\right) = \mathbf{0} \quad (31)$$

$$\arg \min_{\boldsymbol{\tau}_{NEQ}^M} \hat{\Phi}_3\left((I_1^\tau(\boldsymbol{\tau}_{NEQ}^M))^2 - 3I_2^\tau(\boldsymbol{\tau}_{NEQ}^M)\right) = \mathbf{0} \quad (32)$$

or, considering the fact that $I_1^\tau = (I_1^\tau)^2 = (I_1^\tau)^2 - 3I_2^\tau = 0$ when $\boldsymbol{\tau}_{NEQ}^M = \mathbf{0}$, this condition can alternatively be stated as

$$\arg \min_{g_i} \hat{\Phi}_1(g_i) = 0, \quad i = \{1, 2, 3\} . \quad (33)$$

The convexity requirements can be equivalently stated in terms of the derivatives of $\hat{\Phi}_i$

$$\hat{\Phi}_i(g_i) \text{ convex} \iff \frac{d\hat{\Phi}_i}{dg_i} \text{ monotonically increasing} \quad (34)$$

$$\hat{\Phi}_i(g_i) \text{ non-decreasing} \iff \frac{d\hat{\Phi}_i}{dg_i} \text{ non-negative} \quad (35)$$

$$\arg \min_{g_i} \hat{\Phi}_i = 0 \iff \left. \frac{d\hat{\Phi}_i}{dg_i} \right|_{g_i=0} = 0 \quad (36)$$

We use NODEs to approximate the derivative functions $d\hat{\Phi}_i/dg_i$, such that these three requirements are satisfied *a priori*, see also Section 2.6. At this point we need to make two remarks

Remark 2.1. Φ_1 is not in the form of a composition of functions. Rather, since I_1^τ is linear in the elements of τ_{NEQ} , then $\hat{\Phi}_1(I_1^\tau) = \hat{\Phi}_1((\tau_{NEQ}^M)_{11} + (\tau_{NEQ}^M)_{22} + (\tau_{NEQ}^M)_{33})$. As such, $\hat{\Phi}_1$ does not need to be a non-decreasing function of its argument. In other words, only requirements (34) and (36) need to be satisfied for Φ_1 .

Remark 2.2. Notice that the argument of $\hat{\Phi}_2$ is non-negative. The argument of $\hat{\Phi}_3$ can be expressed in terms the eigenvalues of τ_{NEQ}

$$(I_1^\tau)^2 - 3I_2^\tau = \tau_1^2 + \tau_2^2 + \tau_3^2 - \tau_1\tau_2 - \tau_1\tau_3 - \tau_2\tau_3 \quad (37)$$

which shows that it too is non-negative. This will be important in Section 2.6 where we will construct NODEs such that non-negative inputs lead to non-negative outputs thereby fulfilling the requirement (35).

2.5. Evolution law for the fiber phase

Substituting the free energy function of (14) into the dissipation inequality of (20) yields

$$\underbrace{2I_{4e}^F \frac{\partial \Psi_{NEQ}^F}{\partial I_{4e}^F}}_{\tau_{NEQ}^F} \mathbf{C}_i^F : \mathbf{M} : \frac{1}{2} \dot{\mathbf{C}}_i^F \geq 0 \quad (38)$$

where $\tau_{NEQ}^F \in \mathbb{R}$ denotes the fiber stress. Note that the double contraction in this equation corresponds to the inelastic part of the stretch in the fiber

$$(\lambda_i^F)^2 = I_{4i}^F = \mathbf{C}_i^F : \mathbf{M}. \quad (39)$$

With Eq. (39), the dissipation inequality can be written more concisely as

$$\tau_{NEQ}^F \left(\frac{\dot{\lambda}_i^F}{\lambda_i^F} \right) \geq 0. \quad (40)$$

Eq. (40) requires the specification of a constitutive equation for the evolution of λ_i^F . The model has to satisfy $\text{sign}(\dot{\lambda}_i^F/\lambda_i^F) = \text{sign}(\tau_{NEQ}^F)$. In particular, any monotonically increasing function passing through (0, 0) will satisfy this. In light of this, we propose the following form of the evolution equation for the fiber

$$\left(\frac{\dot{\lambda}_i^F}{\lambda_i^F} \right) = \mathcal{N}(\tau_{NEQ}^F) \quad (41)$$

where $\mathcal{N}$ denotes a NODE. As we show in Section 2.6, a NODE is guaranteed to be monotonic and it can be constructed to pass through the origin.

Note that Eq. (41) is a scalar equation that should be enough to specify the evolution of $\mathbf{C}_i^F, \mathbf{C}_e^F$. One way of closing the system of question is to specify the tensor $\mathbf{C}_i^F$ from λ_i^F , for example

$$\mathbf{C}_i^F = (\lambda_i^F)^2 \mathbf{M}. \quad (42)$$

However, the evolution of the fiber viscoelastic branch can be done without the explicit multiplicative split of the deformation gradient (1). Instead, only the scalar multiplicative split

$$I_4^F = I_{4e}^F I_{4i}^F \quad (43)$$

is needed. Then, Eqs. (38), (41) and (43) fully specify the evolution of the fiber viscoelastic branch without computing $\mathbf{C}_i^F$, $\mathbf{C}_e^F$. To close, the contribution to the stress tensor from the fiber phase is

$$\sigma_{NEQ}^F = \frac{1}{J} \mathbf{F} \mathbf{S}_{NEQ}^F \mathbf{F}^T = \frac{2}{J} \mathbf{F} \frac{\partial \Psi_{NEQ}^F}{\partial \mathbf{C}} \mathbf{F}^T = \frac{2}{J} \mathbf{F} \frac{\partial \Psi_{NEQ}^F}{\partial I_{4e}^F} \frac{\partial I_{4e}^F}{\partial \mathbf{C}} \mathbf{F}^T \quad (44)$$

which reduces to

$$\sigma_{NEQ}^F = \frac{1}{J} \tau_{NEQ}^F \frac{\mathbf{F} \mathbf{M} \mathbf{F}^T}{\mathbf{C} : \mathbf{M}}. \quad (45)$$

2.6. Neural ordinary differential equations

A NODE is a machine learning framework that generalizes some classical models such as recurrent neural networks. We use NODEs to construct monotonic and nonnegative relationships between inputs $\mathbf{x}$ and outputs $\mathbf{y}$. The key idea behind NODEs is the use of an ODE defined on pseudotime ω to compose continuous transformations of a hidden state $\mathbf{h}(\omega)$

$$\frac{d\mathbf{h}(\omega)}{d\omega} = f(\mathbf{h}(\omega), \omega, \boldsymbol{\theta}) \quad (46)$$

where the function $f(\cdot, \cdot, \boldsymbol{\theta})$ is given by a fully connected feed-forward neural network with parameters $\boldsymbol{\theta}$. Given the initial value $\mathbf{h}(0)$ of the hidden state, the output of the model can be obtained by integrating the ODE up to a predetermined pseudotime, say $\omega = 1$

$$\mathbf{h}(1) = \mathbf{h}(0) + \int_0^1 f(\mathbf{h}(\omega), \omega, \boldsymbol{\theta}) d\omega \quad (47)$$

We assign the input to the initial condition $\mathbf{h}(0) = \mathbf{x}$ and the final state to the output $\mathbf{y} = \mathbf{h}(1)$.

From the fundamentals of ODEs we know that no two trajectories of an ODE intersect, because the contrary would indicate that the right hand side $f(\cdot)$ will map the two trajectories to different direction at the point of the intersection, which contravenes the definition of a function. In the 1-dimensional case, this indicates that for any two trajectories $h_1(t)$ and $h_2(t)$

$$h_1(0) \geq h_2(0) \iff h_1(1) \geq h_2(1)$$

$$h_1(0) \leq h_2(0) \iff h_1(1) \leq h_2(1)$$

or more succinctly

$$\begin{aligned} (h_2(1) - h_1(1))(h_2(0) - h_1(0)) &\geq 0 \\ \implies (y_2 - y_1)(x_2 - x_1) &\geq 0 \end{aligned}$$

which shows that the input–output map of a NODE is monotonic.

As mentioned previously, the right hand side of the ODE in (46) is given by a neural network

$$f(x) = \mathbf{W}_n h(\dots \mathbf{W}_2 h(\mathbf{W}_1 x + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_n \quad (48)$$

where $\mathbf{W}_i$ and $\mathbf{b}_i$ are the weights and biases of the i th layer of the neural network. Removing the biases from the neural network results in a new feed-forward operation consisting of a series of multiplications that map $x = 0$ to $f(x) = 0$ provided the activation function satisfies $h(0) = 0$,

$$f(x) = \mathbf{W}_n h(\dots \mathbf{W}_2 h(\mathbf{W}_1 x)) . \quad (49)$$

Together with the monotonicity of NODEs, an architecture such as (49) guarantees that the outputs of the NODEs are non-negative and that the NODEs attain their global minima at $x = 0$, i.e.,

$$f(x) = 0 \implies y = 0 \quad \text{when} \quad x = 0 \quad (50)$$

$$f(x) > 0 \implies y > 0 \quad \text{when} \quad x > 0 \quad (51)$$

While this form of NODEs is suitable, removing all the biases is not necessary. In application we use neural networks of the form

$$f(x) = \mathbf{W}_n h(\dots \mathbf{W}_2 h(\mathbf{W}_1 x)) + \exp(\mathbf{b}_n) \quad (52)$$

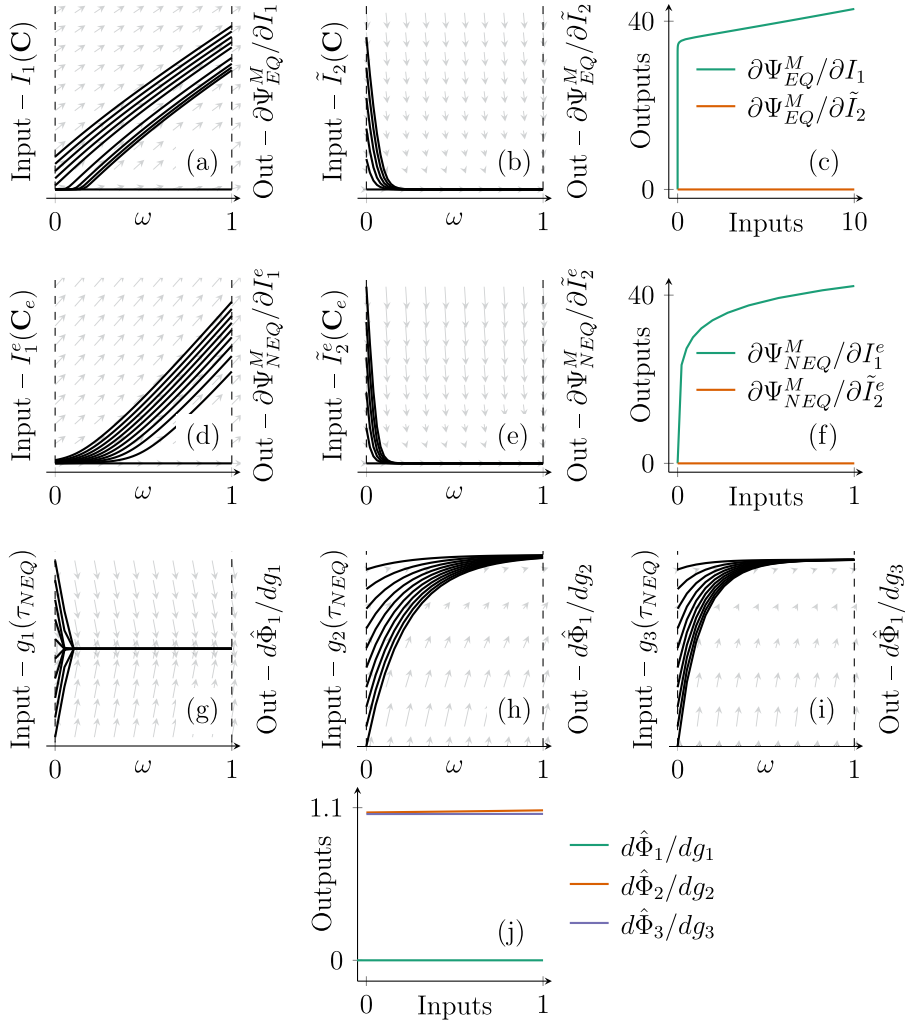


Fig. 2. Example of ODE trajectories generated by NODEs. Trajectories of the ODEs for $\partial \Psi_{EQ}^M / \partial I_1$ (a) and $\partial \Psi_{EQ}^M / \partial \tilde{I}_2$ (b). The NODE defines the underlying vector field shown in gray, which generates the observed trajectories. The corresponding input–output map is monotonic (c). Trajectories of the ODEs for $\partial \Psi_{NEQ}^M / \partial I_1^e$ (d) and $\partial \Psi_{NEQ}^M / \partial \tilde{I}_2^e$ (e) and the resulting input–output maps (f) have a similar response to the equilibrium branch. Trajectories of the ODEs for $d\hat{\Phi}_1/dg_1$ (g), $d\hat{\Phi}_2/dg_2$ (h) and $d\hat{\Phi}_3/dg_3$ (i) with g_i as defined in Section 2.4.1 and the resulting input–output maps (j). In this case the domain of the inputs includes negative non-equilibrium stresses and the outputs are convex functions with minimum at vanishing non-equilibrium stress.

This form of f is identical to the one given in (49) with the exception that the output of the neural network is shifted by a positive value, which adds more flexibility to the neural network. However, all the desirable qualities of NODEs that we have achieved so far are preserved.

In summary, we have shown that the input–output map of a NODE is monotonic. When neural networks of the form (52) are used, non-negative inputs result in non-negative outputs. And finally, the NODEs attain their global minima at $x = 0$. We use NODEs to approximate the functions $\partial \Psi_{EQ} / \partial I_1$, $\partial \Psi_{EQ} / \partial I_2$, $\partial \Psi_{NEQ} / \partial I_1^e$, $\partial \hat{\Phi}_1 / \partial I_1^e$, $\dots$. The resulting ODE trajectories and input–output maps for one model are shown in Fig. 2.

2.7. Integration of the evolution equation

We offer two separate methods for integrating the evolution equations. The first method uses the iterative predictor–corrector algorithm reported in [25] for the matrix phase and the Newton–Raphson solution scheme

reported in [26] for the fiber phase. This solution method is necessary when the deformations are arbitrary. However, the Newton–Raphson algorithms used in this method make it computationally costly and unstable when combined with gradient based optimizers.

The second method can be used for biaxial experiments, i.e., when plane stress conditions hold and $\mathbf{F}$ is diagonal. Under these circumstances the solution of the evolution equation can be expressed as a system of ODEs which makes convergence much faster and stable.

2.7.1. Method 1: Predictor–corrector algorithm for arbitrary $\mathbf{F}$

Matrix phase

The matrix evolution equation provided in (22) is rewritten here

$$-\frac{1}{2}(\mathcal{L}\mathbf{b}_e^M)\mathbf{b}_e^{M-1} = \frac{\partial \Phi}{\partial \boldsymbol{\tau}_{NEQ}^M} . \quad (53)$$

The key idea is to carry out an operator split of the material time derivative of $\mathbf{b}_e$ into elastic predictor (E) and inelastic corrector (I) parts

$$\dot{\mathbf{b}}_e^M = \overline{(\mathbf{F}\mathbf{C}_i^{M-1}\mathbf{F}^T)} = \underbrace{\mathbf{I}\mathbf{b}_e^M + \mathbf{b}_e^M\mathbf{I}^T}_{\text{E}} + \underbrace{\overline{\mathbf{F}\mathbf{C}_i^{M-1}\mathbf{F}^T}}_{\text{I}} \quad (54)$$

where $\mathbf{I} = \dot{\mathbf{F}}\mathbf{F}^{-1}$ is the spatial velocity gradient.

In the elastic predictor step the material time derivative of $\mathbf{C}_i^{M-1}$ is set to zero, giving us

$$(\mathbf{C}_i^{M-1})_{\text{trial}} = (\mathbf{C}_i^{M-1})_{t=t_{n-1}} \implies (\mathbf{b}_e^M)_{\text{trial}} = (\mathbf{F})_{t=t_n}(\mathbf{C}_i^{M-1})_{t=t_{n-1}}(\mathbf{F})_{t=t_n} \quad (55)$$

In the inelastic corrector step the spatial velocity gradient is set to zero, giving us

$$-\frac{1}{2}(\mathcal{L}\mathbf{b}_e^M)\mathbf{b}_e^{M-1} = -\frac{1}{2}\dot{\mathbf{b}}_e^M\mathbf{b}_e^{M-1} = \frac{\partial \Phi}{\partial \boldsymbol{\tau}_{NEQ}^M} \quad (56)$$

$$\implies \dot{\mathbf{b}}_e^M = -2\frac{\partial \Phi}{\partial \boldsymbol{\tau}_{NEQ}^M}\mathbf{b}_e^M \quad (57)$$

Solving this differential equation by exponential mapping and discretizing leads to

$$(\mathbf{b}_e^M)_{t=t_n} \approx \exp\left[-2\Delta t \left(\frac{\partial \Phi}{\partial \boldsymbol{\tau}_{NEQ}^M}\right)_{t=t_n}\right](\mathbf{b}_e^M)_{\text{trial}} \quad (58)$$

But in the isotropic case $\mathbf{b}_e^M$ and $\boldsymbol{\tau}_{NEQ}^M$ are co-axial, which allows us to write this equation in the principal axes

$$\lambda_{e,i}^{M,2} = \exp\left[-2\Delta t \text{diag}\left(\frac{\partial \Phi}{\partial \boldsymbol{\tau}_i}\right)\right](\lambda_{e,i}^{M,2})_{\text{trial}} \quad (59)$$

where $\lambda_{e,i}^M$ are the principal elastic stretches of the matrix phase, $\text{diag}(\partial \Phi / \partial \boldsymbol{\tau}_i)$ is a diagonal matrix consisting of the derivatives of Φ with respect to principal values of $\boldsymbol{\tau}_{NEQ}^M$. Finally, taking the logarithm of both sides gives us

$$\epsilon_{e,i} = -\Delta t \text{diag}\left(\frac{\partial \Phi}{\partial \boldsymbol{\tau}_i}\right) + (\epsilon_{e,i})_{\text{trial}} \quad (60)$$

The nonlinear equation of (60) can be solved using a Newton–Raphson algorithm.

Fiber phase

For an arbitrary $\mathbf{F}$ and corresponding $\mathbf{C} = \mathbf{F}^T\mathbf{F}$, using the scalar split of the fiber stretch in Eq. (43), the ODE in Eq. (41) can be explicitly expressed in terms of λ_i^F ,

$$\left(\frac{\dot{\lambda}_i^F}{\lambda_i^F}\right) = \mathcal{N}(\tau_{NEQ}^F(\lambda_i^F, \mathbf{C}, \mathbf{M})) . \quad (61)$$

Discretizing this differential equation using a backward Euler scheme results in

$$(\lambda_i^F)_{t_n} = (\lambda_i^F)_{t_{n-1}} + \Delta t(\lambda_i^F)_{t_n}\mathcal{N}(\tau_{NEQ}^F((\lambda_i^F)_{t_n}, \mathbf{C}, \mathbf{M})) . \quad (62)$$

This constitutes a nonlinear and implicit equation for λ_i^F at time t_n since $\mathbf{M}$ is constant and $\mathbf{C}$ is prescribed. This equation can be solved for the new value of the internal variable with a numerical solver such as the Newton–Raphson algorithm. The contribution to the stress from the non-equilibrium fiber branch is specified in (45).

2.7.2. Method 2: System of ODEs when $\mathbf{F}$ is diagonal and plane stress conditions hold

Matrix phase

The Lie derivative of $\mathbf{b}_e^M$ in the evolution equation can be written in terms of $\dot{\mathbf{b}}_e^M$ to get

$$\dot{\mathbf{b}}_e^M - (\mathbf{l}\mathbf{b}_e^M + \mathbf{b}_e^M \mathbf{l}^T) = -\frac{\partial \Phi}{\partial \boldsymbol{\tau}_{NEQ}^M} \mathbf{b}_e^M \quad (63)$$

where

$$\mathbf{l} = \dot{\mathbf{F}}\mathbf{F}^{-1} \quad (64)$$

$$\mathbf{b}_e^M = \mathbf{F}_e^M \mathbf{F}_e^{MT} \quad (65)$$

$$\dot{\mathbf{b}}_e^M = \dot{\mathbf{F}}_e^M (\mathbf{F}_e^M)^T + \mathbf{F}_e^M (\dot{\mathbf{F}}_e^M)^T \quad (66)$$

and since $\boldsymbol{\tau}_{NEQ}^M$ is diagonal,

$$\frac{\partial \Phi}{\partial \boldsymbol{\tau}_{NEQ}^M} = \text{diag} \left(\frac{\partial \Phi}{\partial \tau_i} \right). \quad (67)$$

Substituting these back to (63) results in

$$\dot{\mathbf{F}}_e^M \mathbf{F}_e^{MT} + \mathbf{F}_e^M (\dot{\mathbf{F}}_e^M)^T - \left(\dot{\mathbf{F}}\mathbf{F}^{-1} \mathbf{F}_e^M \mathbf{F}_e^{MT} + \mathbf{F}_e^M \mathbf{F}_e^{MT} \mathbf{F}^{-T} \dot{\mathbf{F}}^T \right) = \text{diag} \left(\frac{\partial \Phi}{\partial \tau_i} \right) \mathbf{F}_e^M \mathbf{F}_e^{MT}. \quad (68)$$

where, in the absence of shear terms, every matrix is diagonal, i.e.

$$\mathbf{F} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}, \mathbf{F}_e^M = \begin{bmatrix} \lambda_{e,1}^M & 0 & 0 \\ 0 & \lambda_{e,2}^M & 0 \\ 0 & 0 & \lambda_{e,3}^M \end{bmatrix}, \dot{\mathbf{F}} = \begin{bmatrix} \dot{\lambda}_1 & 0 & 0 \\ 0 & \dot{\lambda}_2 & 0 \\ 0 & 0 & \dot{\lambda}_3 \end{bmatrix}, \dots \quad (69)$$

Then (68) can be written as a system of equations, which after some simplifications results in

$$\dot{\lambda}_{e,1}^M = \left(\frac{\dot{\lambda}_1}{\lambda_1} - \frac{1}{2} \frac{\partial \Phi}{\partial \tau_1} \right) \lambda_{e,1}^M \quad (70)$$

$$\dot{\lambda}_{e,2}^M = \left(\frac{\dot{\lambda}_2}{\lambda_2} - \frac{1}{2} \frac{\partial \Phi}{\partial \tau_2} \right) \lambda_{e,2}^M \quad (71)$$

$$\dot{\lambda}_{e,3}^M = \left(\frac{\dot{\lambda}_3}{\lambda_3} - \frac{1}{2} \frac{\partial \Phi}{\partial \tau_3} \right) \lambda_{e,3}^M. \quad (72)$$

Biaxial loading

Under biaxial loading conditions the out of plane stress component and its time derivative are equal to zero, $\sigma_{33}^M = \dot{\sigma}_{33}^M = 0$. Then, assuming the fibers are in the 1–2 plane,

$$\dot{\sigma}_{33}^M = \frac{d\sigma_{33}^M}{dt} = \frac{\partial \sigma_{33}^M}{\partial \lambda_1} \dot{\lambda}_1 + \frac{\partial \sigma_{33}^M}{\partial \lambda_2} \dot{\lambda}_2 + \frac{\partial \sigma_{33}^M}{\partial \lambda_3} \dot{\lambda}_3 + \frac{\partial \sigma_{33}^M}{\partial \lambda_{e,1}^M} \dot{\lambda}_{e,1}^M + \frac{\partial \sigma_{33}^M}{\partial \lambda_{e,2}^M} \dot{\lambda}_{e,2}^M + \frac{\partial \sigma_{33}^M}{\partial \lambda_{e,3}^M} \dot{\lambda}_{e,3}^M = 0$$

$$\implies \dot{\lambda}_3 = A + B \dot{\lambda}_3^e \quad (73)$$

with

$$A = -\frac{\frac{\partial \sigma_{33}^M}{\partial \lambda_1} \dot{\lambda}_1 + \frac{\partial \sigma_{33}^M}{\partial \lambda_2} \dot{\lambda}_2 + \frac{\partial \sigma_{33}^M}{\partial \lambda_{e,1}^M} \dot{\lambda}_{e,1}^M + \frac{\partial \sigma_{33}^M}{\partial \lambda_{e,2}^M} \dot{\lambda}_{e,2}^M}{\partial \sigma_{33}^M / \partial \lambda_3}$$

$$B = -\frac{\partial \sigma_{33}^M / \partial \lambda_3^e}{\partial \sigma_{33}^M / \partial \lambda_3},$$

substituting this into (72) leads to

$$\dot{\lambda}_{e,3}^M = \frac{\frac{A}{\lambda_3} - \frac{1}{2} \frac{\partial \Phi}{\partial \tau_3}}{1 - \frac{B}{\lambda_3} \lambda_{e,3}^M} \lambda_{e,3}^M \quad (74)$$

Then Eqs. (70), (71) and (74) constitute a system of three ODEs for the three unknowns $\lambda_{e,1}^M$, $\lambda_{e,2}^M$ and $\lambda_{e,3}^M$.

Uniaxial loading

The case of uniaxial loading closely resembles the case of biaxial loading. The only difference being the symmetry $\lambda_2 = \lambda_3$, $\dot{\lambda}_2 = \dot{\lambda}_3, \dots$. This simplifies the Eqs. (70), (71) and (74) to

$$\dot{\lambda}_{e,1}^M = \left(\frac{\dot{\lambda}_1}{\lambda_1} - \frac{1}{2} \frac{\partial \Phi}{\partial \tau_{11}} \right) \lambda_1^e \quad (75)$$

$$\dot{\lambda}_{e,2}^M = \dot{\lambda}_3^e \quad (76)$$

$$\dot{\lambda}_{e,3}^M = \frac{\frac{A'}{\lambda_3} - \frac{1}{2} \frac{\partial \Phi}{\partial \tau_{33}}}{1 - \frac{B'}{\lambda_3} \lambda_{e,3}^M} \lambda_{e,3}^M \quad (77)$$

with

$$A' = -\frac{\frac{\partial \sigma_{33}^M}{\partial \lambda_1} \dot{\lambda}_1 + \frac{\partial \sigma_{33}^M}{\partial \lambda_{e,1}^M} \dot{\lambda}_{e,1}^M}{\partial \sigma_{33}^M / \partial \lambda_2 + \partial \sigma_{33}^M / \partial \lambda_3}$$

$$B' = -\frac{\partial \sigma_{33}^M / \partial \lambda_{e,2}^M + \partial \sigma_{33}^M / \partial \lambda_{e,3}^M}{\partial \sigma_{33}^M / \partial \lambda_2 + \partial \sigma_{33}^M / \partial \lambda_3},$$

Fiber phase

Note that the fiber evolution equation of (61) is an ordinary differential equation in λ_i^F . This equation can be integrated using a numerical integrator of choice thus avoiding an iterative solution scheme. This integration can be performed independent of the matrix phase in biaxial experiments if the fibers are assumed to lie in the 1–2 plane.

2.8. Model calibration and verification

We use both experimental and synthetic training data to train and test our models. The synthetic data is generated using the creep potential used in [25] and neo-Hookean material models for the Helmholtz free energy functions. The creep potential in [25] can be shown to have the following form

$$\Phi_{RG} = \frac{1}{9\eta_V} (I_1^r)^2 + \frac{1}{3\eta_D} ((I_1^r)^2 - 3I_2^r). \quad (78)$$

Note that differentiating this form of the creep potential with respect to τ_{NEQ} twice leads to the familiar isotropic matrix $\mathcal{V}^{-1} = \frac{1}{2\eta_D} (\mathbb{I} - \frac{1}{3} \mathbf{I} \otimes \mathbf{I}) + \frac{1}{9\eta_V} \mathbf{I} \otimes \mathbf{I}$ reported in [25], where η_D and η_V are material parameters and $\mathbb{I}$ is the 4th order identity tensor.

In the case of anisotropic data we use a linear form of the evolution Eq. (20) as proposed in [26]

$$\begin{pmatrix} \dot{\lambda}_i^F \\ \lambda_i^F \end{pmatrix} = \frac{1}{\eta_F} \tau_{NEQ}^F \quad (79)$$

where η_F is a positive parameter that represents the characteristics viscosity of the fiber.

We use the following form of neo-Hookean strain energy functions for Ψ_{EQ}^M and Ψ_{NEQ}^M when generating synthetic training data

$$\Psi = \frac{1}{2} \mu (J^{-2/3} \text{tr} \mathbf{b} - 3) + \frac{1}{2} K (J - 1)^2 \quad (80)$$

and the following form for Ψ_{EQ}^F and Ψ_{NEQ}^F

$$\Psi = k(\exp(I_4 - 1) - 1). \quad (81)$$

We train and test the proposed model with experimental stress data from a variety of materials. Our choice of materials was influenced by the following criteria:

- Include a reasonably large variety of materials
- Focus mainly on soft tissue but include some synthetic materials
- Include both isotropic and anisotropic materials

As a result we train and test the models with experimental data obtained from biaxial stress relaxation experiments on blood clots [29], natural rubber [30], four brain tissue regions [31] and myocardium [32] to train our models. Myocardium is assumed to be anisotropic and the stress predictions are given in two orthogonal directions, while the other materials are assumed to be isotropic in this study.

2.9. Training data-driven models of viscoelasticity

Training a data-driven model of viscoelasticity is computationally expensive. This is due to the fact that the state of stress at any given point depends not only on the observable quantities like the strains, but also on the values of the hidden variables such as $\lambda_{e,1}^M, \lambda_{e,2}^M, \lambda_{e,3}^M$ and λ_i^F in biaxial experiments, or $(\mathbf{C}_i^M)^{-1}, \lambda_i^F$ in the case of arbitrary deformations. This makes training data-driven models time consuming as it entails a full integration of the entire history of stress in every epoch of training. Therefore, even though it was possible to train the NODE-based models directly on data, we found that it was sometimes more efficient to use closed-form models of the form (78) and (80) to *pre-train* the NODE based model. Then the NODEs can be *re-trained* against the actual stress data.

In this study we make use of JAX; a package for high performance numerical calculations in Python to train our models. We use the ADAM optimization algorithm with a typical step size of 0.5×10^{-4} and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ (used by JAX). Training time highly depends on the number of data points being processed, the number of epochs and the size of the neural networks. Usual training times are between 1 and 5 h on an Apple M1 CPU. The code and data for this study are provided in a public repository given in Appendix B to enable independent verification of the findings of this study.

Verification is not as computationally intensive as the training since only a forward-pass is required. Evaluation of stress history for a given set of boundary conditions usually takes less than a minute with a JAX-compiled function.

3. Results

A physically sound formulation of viscoelasticity satisfies the 2nd law of Thermodynamics. To this end, we design creep potentials Φ with appropriate requirements to satisfy this criterion. A demonstration of this is shown in Fig. 3 where, even randomly sampled NODEs are shown to result in non-negative dissipation of energy under arbitrary stresses. On the other hand, feed-forward neural networks (FFNN) fail to satisfy this criterion. We show the dissipation of energy with two FFNN-based creep potentials, one randomly sampled like the NODEs and the other trained against the creep potential of (78). We expect that the randomly sampled FFNN would violate the 2nd law of Thermodynamics, but even the FFNN trained to interpolate an appropriate creep potential violates this criterion especially outside the training region.

We design the creep potential with a very expansive basis to enhance the capability of our model to capture all modes of energy dissipation. In particular, our formulation encompasses a generalization of the creep potentials of the form (78) which are ubiquitously used to describe viscoelastic behavior of isotropic materials. To demonstrate this we generate synthetic stress relaxation data with the creep potential of (78) and neo-Hookean models and train the NODE based model against this data. The resulting plots of stress for various rates of loading and peak stress are shown in Fig. 4. As expected, the NODE based model learns and replicates the training data almost exactly.

Next we train the model with experimental data. As a first step we train the model with experimental stress-relaxation data from human brain tissue [31]. This dataset contains compression relaxation and uniaxial tension data from four regions of human brain; basal ganglia, corpus callosum, corona radiata and cortex. We use the two datasets independently to train and test the NODE and the analytical model. The compression relaxation data and

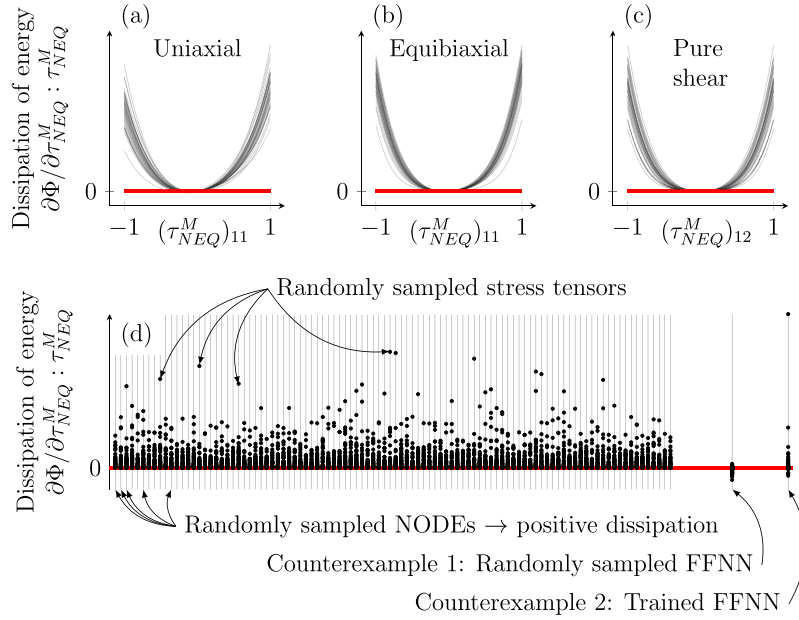


Fig. 3. The NODE modeling framework is guaranteed to result in non-negative dissipation of energy under all deformations. Dissipation of energy under uniaxial, equibiaxial and pure shear conditions with 50 randomly sampled NODEs (a–c), and, dissipation of energy under 50 randomly sampled states of stress (τ_{NEQ}) with 100 randomly sampled NODEs, 1 randomly sampled FFNN and 1 trained FFNN on physically admissible data (d). All points above the red lines correspond to positive dissipation of energy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

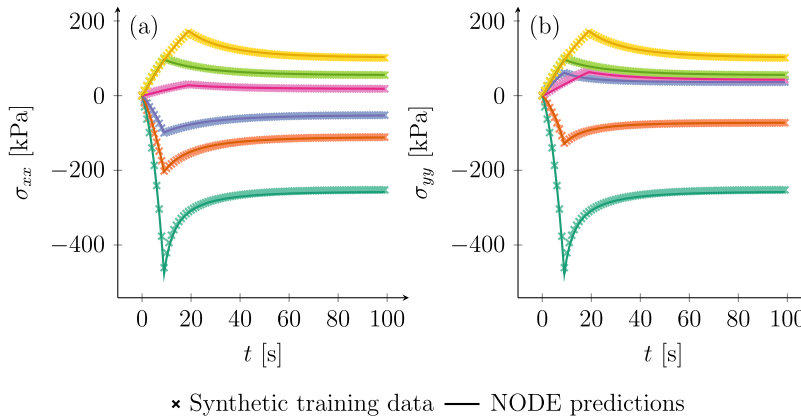


Fig. 4. Results of training the NODEs against synthetic data generating with the analytical model described in (78) and (80).

the predictions of the two models after training are shown in Fig. 5. On average, the NODE based model has a mean absolute error (MAE) of 7.3 Pa, whereas the analytical model has a higher error at 13.3 Pa. The uniaxial tension data consist of loading–unloading curves on the same tissue samples as before and are plotted in Fig. 6. The samples are loaded up to a stretch of 1.1 at a rate of 2 mm/min in this experiment. On average, the NODE has MAE of 4.5 kPa and the analytical model has a 45% higher MAE at 6.5 kPa.

Next, we train the model with data on blood clots [29]. This dataset consists of three stress relaxation experiments on a blood clot sample with peak stresses of 10%, 20% and 30%, respectively. In the experiments a 10 mm × 40 mm blood clot sample with a thickness of 2 mm is stretched along the short edge with a strain rate of 0.001 m/s up to

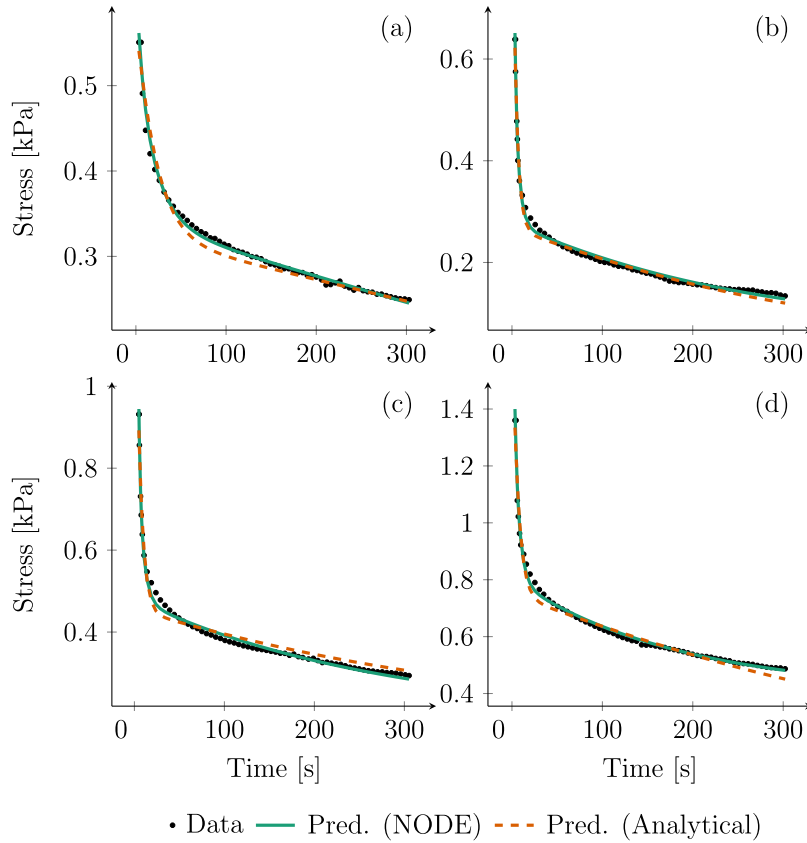


Fig. 5. Compression relaxation data obtained from four regions of human brain; basal ganglia (a), corpus callosum (b), corona radiata (c), cortex (d), and the predictions of the NODE and the analytical model after training.

the desired peak stress and held for 300 s. We train with two of the datasets at a time while testing with the third experiment. The data and predictions of the models after training are shown in Fig. 7. In Fig. 7(a) the training data consists of the two curves with the lower peak force, in Fig. 7(b) it consists of the lower and the upper curves while the middle curve is used for validation and in Fig. 7(c) the lowermost curve is kept for validation while the upper two curves are used for training. As we go from Fig. 7(a) to (c) the training MAE steadily increases from 0.033 mN to 0.056 mN to 0.070 mN. This is partly because the magnitude of training stress is higher as we go from (a) to (c). The rise in *mean relative error* (MRE) is much less pronounced at 0.015, 0.019 and 0.021, respectively. The validation MAE, on the other hand, is lowest for Fig. 7(b) at 0.193 mN compared to MAEs of 0.437 mN and 0.209 mN of the other two cases. This indicates that the model performs better in interpolation than extrapolation. Nevertheless, the prediction in the validation dataset still captures qualitatively and quantitatively the response of the blood clot.

Lastly, we train the isotropic model with data from natural rubber [30]. The data for natural rubber consists of both a stress relaxation and various monotonic compression experiments under different strain rates. We train the model with the stress relaxation data and test with the monotonic compression curves as a validation. The results are shown in Fig. 8. The plots of Fig. 8(a) show that the NODE captures the training data almost exactly, while the analytical model struggles to do so. The MAE of the analytical model is approximately 5 times higher than that of the NODE at 0.071 MPa for the analytical model versus 0.014 MPa of the NODE model. Qualitatively, there is a notable difference in the analytical compared to the NODE predictions in the Training plot of Fig. 8. The analytical model is simply incapable of capturing the response, but this is not a problem for the NODE model.

In validation, NODE has lower errors in the three curves with 0.025/s, 0.075 and 0.225/s strain rates with an MAE between 0.131 and 0.231 MPa whereas the MAE of the analytical model ranges from 0.218 to 0.351 MPa,

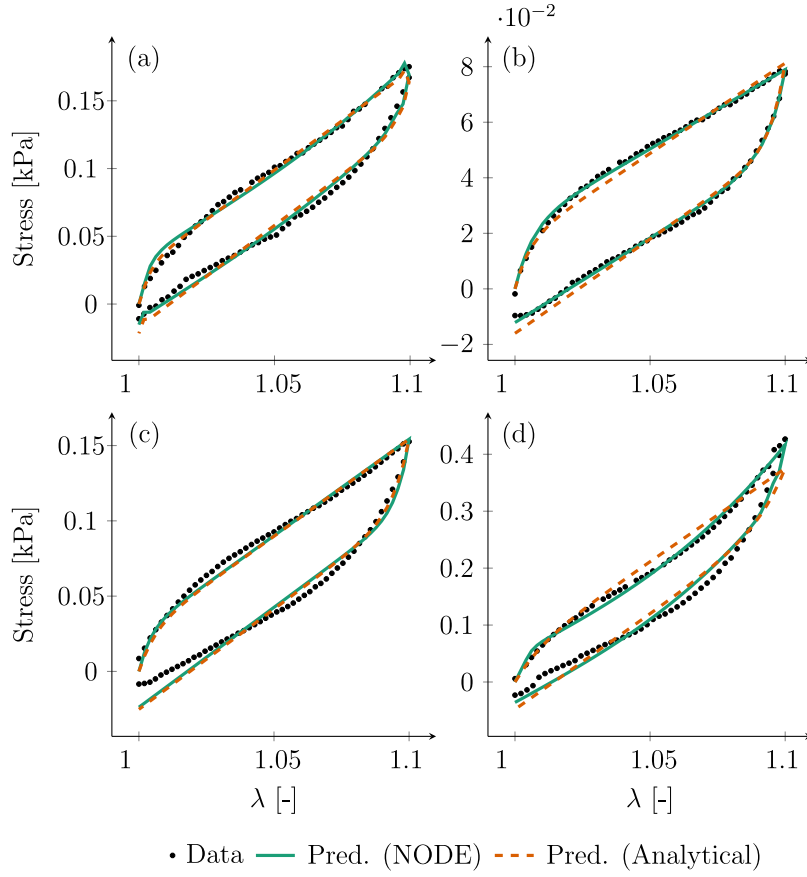


Fig. 6. Uniaxial tension data obtained from four regions of human brain; basal ganglia (a), corpus callosum (b), corona radiata (c), cortex (d), and the predictions of the NODE and the analytical model after training.

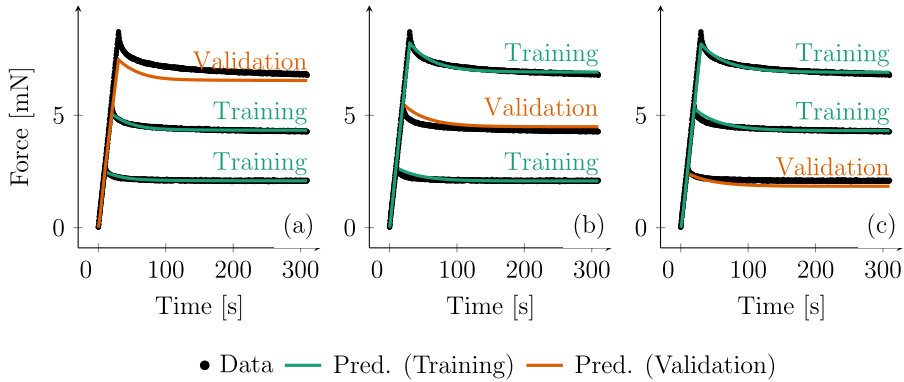


Fig. 7. Stress relaxation data obtained from blood clots and predictions of the model after training.

a much poorer performance. On the 0.96/s curve the analytical model has the lower error at 0.413 MPa vs 0.672 MPa of the NODE. This demonstrates that the model yields acceptably close results even when trained with one type of testing data and tested with another.

Next, we train the model with anisotropic data. For this we use tensile stress relaxation data for human myocardium [32]. Unlike the previous examples, myocardium is a fibrous and anisotropic material and exhibits

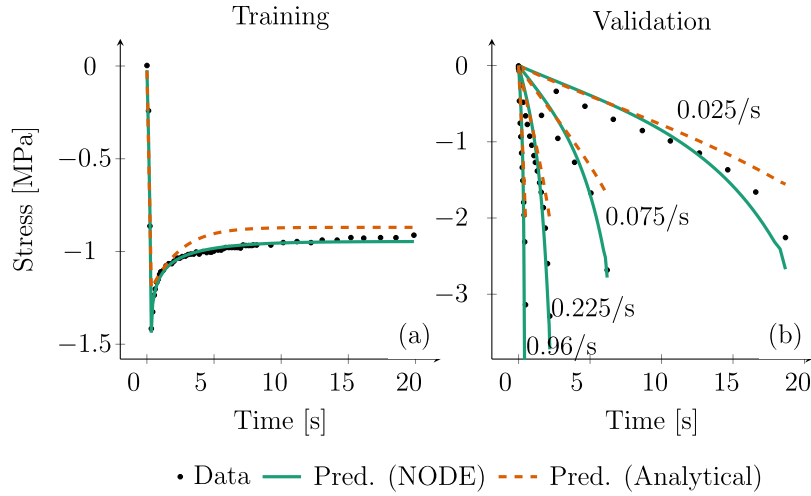


Fig. 8. The NODE and the analytical model were trained with experimental stress relaxation data on natural rubber (a) and tested with various monotone compression curves with different strain rates (b).

significant difference in the stress response in the Mean Fiber Direction (MFD) and Cross Fiber Direction (CFD). We model myocardium assuming the fibers are in-plane, thus the anisotropy direction is specified by a single parameter, the fiber angle θ . The data consists of the average stress of $n = 5$ human myocardium samples in a 5 min long stress relaxation experiment with a 10% peak stress. The results of training both analytical and NODE-based models are shown in Fig. 9. The NODE has MAEs of 0.187 kPa and 0.145 kPa in MFD and CFD, respectively, whereas the analytical model has MAEs of 0.327 kPa and 0.271 kPa.

A summary of the performance of the two models under all experimental training data is provided in Fig. 10. We use mean relative error in this figure to unify the vertical axis of all the different materials. The NODE has the lower error in *all* cases demonstrating the flexibility of the data-driven model.

4. Discussion

We present a fully data-driven, physics-informed model of finite viscoelasticity. Unlike theories of linear or finite linear viscoelasticity, finite viscoelasticity allows for large deformations and large deviations from the thermodynamic equilibrium [25]. This functionality is especially important for soft materials like rubber and soft biological tissue because such materials routinely undergo extremely large deformations in dynamic environments.

The state of stress in a viscoelastic solid can be described using two class of potentials, a creep or dissipation potential and the Helmholtz free energy function [33]. In the classical approach to viscoelasticity both of these potentials are modeled with closed-form expressions. The creep potential of Eq. (78) is nearly the *default* isotropic creep potential of choice for most studies. Some examples of its use for modeling various materials like blood clots, polymers and rubber can be found in [29,34,35]. Some other studies extend the model of [25] to the anisotropic case using an isotropic–anisotropic split of the Helmholtz free energy function [26,36], but even in these cases, linear dissipation is used for the isotropic and anisotropic parts.

The dissipation potential Eq. (78) is a simple function with only two parameters. While it has remained popular, the restrictions of a linear model limit the flexibility of viscoelastic materials that can be modeled. Nonlinear extensions of Eq. (78) in the literature retain the same functional form but replace the viscosity with a function of the deformation [37,38]. We offer a physics-informed, data-driven potential based on NODEs that has much more flexibility, allowing it to capture complex viscoelastic dissipation patterns. In fact, it can be shown that the dissipation potential of (78) is a special case of the proposed potential when the outputs of the NODEs are set to constant values rather than convex functions of their arguments.

The methodology proposed in this paper encompasses a large class of viscoelastic dissipation response. We require that the dissipation potential be a convex function with its global minimum at $\tau_{NEQ}^M = 0$ as stated in

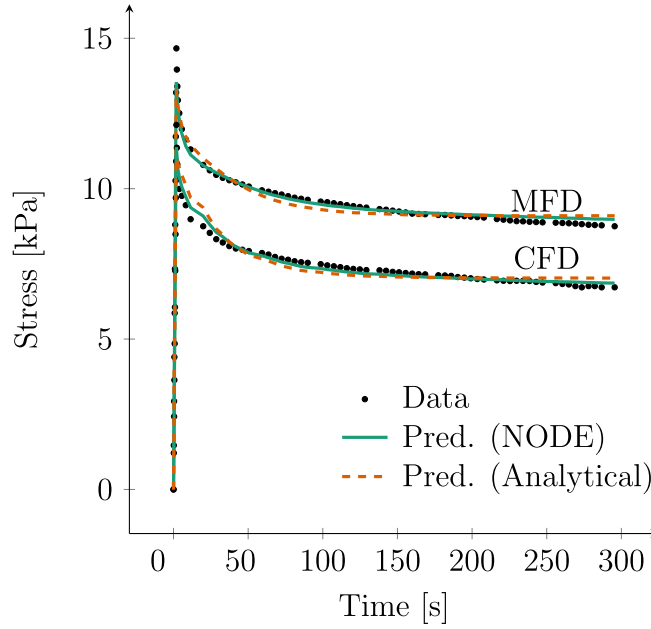


Fig. 9. Stress relaxation data for myocardium in Mean Fiber Direction (MFD) and Cross Fiber Direction (CFD) and the predictions of the two models after training.

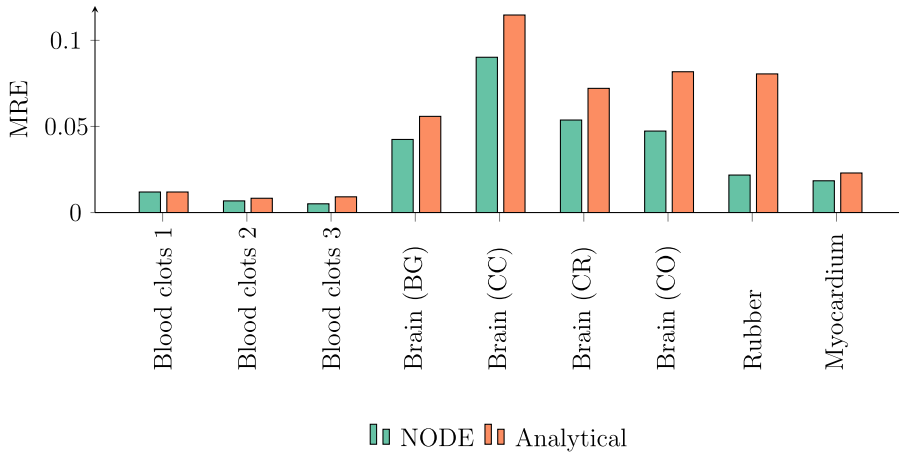


Fig. 10. Training mean relative error (MRE) for all experimental data considered in this study for the NODE and the analytical model.

Section 2.1. However, this is not necessary, as *all* functions f whose partial derivative $\partial f / \partial (\boldsymbol{\tau}_{NEQ}^M)_{ij}$ with respect to the elements $(\boldsymbol{\tau}_{NEQ}^M)_{ij}$ of $\boldsymbol{\tau}_{NEQ}^M$ is positive when $(\boldsymbol{\tau}_{NEQ}^M)_{ij} \geq 0$ and negative when $(\boldsymbol{\tau}_{NEQ}^M)_{ij} \leq 0$ will satisfy the positive dissipation criterion. Nevertheless, convexity is an elegant and mathematically convenient way of expressing functions that satisfy this criterion [33]. Furthermore, we choose to work with only three functions of the invariants of $\boldsymbol{\tau}_{NEQ}$ (g_1 , g_2 and g_3), but there are more possibilities that could be utilized to build even wider classes of constitutive models. Practically speaking, even with just three invariants of $\boldsymbol{\tau}_{NEQ}$ we can already extend the class of viscoelastic behavior way beyond the linear potential Eq. (78), and beyond the nonlinear extensions which keep

Eq. (78) but turn the viscosity into a function of the deformation [38]. The NODE functions can capture arbitrary convex functions (provided enough neurons are used to represent the creep potentials). When trained on various experimental data, the NODE has no problem fitting the data, see Figs. 5–9.

Other theories to model viscoelasticity, in particular for polymers, involves the use of statistical arguments to describe the slip and reorganization of the polymer chains [39]. A similar approach, based on reactive mixtures, has been proposed for biological tissue [40]. Another large class of models can be derived from extensions of the linear viscoelasticity models to large deformations. For example, in [41], non-equilibrium stresses are introduced as internal variables, and each satisfies a linear ordinary differential equation. This class of models result in sums of exponential relaxation functions. To capture non-exponential decay, extensions of [41] using fractional derivatives can result in richer relaxation behavior [42,43].

Regarding the Helmholtz free energy, we have previously demonstrated that NODEs can be used to construct polyconvex and objective strain energy functions for modeling hyperelastic materials [3]. Therefore, for the equilibrium potential in the present work we use almost the same methodology as in [3]. One change in the present formulation is the modeling of compressible and nearly incompressible behavior as opposed to fully incompressible materials. The non-equilibrium free energy function has the same requirements as the equilibrium potential, only with respect to the elastic component of the deformation. Thus, the NODE framework can be used as well for the calculation of the non-equilibrium stresses. Taken together, the proposed methodology is able to describe the entire material behavior using data-driven potentials, without the need of a single *ad hoc* material parameter or constitutive relation.

The flexibility of the method is demonstrated by its ability to learn the viscous behavior of a wide range of biological and synthetic materials including multiple brain tissue regions, rubber, blood clots, and human myocardium. The training error in all cases is consistently lower than the analytical model of [25]. However, there is still some discrepancy between the experiments and the predictions of the model after training. While this may in part be caused by noisy data, it is likely that some of our simplifications and assumptions are preventing the model from learning the material responses better. For instance, we assume the deformations in the experimental data are homogeneous. However, it is likely there are some boundary effects in biaxial test data due to the nature of these experiments [31].

The results of training with data from natural rubber also indicates that the model is capable of predicting stresses in *unseen* types of loading. The framework is suitable for isotropic or anisotropic materials. Importantly, all this flexibility is achieved while polyconvexity and positive energy dissipation are satisfied exactly.

The utility of constitutive material models can be increased by implementation in computational solvers like the finite element method. While we do not attempt to implement the proposed model in a finite element framework in the current study, it remains a future goal. This would require further consideration of the time discretization algorithm and the consistent tangent (interested readers are encouraged to read Section 4 of [25] for a predictor–corrector scheme for the isotropic case).

5. Conclusions

In this paper we introduce a physics-informed data-driven model of general, finite, anisotropic, three-dimensional viscoelasticity based on NODEs. We show that the dissipation potential based on NODEs automatically satisfies the 2nd law of Thermodynamics by leveraging properties of NODEs to build monotonic functions. Polyconvexity of the Helmholtz free energy is also ensured with a NODE framework. We train the proposed model with various experimental datasets and demonstrate the flexibility of the method in capturing complex viscoelastic responses of a wide class of materials including rubbers, blood clots, brain, and myocardium. We anticipate that this work will enable the modeling and simulation of arbitrary viscoelastic materials without the burden of selecting closed-form constitutive equations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is in a public repository.

Acknowledgments

This work was supported by NIAMS, USA award R01AR074525, NSF CMMI, USA award 1916668.

Appendix A. Proof of convexity of I_1^2 and $I_1^2 - \alpha I_2$

A function f of a symmetric 3×3 matrix $\mathbf{M}$ can alternatively be expressed as a function of the 6 independent elements of $\mathbf{M}$ as

$$f(\mathbf{M}) = f(M_{11}, M_{22}, M_{33}, M_{12}, M_{13}, M_{23}). \quad (\text{A.1})$$

The function f is convex in the elements of $\mathbf{M}$ if the Hessian matrix of f is positive semi-definite. The Hessian matrix of f is given by

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial M_{11}^2} & \frac{\partial^2 f}{\partial M_{11} \partial M_{22}} & \frac{\partial^2 f}{\partial M_{11} \partial M_{33}} & \frac{\partial^2 f}{\partial M_{11} \partial M_{12}} & \cdots \\ \frac{\partial^2 f}{\partial M_{22} \partial M_{11}} & \frac{\partial^2 f}{\partial M_{22}^2} & \frac{\partial^2 f}{\partial M_{22} \partial M_{33}} & \frac{\partial^2 f}{\partial M_{22} \partial M_{12}} & \cdots \\ \frac{\partial^2 f}{\partial M_{33} \partial M_{11}} & \frac{\partial^2 f}{\partial M_{33} \partial M_{22}} & \frac{\partial^2 f}{\partial M_{33}^2} & \frac{\partial^2 f}{\partial M_{33} \partial M_{12}} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (\text{A.2})$$

$\mathbf{H}$ is positive semi-definite if all of its eigenvalues are non-negative.

A.1. I_1^2

The function $f(\mathbf{M}) = (I_1(\mathbf{M}))^2 = (\text{tr} \mathbf{M})^2$ can be expressed as

$$f(\mathbf{M}) = f(M_{11}, M_{22}, M_{33}) = (M_{11} + M_{22} + M_{33})^2 \quad (\text{A.3})$$

Then it is readily seen that the Hessian of f is

$$\mathbf{H} = \begin{bmatrix} 2 & 2 & 2 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.4})$$

which has eigenvalues $\lambda_1 = 6, \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = 0$. Therefore I_1^2 is convex.

A.2. $I_1^2 - \alpha I_2$

The second principal invariant I_2 of a matrix $\mathbf{M}$ can be expressed in terms of the elements of $\mathbf{M}$ as

$$I_2(\mathbf{M}) = M_{11}M_{22} + M_{11}M_{33} + M_{22}M_{33} - M_{12}^2 - M_{13}^2 - M_{23}^2 \quad (\text{A.5})$$

using this in $f(\mathbf{M}) = I_1^2 - \alpha I_2$ results in

$$\begin{aligned} f(\mathbf{M}) &= (M_{11} + M_{22} + M_{33})^2 \\ &\quad - \alpha(M_{11}M_{22} + M_{11}M_{33} + M_{22}M_{33} - M_{12}^2 - M_{13}^2 - M_{23}^2) \\ &= M_{11}^2 + M_{22}^2 + M_{33}^2 + (2 - \alpha)(M_{11}M_{22} + M_{11}M_{33} + M_{22}M_{33}) \\ &\quad + \alpha(M_{12}^2 + M_{13}^2 + M_{23}^2). \end{aligned}$$

Then the Hessian of f is given as

$$\mathbf{H} = \begin{bmatrix} 2 & 2 - \alpha & 2 - \alpha & 0 & 0 & 0 \\ 2 - \alpha & 2 & 2 - \alpha & 0 & 0 & 0 \\ 2 - \alpha & 2 - \alpha & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\alpha \end{bmatrix}. \quad (\text{A.6})$$

This matrix has eigenvalues

$$\lambda_1 = 2(3 - \alpha) \quad (\text{A.7})$$

$$\lambda_2 = \alpha \quad (\text{A.8})$$

$$\lambda_3 = \alpha \quad (\text{A.9})$$

$$\lambda_4 = 2\alpha \quad (\text{A.10})$$

$$\lambda_5 = 2\alpha \quad (\text{A.11})$$

$$\lambda_6 = 2\alpha. \quad (\text{A.12})$$

Then it can be seen that all the eigenvalues are non-negative for $0 \leq \alpha \leq 3$. Therefore $I_1^2 - 3I_2$ is convex for $0 \leq \alpha \leq 3$.

Appendix B. Supplementary material

All data, model parameters and code associated with this study are available in a public Github repository at <https://github.com/tajtac/nvisco>.

References

- [1] H. Dal, K. Açıkgöz, Y. Badienia, On the performance of isotropic hyperelastic constitutive models for rubber-like materials: a state of the art review, *Appl. Mech. Rev.* 73 (2) (2021) 020802, <http://dx.doi.org/10.1115/1.4050978>.
- [2] Y. Xiang, D. Zhong, S. Rudykh, H. Zhou, S. Qu, W. Yang, A review of physically based and thermodynamically based constitutive models for soft materials, *J. Appl. Mech.* 87 (11) (2020).
- [3] V. Tac, F.S. Costabal, A.B. Tepole, Data-driven tissue mechanics with polyconvex neural ordinary differential equations, *Comput. Methods Appl. Mech. Engrg.* 398 (2022) 18, <http://dx.doi.org/10.1016/j.cma.2022.115248>.
- [4] V. Tac, V.D. Sree, M.K. Rausch, A.B. Tepole, Data-driven modeling of the mechanical behavior of anisotropic soft biological tissue, *Eng. Comput.* 38 (5) (2022) 4167–4182, <http://dx.doi.org/10.1007/s00366-022-01733-3>.
- [5] M. Liu, L. Liang, W. Sun, A generic physics-informed neural network-based constitutive model for soft biological tissues, *Comput. Methods Appl. Mech. Engrg.* 372 (2020) 113402, <http://dx.doi.org/10.1016/j.cma.2020.113402>.
- [6] P. Weber, J. Geiger, W. Wagner, Constrained neural network training and its application to hyperelastic material modeling, *Comput. Mech.* 68 (5) (2021) 1179–1204, <http://dx.doi.org/10.1007/s00466-021-02064-8>.
- [7] V. Tac, K. Linka, F. Sahli-Costabal, E. Kuhl, A.B. Tepole, Benchmarks for physics-informed data-driven hyperelasticity, 2023, [arXiv:2301.10714](https://arxiv.org/abs/2301.10714).
- [8] N.N. Vlassis, R. Ma, W. Sun, Geometric deep learning for computational mechanics Part I: Anisotropic hyperelasticity, *Comput. Methods Appl. Mech. Engrg.* 371 (2020) 113299, <http://dx.doi.org/10.1016/j.cma.2020.113299>.
- [9] Y. Leng, V. Tac, S. Calve, A.B. Tepole, Predicting the mechanical properties of biopolymer gels using neural networks trained on discrete fiber network data, *Comput. Methods Appl. Mech. Engrg.* 387 (2021) 114160, <http://dx.doi.org/10.1016/j.cma.2021.114160>.
- [10] B. Amos, L. Xu, J.Z. Kolter, Input convex neural networks, in: 34th International Conference on Machine Learning, Sydney, Australia, 2017, p. 10.
- [11] R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, Neural ordinary differential equations, 2019, [arXiv:1806.07366](https://arxiv.org/abs/1806.07366), [cs, stat].
- [12] D.K. Klein, M. Fernández, R.J. Martin, P. Neff, O. Weeger, Polyconvex anisotropic hyperelasticity with neural networks, *J. Mech. Phys. Solids* 159 (2022) 104703, <http://dx.doi.org/10.1016/j.jmps.2021.104703>.
- [13] P. Chen, J. Guillemot, Polyconvex neural networks for hyperelastic constitutive models: A rectification approach, *Mech. Res. Commun.* 125 (2022) 103993, <http://dx.doi.org/10.1016/j.mechrescom.2022.103993>, <https://www.sciencedirect.com/science/article/pii/S0093641322001252>.
- [14] F. As'ad, P. Avery, C. Farhat, A mechanics-informed artificial neural network approach in data-driven constitutive modeling, *Internat. J. Numer. Methods Engrg.* 123 (12) (2022) 2738–2759, <http://dx.doi.org/10.1002/nme.6957>.
- [15] K. Linka, S.S. Pierre, E. Kuhl, Automated model discovery for human brain using Constitutive Artificial Neural Networks, 2022, <http://dx.doi.org/10.1101/2022.11.08.515656>.
- [16] K. Linka, E. Kuhl, A new family of Constitutive Artificial Neural Networks towards automated model discovery, 2022, [arXiv:2210.02202](https://arxiv.org/abs/2210.02202).
- [17] M. Flaschel, S. Kumar, L. De Lorenzis, Unsupervised discovery of interpretable hyperelastic constitutive laws, *Comput. Methods Appl. Mech. Engrg.* 381 (2021) 113852, <http://dx.doi.org/10.1016/j.cma.2021.113852>.
- [18] P. Thakolkaran, A. Joshi, Y. Zheng, M. Flaschel, L. De Lorenzis, S. Kumar, NN-EUCLID: Deep-learning hyperelasticity without stress data, 2022, [arXiv:2205.06664](https://arxiv.org/abs/2205.06664).
- [19] H. Salahshoor, M. Ortiz, Model-free data-driven viscoelasticity in the frequency domain, 2022, [arXiv:2205.06674](https://arxiv.org/abs/2205.06674).
- [20] S. Wang, P. Perdikaris, Long-time integration of parametric evolution equations with physics-informed DeepONets, *J. Comput. Phys.* 475 (2023) 111855, <http://dx.doi.org/10.1016/j.jcp.2022.111855>.
- [21] K. Xu, A.M. Tartakovsky, J. Burghardt, E. Darve, Learning viscoelasticity models from indirect data using deep neural networks, *Comput. Methods Appl. Mech. Engrg.* 387 (2021) 114124, <http://dx.doi.org/10.1016/j.cma.2021.114124>.

- [22] G. Chen, Recurrent neural networks (RNNs) learn the constitutive law of viscoelasticity, *Comput. Mech.* 67 (3) (2021) 1009–1019, <http://dx.doi.org/10.1007/s00466-021-01981-y>.
- [23] E. Marino, M. Flaschel, S. Kumar, L. De Lorenzis, Automated identification of linear viscoelastic constitutive laws with euclid, 2022, arXiv preprint [arXiv:2212.10969](https://arxiv.org/abs/2212.10969).
- [24] M. Flaschel, S. Kumar, L. De Lorenzis, Automated discovery of generalized standard material models with euclid, 2022, arXiv preprint [arXiv:2211.04453](https://arxiv.org/abs/2211.04453).
- [25] S. Reese, S. Govindjee, A theory of finite viscoelasticity and numerical aspects, *Int. J. Solids Struct.* 35 (26–27) (1998) 3455–3482, [http://dx.doi.org/10.1016/S0020-7683\(97\)00217-5](http://dx.doi.org/10.1016/S0020-7683(97)00217-5).
- [26] T. Nguyen, R. Jones, B. Boyce, Modeling the anisotropic finite-deformation viscoelastic behavior of soft fiber-reinforced composites, *Int. J. Solids Struct.* 44 (25–26) (2007) 8366–8389, <http://dx.doi.org/10.1016/j.ijsolstr.2007.06.020>.
- [27] S. Hartmann, P. Neff, Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility, *Int. J. Solids Struct.* 40 (11) (2003) 2767–2791, [http://dx.doi.org/10.1016/S0020-7683\(03\)00086-6](http://dx.doi.org/10.1016/S0020-7683(03)00086-6).
- [28] S.P. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK; New York, 2004.
- [29] M.K. Rausch, G.P. Sugerman, S. Kakaletsis, B. Dortdivanlioglu, Hyper-viscoelastic damage modeling of whole blood clot under large deformation, *Biomech. Model. Mechanobiol.* 20 (5) (2021) 1645–1657, <http://dx.doi.org/10.1007/s10237-021-01467-z>.
- [30] A. Amin, M. Alam, Y. Okui, An improved hyperelasticity relation in modeling viscoelasticity response of natural and high damping rubbers in compression: Experiments, parameter identification and numerical verification, *Mech. Mater.* 34 (2) (2002) 75–95, [http://dx.doi.org/10.1016/S0167-6636\(01\)00102-8](http://dx.doi.org/10.1016/S0167-6636(01)00102-8).
- [31] S. Budday, G. Sommer, J. Haybaeck, P. Steinmann, G. Holzapfel, E. Kuhl, Rheological characterization of human brain tissue, *Acta Biomater.* 60 (2017) 315–329, <http://dx.doi.org/10.1016/j.actbio.2017.06.024>.
- [32] G. Sommer, A.J. Schriefel, M. Andrä, M. Sacherer, C. Viertler, H. Wolinski, G.A. Holzapfel, Biomechanical properties and microstructure of human ventricular myocardium, *Acta Biomater.* 24 (2015) 172–192, <http://dx.doi.org/10.1016/j.actbio.2015.06.031>.
- [33] A. Kumar, O. Lopez-Pamies, On the two-potential constitutive modeling of rubber viscoelastic materials, *Comptes Rendus Mecanique* 344 (2) (2016) 102–112.
- [34] B. Kleuter, A. Menzel, P. Steinmann, Generalized parameter identification for finite viscoelasticity, *Comput. Methods Appl. Mech. Engrg.* 196 (35–36) (2007) 3315–3334, <http://dx.doi.org/10.1016/j.cma.2007.03.010>.
- [35] P.J. Loew, B. Peters, L.A. Beex, Rate-dependent phase-field damage modeling of rubber and its experimental parameter identification, *J. Mech. Phys. Solids* 127 (2019) 266–294, <http://dx.doi.org/10.1016/j.jmps.2019.03.022>.
- [36] H. Liu, G.A. Holzapfel, B.H. Skallerud, V. Prot, Anisotropic finite strain viscoelasticity: Constitutive modeling and finite element implementation, *J. Mech. Phys. Solids* 124 (2019) 172–188, <http://dx.doi.org/10.1016/j.jmps.2018.09.014>.
- [37] A. Amin, A. Lion, S. Sekita, Y. Okui, Nonlinear dependence of viscosity in modeling the rate-dependent response of natural and high damping rubbers in compression and shear: Experimental identification and numerical verification, *Int. J. Plast.* 22 (9) (2006) 1610–1657.
- [38] J. Zhou, L. Jiang, R.E. Khayat, A micro–macro constitutive model for finite-deformation viscoelasticity of elastomers with nonlinear viscosity, *J. Mech. Phys. Solids* 110 (2018) 137–154.
- [39] F.J. Vernerey, R. Long, R. Brighenti, A statistically-based continuum theory for polymers with transient networks, *J. Mech. Phys. Solids* 107 (2017) 1–20.
- [40] G.A. Ateshian, Viscoelasticity using reactive constrained solid mixtures, *J. Biomech.* 48 (6) (2015) 941–947.
- [41] G.A. Holzapfel, J.C. Simo, A new viscoelastic constitutive model for continuous media at finite thermomechanical changes, *Int. J. Solids Struct.* 33 (20–22) (1996) 3019–3034.
- [42] W. Zhang, A. Capilnasiu, G. Sommer, G.A. Holzapfel, D.A. Nordsletten, An efficient and accurate method for modeling nonlinear fractional viscoelastic biomaterials, *Comput. Methods Appl. Mech. Engrg.* 362 (2020) 112834.
- [43] A. Freed, K. Diethelm, Fractional calculus in biomechanics: a 3d viscoelastic model using regularized fractional derivative kernels with application to the human calcaneal fat pad, *Biomech. Model. Mechanobiol.* 5 (4) (2006) 203–215.