A NEURAL ORDINARY DIFFERENTIAL EQUATION FRAMEWORK FOR MODELING INELASTIC STRESS RESPONSE VIA INTERNAL STATE VARIABLES

R.E. Jones, A.L. Frankel, & K.L. Johnson

1Sandia National Laboratories, Livermore, California 94551, USA
2Sandia National Laboratories, Albuquerque, New Mexico 87185, USA

*Address all correspondence to: R.E. Jones, Sandia National Laboratories, Albuquerque, New Mexico 87185, USA; Tel.: +1 925 294 4744; Fax: +1 925 294 3231, E-mail: rjones@sandia.gov

Original Manuscript Submitted: 1/25/2022; Final Draft Received: 5/2/2022

We propose a neural network framework to preclude the need to define or observe incompletely or inaccurately defined states of a material in order to describe its response. The neural network design is based on the classical Coleman-Gurtin internal state variable theory. In the proposed framework the states of the material are inferred from observable deformation and stress. A neural network describes the flow of internal states and another represents the map from internal state and strain to stress. We investigate tensor basis, component, and potential-based formulations of the stress model. Violations of the second law of thermodynamics are prevented by a constraint on the weights of the neural network. We extend this framework to homogenization of materials with microstructure with a graph-based convolutional neural network that preprocesses the initial microstructure into salient features. The modeling framework is tested on large datasets spanning inelastic material classes to demonstrate its general applicability.

KEY WORDS: neural ordinary differential equations, graph convolutional neural networks, material modeling, internal state variables

1. INTRODUCTION

Relative to the certainty of the balance laws and the accuracy of the numerical methods, the fidelity of a constitutive model is typically the weak point of a predictive simulation for a solid. This is in part due to the fact that solid material behavior is complex and frequently influenced by microstructure. Generally it can span a range of regimes with elastic, viscous, damage, and/or plastic characteristics, as well as have anisotropy and/or temperature dependence. This complexity often leads to models with significant discrepancies, which can be expensive to evaluate.

The mechanical response of many technologically relevant solid materials is inelastic, which means these materials are dissipative and their stress and other aspects of their response depend on the deformation history. This history dependence is typically modeled with (a) hereditary integral formulations where kernels encode fading memory (Coleman and Noll, 1961; Lubliner, 1969), or (b) with differential models with evolving state variables (Coleman and Gurtin, 1967;
Examples of the first type, which result in Volterra integral equations, are the nonlinear relaxation kernel of traditional viscoelasticity models (Coleman and Noll, 1961; Lubliner, 1969) and the memory kernel of Mori-Zwanzig formalism employed primarily in fluid mechanics (Chorin et al., 2000; Li et al., 2014; Parish and Duraisamy, 2017) and particle systems (Adelman and Doll, 1976; Wagner and Liu, 2003). The second route, where the present state of the material is expanded beyond observable deformation (Coleman and Gurtin, 1967), is arguably more extensible since it has been applied to a much wider class of solid material response than viscoelasticity (Horstemeyer and Bammann, 2010; Kratochvil and Dillon, 1969; McDowell, 2005). In this second category the observable state is augmented with additional state variables that evolve via ordinary differential equations (ODEs). This is in contrast to hypoelasticity (Truesdell and Noll, 2004) where stress, itself, is governed by an ODE. Although oftentimes physically motivated, such as molecular vibrations, order parameters, or dislocation densities, these state variables are typically hidden from direct observation and may be incomplete or inappropriate to accurately predict the observable behavior. Generally these type of models are referred to as internal state variable (ISV) models and follow from Coleman and Gurtin’s seminal work (Coleman and Gurtin, 1967). Coleman and Gurtin acknowledged that the delineation between observable and hidden can be arbitrary but gave the perspective that the hidden internal variables do not have to be ultimately observable; they must merely represent the process of interest phenomenologically. In fact, Kröner (1963) stated that a complete description of microstructure arrangement is unnecessary as long as the macroscale representation is complete in the sense that it is predictive. This fundamental concept is at the core of model reduction and homogenization where macroscale processes depend on summary statistics of microscopic states. The proposed inferred state variable neural ODE (ISV-NODE) model is motivated by this perspective.

1.1 Internal State Variable Theory

The applications of the ISV constitutive framework to solid material behavior are myriad. Closely following the publication of the Coleman-Gurtin ISV theory (Coleman and Gurtin, 1967), Kratochvil and Dillon (1969) and Rice (1971) framed elastoplasticity in the ISV framework, Bhandari and Oden (1973) provided a viscoplastic ISV theory, and Perzyna (1986) developed an ISV model of ductile failure based on evolution of porosity (what is now known as a damage model). Later Reese and Govindjee (1998) provided a particularly illuminating example of the application of ISV theory. They translated and generalized the standard rheological model for viscoelasticity to finite deformations using an ISV plasticity framework based on the work of Simo et al. in finite deformation associative elastoplasticity (Simo and Miehe, 1992) and viscoplasticity (Simo, 1992). Reese and Govindjee employed the now ubiquitous multiplicative decomposition of deformation gradient and an additive decomposition of the free energy potential, and demonstrated the formulation with oscillatory loading shear and steady loading creep tests.

The breadth and depth of ISV constitutive modeling developments has motivated a number of reviews of the field. In 1983, Germain et al. (1983) published a summary and survey ISV elastoviscoplasticity theory. Later, McDowell (2005) published an in-depth and insightful review of the ISV framework for incorporating experimental data of irreversible/path-dependent behavior and gave an example application to metal viscoplasticity. That work provided a detailed discussion of the multiscale aspects of the theory and physically motivated mesoscale mechanics in general, including the assumption of locality for a representative volume element (RVE) in terms of the correlation length of relevant microscopic fields and other statistical aspects. Horstemeyer and Bammann (2010) wrote a historical survey of the development of ISV theory.
tying it to the inception of irreversible thermodynamics by Onsager (1931a,b) and later Eckart (1940, 1948). Horstemeyer and Bammann also made a comprehensive review of the applications of ISV in solid mechanics which ranged across plasticity, viscosity/creep, and damage, as well as multiphase, composite, biological and particulate materials.

The ISV theory is clearly an effective framework but still suffers from the primary difficulty of traditional constitutive modeling: discrepancies and model-form errors due to calibrating preconceived functional forms and assumed state variables. In fact, there is no generally applicable prescription for determining appropriate/representative variables that quantify internal states (McDowell, 2005). ISV approaches oftentimes result in overly complicated models that are costly to evaluate in large-scale simulation. In contrast, machine learning models, like other general statistical models, are largely data-driven and applicable through a general and extensible methodology. Furthermore, current efforts (Karniadakis et al., 2021; Lee and Carlberg, 2019; Ling et al., 2016; Linka et al., 2021; Masi et al., 2021; Raissi et al., 2019; Wang et al., 2020) to hybridize machine learning models for physics with the principles used to develop traditional models are enabling the best of both approaches: efficient, expressive data-driven and traditional physics-informed modeling.

1.2 Neural Networks

There have been considerable developments using neural networks (NN) (Tsoi and Back, 1997; Yu et al., 2019) for modeling dynamical systems and related tasks, such as sequence learning for natural language parsing (Lipton et al., 2015) and normalizing flows used for generative modeling (Kobyzev et al., 2020). Most of these developments fall into the category of recurrent NNs (RNNs), such as long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and the gated recurrent unit (GRU) (Cho et al., 2014) architectures, which have memory to capture the causality of most time signals. Based on the connection between the ResNet (He et al., 2016), which predict differences between states, and traditional discretizations of ODEs, new architectures have been recently proposed. Chen et al. (2018) introduced the neural ordinary differential equation (NODE) where the right-hand-side driving term of a system of ODEs is represented by a general multilayer perceptron (MLP). This formulation incorporates the time-step scaling of the dynamics, which is missing in RNNs. Closely following this development, Dupont et al. (2019) produced a major improvement with the augmented neural ODE (ANODE). In an ANODE additional degrees of freedom are introduced to untangle the flows of the observed data; this effectively alleviates the necessity for paths to cross in order to represent some datasets. This is the main enhancement of the original NODE, since trajectory crossing is disallowed in nonautonomous ODEs by the Picard–Lindelöf theorem (Arnold, 1973; Coddington and Levinson, 1955). This expansion of the state space of the dynamical system enabled learning complex dynamics with simpler flows, apparently generalized better, achieved lower losses with fewer parameters, and was more stable in training. Rackauckas et al. (2020) developed a similar framework, called universal differential equations (UDEs), based on universal approximation properties and focused on physical applications. In fact, NODEs and the like have been shown to have a universal approximation property (Lu et al., 2019; Rubel, 1981; Teshima et al., 2020), like the related MLPs (Hornik et al., 1989; Scarselli and Tsoi, 1998). They also have a Bayesian extension (Dandekar et al., 2020) that embeds uncertainty in the model predictions. The ODE-based NN architectures are adaptable, generalizable, and have the distinct appeal of resembling classical methods of representing and simulating time-continuous dynamical systems.
There have been a number of developments specific to applying NN to modeling evolving physical systems. Fu et al. (2020) and Qin et al. (2021) have been particularly active in this area. Fu et al. (2020) created a generalized Langevin model resulting from a discrete and finite Mori-Zwanzig memory kernel with a ResNet architecture, and Qin et al. (2021) reframed the time input of nonautonomous differential equations into that of a time-parameterized general forcing input to model a general class of evolutionary behavior. A number of other developments have focused on control applications. Drgona et al. (2020) developed a RNN-type model of buildings subject to real world heating requirements for model-based predictive feedback control and compared it to alternative strategies. Drgona and co-workers (Drgona et al., 2020) also made a spectral analysis of deep networks with common activation functions and categorized their dynamic stability.

Physics-constrained approaches have received considerable attention and development with contributions such as physics informed neural networks (Lagaris et al., 1998; Raissi et al., 2019). Specific to material modeling, the tensor basis neural network (TBNN) (Frankel et al., 2020b; Jones et al., 2018; Ling et al., 2016; Peters et al., 2020) was developed to embed the symmetries imposed by material frame indifference (equivariance). As an adaptation of classical representation theory to neural network architecture, a TBNN stress, or any other tensorial output, is represented by a finite sum of trainable NN coefficient functions of the scalar invariants of the inputs paired with the tensor basis elements of the inputs. Two approaches have been introduced to construct a TBNN: (a) an implicit TBNN, which is trained directly to the stress output and the coefficients are inferred (Ling et al., 2016), and (b) an explicit TBNN, where the coefficients are solved from the stress and strain invariants and then the network is trained to these coefficients (Frankel et al., 2020b). The TBNN approach where symmetry is embedded in the formulation, as opposed to learned, has been shown to reduce the training burden (Ling et al., 2016).

A few notable developments are particular to inelastic solid behavior. Ghaboussi and Sidarta’s seminal work (Ghaboussi and Sidarta, 1998) introduced the use of neural networks to modeling the mechanical response of solid materials using an incremental stress formulation akin to hypoelasticity. Lefik and Schrefler (2003) used a similar formulation and focused on elastoplastic hysteresis. Later, Xu et al. (2020) developed a NN model of viscoelastic behavior and trained it via optimization constrained by solutions to the boundary value problem of the application. Their model also resembles hypoelasticity formulated by stress increment as a function of current stress and strain and was demonstrated on simplified geomechanics problems. Masi et al. (2021) also developed a hypoelastic NN representation based on modeling increments in stress, a dissipation function, and traditionally defined internal state variables. Recently, Logarzo et al. (2021) used a RNN to model homogenized inelastic response of an elastic-plastic matrix with a single stiff elastic inclusion. In contrast, Vlassis and Sun (2020) embedded a key feature of traditional plasticity theory with their implementation of a NN yield function. They demonstrated superior performance to alternative RNN formulations that lack an explicit yield surface with their model. Also relevant to this work, Teichert et al. (2019) demonstrated that thermodynamic potentials can be inferred by NN trained on derivative data, such as stress.

1.3 An Internal State Motivated Neural Network

The goal of this work is to create a representation suitable to modeling a wide class of dissipative materials that obeys physical principles, such as frame invariance and the second law of thermodynamics, and that treats hidden states, like those related to damage, in a flexible and data-driven manner. In this representation the state of the material is inferred from data, not defined by
preconceived quantities, such as plastic-strain and damage, that require unambiguous definition and measurement. As in the ISV framework and our previous work (Jones et al., 2018), the proposed model has interpretable components that are analogs to (a) the traditional flow rule, which evolves internal state based on strain loading, and (b) a stress model, mapping state to observable stress. The evolution of the inferred state vector is handled with traditional time integrators compatible with current large-scale simulators. Furthermore, the state space can be built with complexity appropriate for the available data and physical process, as with the ANODE. We expand the treatment of homogeneous materials, where the state vector is completely inferred, to those with microstructure where the relevant structural features are discovered with a hybrid convolutional neural network (CNN)-RNN architecture devised in previous work (Frankel et al., 2020a, 2021, 2019; Vlassis et al., 2020). In contrast to previous developments, in this work we couple a graph-based CNN (GCNN) applied directly to the unstructured mesh data (Frankel et al., 2021) to the ISV-NODE dynamical model. The GCNN processes the initial microstructure to a latent set of initial features that augment the state vector in our ANODE-like ISV framework. In essence the GCNN informs the initial latent vector characterizing the hidden state, which was assumed to be a constant (zero) in the absence of microstructural information.

In the following sections we develop and demonstrate that the proposed ISV-NODE framework can (a) follow the fundamental thermodynamic prescriptions on the flow of (hidden) internal state variables for a variety of complex loading paths and materials, and (b) infer a latent space of internal variables effective in predicting dissipative behavior either predicated on initial microstructure or merely from assumed initial conditions. In Section 2 we outline Coleman-Gurtin internal state variable theory and connect it to augmented neural ODEs in Section 3 to put the proposed architecture in context. We apply the ISV-NODE framework to modeling viscoelastic and elastoplastic homogeneous materials and similar materials with microstructure. The training methodology is described in Section 4, and the multitude of training data is discussed in Section 5. The training data are divided into two categories: (a) homogeneous material point data used to test the ISV-NODE ability to provide an effective representation of a dissipative material and adhere to thermodynamic principles, and (b) inhomogeneous RVE data used to demonstrate that the ISV-NODE, when coupled with a convolutional neural network, is an effective representation of the homogenized response of materials with microstructure. With these in hand we explore the qualities of the ISV-NODE and demonstrate its performance in both point response and RVE homogenization modes in Section 6. We conclude with a discussion of the developments, open questions, and future work in Section 7.

2. THEORY

Our goal is to create a constitutive modeling framework for the stress response of inelastic materials based solely on thermodynamic principles and observable, measurable quantities, namely: (a) the material motion \( \mathbf{x} = \chi(\mathbf{X}, t) \), from reference position \( \mathbf{X} \) to current position \( \mathbf{x} \) as a function of time, and (b) the Cauchy stress \( \mathbf{T} \). The classical Coleman and Gurtin (1967) internal state variable (ISV) theory postulates that general inelastic response can be described with a few canonical response functions which are dependent on the current state of a deformation measure, temperature, and a collection of additional, internal state variables. The Helmholtz free energy \( \Psi \) determines stress and entropy through its partial derivatives with respect to deformation and temperature, while the heat flux and evolution of the ISVs need to be specified independently from the free energy.
In this work, we restrict the developments to isothermal processes. In this case, the Coleman-Gurtin ISV framework reduces to a stress response,

\[ S = \partial_E \Psi \big|_h = \dot{S}(E, h), \]  

(1)

and a flow of the internal state variables,

\[ \dot{h} = \hat{f}(E, h), \]  

(2)

which are functions of observable strain \( E \) and the additional internal state variables \( h \). Many traditional models of inelastic solids (Gurtin et al., 2010; Silhavy, 2013; Simo and Hughes, 2006; Truesdell and Noll, 2004) fit into this framework. Here we have chosen the second Piola-Kirchhoff stress \( S \) as the stress measure, which is related to the Cauchy stress \( T \) by \( \det(F)/T = FSF^T \), and the Lagrange strain, \( E = (1/2) (F^T F - I) \), where \( F = \partial_X \chi(X, t) \) is the deformation gradient. These stress and strain measures are energetic duals and inherently invariant to superposed rigid motion or changes of material coordinate frame. Both arguments, \( E \) and \( h \), are present in both response functions by Truesdell’s principle of equipresence (Truesdell, 1959; Truesdell and Toupin, 1960), which can be considered as a design principle for constitutive models. Truesdell and Toupin (1960) postulate a number of general principles for constitutive models including dimensional independence, spatial and material invariances, and other symmetries which we also follow.

The second law of thermodynamics is particularly relevant to inelastic materials since it constrains allowable stress responses. The Clausius-Duhem inequality (Silhavy, 2013) reduces to the rate of change of the free energy being bounded by the stress power,

\[ \dot{\Psi} \leq S \cdot \dot{E}, \]  

(3)

in the isothermal case. This principle, together with the definition of stress \( S \), Eq. (1), and the energy balance for isothermal process,

\[ \dot{\Psi} = S \cdot \dot{E} + \partial_h \Psi \cdot f, \]  

(4)

constrain the flow of the internal state variables:

\[ \partial_h \Psi \cdot f \leq 0. \]  

(5)

This result, and the definition of stress \( S \) as the partial derivative of the free energy \( \Psi \) with respect to the strain \( E \) with the internal variables \( h \) fixed, implies the ISVs characterize inelastic behavior and, in particular, the change of state associated with irreversible deformation.

Other than Eq. (5), the Coleman-Gurtin theory does not provide general prescriptions for the form of the function \( f \) driving the evolution of the ISVs. We augment the arguments of \( f \) with the strain rate \( \dot{E} \):

\[ \dot{h} = \hat{f}(E, \dot{E}, h), \]  

(6)

so that the rate dependence does not need to be derived from the history \( E(t) \) and allows the flow rule to have an explicit sense of loading direction. We chose to have state variables \( h \) to be nominally in the reference configuration, so the material time derivative is an appropriate rate.

Since many other objective rates (Haupt and Tsakmakis, 1989) are available, this selection is effectively a model form choice, like our choice of stress and strain measures; however, it is not a particularly restrictive one. The set of internal state variables, \( h \), does not necessarily have
a tensorial character (McDowell, 2005). Lacking general principles motivating more complex choices, we assume the hidden state $h$ can be characterized by a (nontensorial) collection of scalars in the reference configuration. This implies that the state evolution $\dot{h}$ is driven by the invariants, $I$, and the invariants of the strain loading, $\mathcal{I}(E, \dot{E})$:

$$\dot{h} = \hat{f}(\mathcal{I}(E, \dot{E}), h).$$

(7)

Initial conditions for the hidden states $h(t = 0)$ can be set to 0 lacking other information. If additional data are available, for example, a fixed state such as temperature or an initial microstructure, the state $h$ can be augmented with this information as we will show in the next section.

Stress model $\hat{S}(E, h)$ has a number of valid formulations. A potential-based model $\hat{\Psi}$ follows directly from the Coleman-Gurtin theory:

$$S = \partial E \hat{\Psi}(E, h).$$

(8)

This requires inferring the potential $\Psi$ from the observable stress and deformation data. A tensor basis formulation,

$$S = \sum_i \hat{\sigma}_i(\mathcal{I}(E, h))B_i,$$

(9)

is an alternative that has the possible advantages of being a smoother model and having simpler dependence on the strain since the basis $B_i$ embeds some of the functional dependence. In this form, $\hat{\sigma}_i$ are functions of the invariants of $E$ and $h$ and the basis $B_i$ is constructed from $E$ (and possibly $\dot{E}$; see Appendix A for details). Without access to the potential this formulation cannot enforce the Clausius-Duhem inequality, Eq. (5), directly. This will be discussed further in the next section. A component-based representation,

$$S = \sum_{(ij)} \hat{s}_{(ij)}(E, h)e_{(ij)},$$

(10)

presents a second alternative, where $e_{(ij)} \equiv (1/2)(e_i \otimes e_j + e_j \otimes e_i)$ are the symmetric Cartesian basis dyads in the reference configuration. This formulation is similar to the tensor basis formulation; however, the basis is fixed and the components $\hat{s}_{(ij)}$ are a function of the components of strain, not its invariants.

3. ARCHITECTURE

As discussed in the preceding sections, the primary shortcoming of the general ISV modeling framework is that it is not clear what the internal state variables $h$ are nor what dimensionality of this set should be. To alleviate this shortcoming we developed a data-driven neural network (NN) analog to the ISV framework based on an enhanced neural ODE (NODE).

3.1 The ANODE

To put our developments in context we first outline the basic form of the augmented neural ODE (ANODE) (Dupont et al., 2019). Briefly, its objective is to train a model of a time-dependent process $y = y(t)$ assuming its true form is given by an ODE:

$$\dot{y} = f(y, t).$$

(11)
Training is based on observations of \( y \) over time \( t \) and \( f(y, t) \) is modeled with a multilayer perceptron (MLP), a deep, densely connected feed-forward NN (Goodfellow et al., 2016; Rosenblatt, 1961). Each layer of the MLP translates its inputs \( x \) to outputs with a parameterized affine transformation followed by a nonlinear transformation:

\[
x_{i+1}(t) = a(W_i x_i(t) + b_i),
\]

where \( W_i \) is the trainable weight matrix of the \( i \)-th layer, \( b_i \) is the trainable bias, and \( a \) is the pre-selected nonlinear activation function applied elementwise. The input \( x_0(t) = [y(t), t, h(t)]^T \) to the NN approximation of the dynamics, \( \mathbf{NN}(x_0) \), is formed from the observable state vector \( y \) augmented with time \( t \) and additional hidden trajectories \( h \). This augmentation, not found in the original neural ODE (Chen et al., 2018), allows trajectories to be simpler, in the sense that they do not cross each other or themselves, by expanding the dimensionality of the output space. The dimension of \( h \) determines the size of latent space needed to represent the process \( y(t) \) and, in part, the complexity of the approximation. The other primary contributor to the complexity of the representation is the number of layers \( n \) in the MLP. The output of the final layer of \( \mathbf{NN}(x_0(t)) \) is \( x_n = [f, 1, h] \). Typically the application of the nonlinearity \( a \) is omitted for this final layer to allow for linear mixing of the product of previous layers. This function can then be integrated with a standard numerical integrator, for instance, with an explicit midpoint rule,

\[
x(t + \Delta t) = x(t) + \Delta t, \mathbf{NN}(x(t + 1/2\Delta t), t + 1/2\Delta t),
\]

where \( x(t + 1/2\Delta t) = x(t) + 1/2\Delta t, \mathbf{NN}(t) \). As opposed to a RNN, in this architecture the scaling with time is explicitly accounted for by the embedded time integrator. In general, where no additional information is available about the initial state, the initial conditions for \( h(t = 0) \) are set to zero. Training is done with standard back-propagation (Robbins and Monro, 1951; Rumelhart et al., 1986) or via adjoint methods (Chen et al., 2018). Note that both the original NODE formulation (Chen et al., 2018) and the augmented NODE (Dupont et al., 2019) in effect augment the apparent state vector \( y \) with time \( t \) so that the MLP approximates both the dependence of \( f \) on \( y \) and \( t \) with time-invariant weights \( \{W_i\} \) and biases \( \{b_i\} \).

### 3.2 The ISV-NODE

Clearly the flow rule, Eq. (7), resembles the nonautonomous ODE, Eq. (11), the ANODE is built on; however, we need a number of enhancements to adapt this methodology to the ISV framework. With reference to the schematic in Fig. 1, first we replace explicit dependence on time \( t \) in \( \tilde{f} \) in Eq. (11) with dependence on the strain loading. Specifically we use the invariants of the strain and strain rate \( \mathcal{I}(\mathbf{E}, \dot{\mathbf{E}}) \), which can be found in Appendix A, as inputs to form a representation of \( \tilde{f} \) from Eq. (2). In addition to following the ISV framework and the Truesdell-Toupin design principles, this modification enables \( f \) to readily generalize to arbitrary loading paths. Second, we add an additional MLP to transform the output of the integration of the flow rule, i.e., the hidden state \( h \), and the strain \( \mathbf{E} \) to the observable stress \( \mathbf{S} \) following Eq. (1). The inclusion of strain \( \mathbf{E} \) in the arguments of \( \hat{\Psi} \) promotes \( h = 0 \) when no dissipation is occurring. We will also introduce a penalization of the related constraint, Eq. (5), to the loss function, discussed in the next section.

The potential-based Eq. (8), tensor basis Eq. (9), and component-based Eq. (10) variants each lead to slightly different architectures. In the potential-based variant, illustrated in Fig. 1(a), the potential \( \hat{\Psi} \) is represented by the second MLP, and its derivative with respect to strain produces...
A Neural Ordinary Differential Equation Framework

9

\[ E(t) \]

\[ \text{invariants } \{ I(E, \dot{E}) \} \]

\[ \text{flow } h = f(h, I(E, \dot{E})) \]

\[ \text{integration } h = h + \Delta t f \]

\[ \text{potential } \Psi = \Psi(h, E) \]

\[ \text{derivative } S = \partial E \hat{\Psi} \]

\[ \text{output } S(t) \]

(a)

\[ E(t) \]

\[ \text{invariants } \{ I(E, \dot{E}) \} \]

\[ \text{flow } h = f(h, I(E, \dot{E})) \]

\[ \text{integration } h = h + \Delta t f \]

\[ \text{coefficients } \sigma_i = \hat{\sigma}_i(h, I(E)) \]

\[ \text{sum } S = \sum \sigma_i B^i \]

\[ \text{output } S(t) \]

(b)

FIG. 1: ISV-NODE architecture: (a) potential-based stress, (b) tensor basis stress formulations. Colors denote: green: input; orange: trainable (state evolution and state-to-output) MLPs; gray: nontrainable operations; yellow: output. Note the size of \( h \) is user selected and \( h(t=0) = 0 \) initial conditions are given to the integrator.

The ISV-NODE architecture has a number of hyperparameters: \( N_f \), the number of layers in the state-evolution MLP \( NN_f \); \( N_S \), the number of layers in the stress MLP \( NN_S \); and \( N_h \), the number of hidden ISVs. The choice of the number of hidden states \( N_h \) is of primary importance and determines, together with the sizes of \( I(E, \dot{E}) \), \( I(E) \), and \( S \), the width of the MLPs. Following the stability analysis of Drgona et al. (2020) and some preliminary studies, we selected the \( C^1 \)-smooth exponential linear unit (ELU) as the activation \( a \) for both MLPs.

As in the ANODE we use \( h(t=0) = 0 \) initial conditions for the hidden state variables in the absence of data-derived state information. We will demonstrate this is effective in modeling homogeneous materials. For heterogeneous materials we inform the hidden state \( h \) with the available microstructural fields. Following Frankel et al. (2021, 2019) and with reference to Fig. 2(a), we reduce initial microstructural fields with a convolutional neural network (CNN) to latent features correlated with the output and augment the material state, \( h \), with these microstructural features. The CNN component that reduces the initial microstructure to structural features is trained simultaneously with the NODE component. These inferred features take the form of additional initial conditions for \( h \), unlike in our previous RNN-based formulations where this information was a constant input to the RNN. As in Frankel et al. (2021) we employ a graph-based convolutional unit applied directly to the data on the unstructured discretization of the microstructure.

Figure 2(b) illustrates the internal structure of the convolutional unit that processes the fields describing the initial microstructure \( \phi(X) \) into relevant features \( \varphi \). It consists of a number of
FIG. 2: ISV-NODE architecture with microstructure input $\phi(X)$: (a) overall architecture for a potential-based formulation, and (b) details of the convolutional unit. Colors denote: green: input; red: convolution; orange: trainable (state evolution and state-to-output) MLPs; gray: nontrainable operations; yellow: output. The convolution unit is circumscribed by an orange background and the NODE unit is circumscribed by a blue background. Note the output of the convolutional unit, features $\varphi$, is concatenated with the initial conditions of the user-selected state variables $h$.

Graph convolutions applied to $\Phi$; each operates on nearest neighbor elements as defined by the unstructured mesh used to compute the response. Each convolutional layer is endowed with $N_{\text{filters}}$ independent filters. After $N_{\text{convolution}}$ convolutions, global pooling is applied to reduce the output to dimension $N_{\text{filters}}$. This output is fed in $N_{\text{dense}}$ densely connected layers and ultimately a final linear mixing layer, as in a MLP. For this version of the convolutional unit the size of the structural features is equal to the number of filters $N_{\text{filters}}$. Frankel et al. (2021) give more details of GCNN architecture.

The ISV-NODE was implemented with TensorFlow (Abadi et al., 2021) and the GCNN with Spektral (Grattarola, 2021).

4. TRAINING

As can be seen in Fig. 1 the inputs to the fundamental ISV-NODE are the strain $E(t)$ and its rate $\dot{E}(t)$ corresponding to the loading of the sample. These are collected from the simulations used as a data source at discrete times $\{t_i = i\Delta t, i = 0, N_{\text{steps}}\}$ where $\Delta t$ is fixed per sample but varies across the ensemble of data. A fixed time step per trajectory was chosen as a convenience for collecting data from the data generating simulator; it is not a restriction on the ANODE. For the state evolution NN these data are preprocessed into the corresponding joint invariants $I(E(t_i), \dot{E}(t_i))$. The output stress $S(t_i)$ to be compared with the model output $\hat{S}(t_i)$ in the loss function is obtained from the Cauchy stress of the simulations. In the case of the heterogeneous simulations the output stress is the volume-averaged Cauchy stress.

To facilitate training the data are rescaled to be $O(1)$. The ensemble of inputs $\{(E(t_i), \dot{E}(t_i))_k, k = 1, N_{\text{samples}}\}$ and outputs $\{(S(t_i))_k, k = 1, N_{\text{samples}}\}$ are rescaled by their

Journal of Machine Learning for Modeling and Computing
respective maximums over the ensemble, e.g., $S \rightarrow [1/(SS)]S$ where $SS = \max |(S_{ij})_{k}(t_n)|$. Since the data are sufficiently centered no shift of the mean was done. The corresponding invariants are rescaled by powers of $s_{trE} = \max |\text{tr} E|$ and $s_{sE} = \max |\text{tr} \dot{E}|$, e.g., $\text{tr}(E^{2}\dot{E}) \rightarrow [1/(s_{sE}^{2} s_{trE})] \text{tr}(E^{2}\dot{E})$, for consistency. The $N_{\text{samples}}$ sample trajectories were randomly split into training, testing, and validation tranches of complete trajectories in a 70/10/20 proportion.

We employed a standard mean square error (MSE) loss function,

$$L = \frac{1}{N_{\text{samples}}} \sum_{i=1}^{N_{\text{samples}}} \sum_{j=1}^{N_{t}} ||S_{i}(t_j) - \hat{S}(E_{i}(t_j), \dot{E}(t_j))||^2,$$  \hspace{1cm} \left(14\right)$$

applied to the stress data $S$ and model response $\hat{S}$. [The root mean squared error (RMSE) is $\sqrt{L}$.]

To enforce the dissipation constraint imposed by the second law, we augment the loss $L$ with a penalty,

$$L_{\varepsilon} = L + \varepsilon g,$$  \hspace{1cm} \left(15\right)$$

where $\varepsilon$ is the penalty hyperparameter and $g$ is the constraint function. The form of constraint is determined by the stress model. For the potential-based formulation of the stress, the constraint takes the form

$$g = R(\partial_{h}\Psi \cdot f),$$  \hspace{1cm} \left(16\right)$$

based on inequality, Eq. (5), where $R$ is the Macauley bracket, which is identical in form to the ReLU activation function, $R(x) = x$ if $x > 0$ and $R(x) = 0$ otherwise. In the tensor basis and component-based stress formulations, the potential is not available so we chose to match the expended power,

$$g = ||(S - \hat{S}) \cdot \dot{E}||^2,$$  \hspace{1cm} \left(17\right)$$

as the constraint. Note with a tensor basis formulation, Eq. (9) allows Eq. (17) to be written in terms of the coefficient functions $\hat{\sigma}_{i}$. Furthermore, Jones et al. (2018) showed how to construct a strictly dissipative flow rule for assumed plastic strain ISV using selective powers of the input invariants; however, we have left adapting this constraint to the generality of the ISV-NODE for the future.

The training scheme to minimize the loss, $L_{\varepsilon}$, involved a standard stochastic gradient descent algorithm, Adam (Kingma and Ba, 2014), with an initial learning rate of 0.001 and batch size 64 for the homogeneous data. Given the size of the microstructure inputs, the models for the heterogeneous data were trained with a batch size of 1, as in Frankel et al. (2021). Sequential training was used to ensure convergence, where the model was trained to the initial steps of the evolution $N_{\text{steps}}$, then to incrementally more of the evolution till a target portion of the training evolution was used. A typical schedule was $N_{\text{steps}} = \{40, 80, 120, 200\}$ where early stopping based on a target accuracy was employed. The target accuracy was increased as more of the evolution was included in the training. Note that a parallel approach where time-batched training sets would be used independently (Gunther et al., 2020) was not feasible since the internal state $h$ evolves and we assume $h = 0$ only at $t = 0$.

5. DATA

We use training data drawn from two general classes of inelastic material behavior: viscoelasticity and elastoplasticity. As in other work, typical, well-proven models of this behavior serve as stand-ins for sufficient experimental data as data fusion and other techniques are developed.
In addition to creating response data for homogeneous systems undergoing homogeneous deformations, we generated response data for large-scale representative volumes with pores or inclusions. For these data our goal was to predict the homogenized response based on the initial microstructure. The Sierra simulation suite (Stewart and Edwards, 2020) was used to generate all training and testing data. Table 1 provides a summary of each dataset.

5.1 Viscoelastic Material

For a traditional model of a common engineering polymeric material, we employed a universal polymer model (UPM) (Adolf et al., 2009) for Sylgard 184 silicone, a lightly cross-linked, flexible, isotropic elastomer (Long and Brown, 2017). The UPM is a viscoelastic model of the hereditary integral type:

\[
T = (K - K_\infty) \mathbf{I} \int_0^t (f_K(t - s) \text{tr} \dot{\epsilon}(s)) \, ds + K_\infty \mathbf{I} \text{tr} \epsilon, \\
+ 2(G - G_\infty) \int_0^t (f_G(t - s) \text{dev} \dot{\epsilon}(s)) \, ds + 2G_\infty \text{dev} \epsilon, \tag{18}
\]

based on a bulk\((K)\)/shear\((G)\) split. The strain measure \(\epsilon\) is given by the integration of the unrotated rate of deformation \(\mathbf{D} = (1/2)(\nabla_x v + \nabla^T_x v)\),

\[
\epsilon = \int_0^t \mathbf{R}^T(s) \mathbf{D}(s) \mathbf{R}(s) \, ds, \tag{19}
\]

where \(\mathbf{R}\) is the rotation tensor from the polar decomposition of the deformation gradient \(\mathbf{F}\). The relaxation kernels \(f_K\) and \(f_G\) are represented with Prony series with 20 relaxation times ranging from 1 \(\mu\)s to 3160 s. The instantaneous bulk and shear moduli, \(K = 920\) MPa and \(G = 0.362\) MPa, and equilibrium bulk and shear moduli, \(K_\infty = 920\) MPa and \(G_\infty = 0.084\) MPa, and all other parameters are given in Long and Brown (2017).

**Table 1:** Homogeneous (point) and heterogeneous (RVE) material datasets used for training and testing the ISV-NODE. Here \(\mathbf{E}(t)\) is strain evolution, \(\mathbf{S}(t)\) is stress evolution, and \(\phi(X)\) is the material phase field in the reference configuration. Note the homogeneous material data were sampled over uniaxial, biaxial, and shear modes in cyclic loading over a range of timescales, while the heterogeneous material data were monotonic uniaxial loading for RVEs consisting of \(\approx\) 10,000–100,000 cells.

<table>
<thead>
<tr>
<th>Type</th>
<th>Samples</th>
<th>Inputs</th>
<th>Outputs</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homogeneous</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPM viscoelastic</td>
<td>8000</td>
<td>(\mathbf{E}(t), \dot{\mathbf{E}}(t))</td>
<td>(\mathbf{S}(t))</td>
<td>4000</td>
</tr>
<tr>
<td>J2 elastoplastic</td>
<td>8000</td>
<td>(\mathbf{E}(t), \dot{\mathbf{E}}(t))</td>
<td>(\mathbf{S}(t))</td>
<td>4000</td>
</tr>
<tr>
<td>Heterogeneous</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Porous elastoplastic</td>
<td>1120</td>
<td>(\ddot{\mathbf{E}}<em>{11}(t), \dot{\mathbf{E}}</em>{11}(t), \phi(X))</td>
<td>(\ddot{\mathbf{S}}_{11})</td>
<td>400</td>
</tr>
<tr>
<td>Porous viscoelastic</td>
<td>1120</td>
<td>(\ddot{\mathbf{E}}<em>{11}(t), \dot{\mathbf{E}}</em>{11}(t), \phi(X))</td>
<td>(\ddot{\mathbf{S}}_{11})</td>
<td>400</td>
</tr>
<tr>
<td>Filled viscoelastic</td>
<td>1120</td>
<td>(\ddot{\mathbf{E}}<em>{11}(t), \dot{\mathbf{E}}</em>{11}(t), \phi(X))</td>
<td>(\ddot{\mathbf{S}}_{11})</td>
<td>400</td>
</tr>
</tbody>
</table>
5.2 Elastoplastic Material

Aluminum was chosen as a typical elastic-plastic material as represented by a J2 model (Lubliner, 2008). The stress \( S \) is given by a linear elastic rule:

\[
S = C : E_e,
\]

where “:” is a double inner product that allows the fourth-order elastic modulus tensor \( C \) to map the elastic strain \( E_e \) to the stress \( S \). The elastic logarithmic strain is derived from a multiplicative split of the deformation gradient \( F = F_e F_p \). For an isotropic material, like common aluminum, the components of \( C \) reduce to

\[
[C]_{ijkl} = \frac{E}{(1 + \nu)} \left( \frac{\nu}{1 - 2\nu} \delta_{ij} \delta_{kl} + \frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \right),
\]

which depend only on Young’s modulus \( E = 59.2 \) GPa and Poisson’s ratio \( \nu = 0.33 \).

The plastic flow is derived from the von Mises yield condition,

\[
\sigma_{vm}(S) - \bar{\sigma}(\varepsilon_p) \leq 0,
\]

which limits the elastic regime to a convex region in stress space and offsets the elastic strain \( E_e \) from the total strain. Here \( \sigma_{vm} = \sqrt{(3/2)s \cdot s} \) is the von Mises stress where \( s = S - \text{tr}(S)I \) is the deviatoric part of \( S \), and \( \varepsilon_p \) is the equivalent plastic strain, which is a measure of the accumulated plastic strain computed from the plastic velocity gradient \( D_p \):

\[
\varepsilon_p = \int_0^t \sqrt{\frac{2}{3} D_p(s) : D_p(s)} ds.
\]

The yield limit \( \bar{\sigma} \) is given by a Voce hardening law,

\[
\bar{\sigma} = Y + H (1 - \exp(-\alpha \varepsilon_p)),
\]

with the parameters: initial yield \( Y = 200.0 \) MPa, hardening \( H = 163.6 \) MPa, and saturation exponent \( \alpha = 73.3 \).

5.3 Homogeneous Response

For the homogeneous material dataset used to test the ISV-NODE framework, we created response histories using three common experimental deformation modes: uniaxial, biaxial, and simple shear. The training data consist of sinusoidal displacement loading such that the stress-strain hysteresis loops become steady after a number of cycles depending on loading frequency. Figure 3 shows the hysteresis data for both the UPM and J2 materials.

For each loading, mode frequencies were sampled uniformly on a log scale and amplitudes were sampled uniformly on a linear scale. For the J2 model the frequency was fixed at 0.001 Hz since it is rate independent. The domain was a cube represented by a single element and minimum boundary condition for static determinacy were applied. For the uniaxial mode, boundary displacement of the \( +x_1 \) boundary was prescribed as

\[
u_1(t) = A_1 \sin(\omega_1 t) e_1.
\]
FIG. 3: Stress-strain hysteresis of homogeneous UPM and J2 data for randomly selected samples across all loading modes. (a) Viscoelastic UPM and (b) elastic-plastic J2. Note normal (left) and shear (right) components of the same trajectories are plotted with the same colors.

For the UPM model $A_1 \in [0.1, 0.5]$ and $\omega_1 \in [10^5, 10^6]$, and for the J2 model $A_1 \in [0.01, 0.03]$ and $\omega_1 = 0.001$. For the biaxial mode normal displacement on the $+x_1$ boundary was prescribed as

$$ u_1(t) = A_1 \sin(\omega_1 t), $$ (26)

and on the $+x_2$ boundary,

$$ u_2(t) = A_2 \sin(\omega_2 t), $$ (27)

where $A_1, A_2 \in [-0.2, 0.2]$ and $\omega_1, \omega_2 \in [10^5, 10^6]$ for the UPM data, and $A_1, A_2 \in [-0.02,
0.02] and $\omega_2/\omega_1 \in [-1, 1]$ for the J2 data. For the simple shear mode the tangential displacement on the $+x_2$ boundary was prescribed as

$$u_1(t) = A_1 \sin(\omega_1 t),$$

(28)

where $A_1 \in [0, 0.2]$ and $\omega_1 \in [10^5, 10^6]$ for the UPM data, and $A_1 \in [0.005, 0.02]$ for the J2 data. From each trajectory 4000 steps with different $\Delta t$ to cover the frequency range were stored and a random sample of 8000 trajectories across modes were selected for the final dataset. These single element simulations took on average 10 cpu·ms per step (8.7 ms/step for J2 and 9.6 ms/step for UPM). Note no shuffling of loading directions was done to promote learning, as opposed to embedding, invariance.

5.4 Heterogeneous Response

To generate the data for samples with variable microstructure, a single set of realizations with spherical pores in a cubic sample was created and then three different material combinations were utilized to create three separate datasets. The three material combinations were (a) porous aluminum where the matrix followed the J2 model, (b) porous silicone where the matrix followed the UPM model, and (c) a glass-bead filled silicone where the matrix response was given by the UPM model and the inclusions were elastic. In this last case the Young’s modulus of the glass was 60 GPa and its Poisson’s ratio was 0.33. Case (c) is an example of a nonthermomechanically simple material where the similarity due to time-temperature scaling is not present.

Realizations were created by a random placement scheme of spherical voids in the sample cube with constraints on pore overlap with other pores and the sample boundary (Brown et al., 2018). This process created unit cells with mean porosity 0.09 and standard deviation 0.03 following a beta distribution. The 1120 nonperiodic realizations had porosities ranging from 0.015 to 0.017. The cube samples were on the order of $1.5^3$ mm$^3$ with pore radius $\approx 150$ µm. Pores in each of the realizations were explicitly meshed and resulted in unstructured discretizations with 14,640–101,360 eight node elements. The nominal element size was 60 µm. Meshing was performed using the Cubit/Sculpt meshing tool (Owen et al., 2014, 2017) following a process similar to Brown et al. (2018). These meshes were preprocessed into sparse adjacency matrices based on element neighbors for use with the GCNN component of the enhanced ISV-NODE, as in Frankel et al. (2021).

With the homogeneous models we could afford loading mode variety that was not feasible with the large meshes needed to represent the microstructure samples. In this case each realization was subjected to quasistatic uniaxial tension up to 20% engineering strain for case (a), the elastoplastic material, and to 50% for the viscoelastic materials, cases (b) and (c). Minimal Dirichlet boundary conditions were applied for static determinacy which resulted in inhomogeneous deformation due to the heterogeneous microstructure, as Fig. 4 shows. Figure 4 also illustrates the stress concentrations due to the second phase which are most extreme for the stiff inclusions, case (c). Each time step of the evolution took on the order of 10 cpu·s (10.4 s/step on average for J2, and 26.9 s/step for the UPM models). From these simulations we extracted microstructure $\Phi(X)$, applied strain $\varepsilon(t)$, and volume-averaged tensile stress $\bar{\sigma}(t)$ data to demonstrate the efficacy of mesh-based GCNNs in the Results section. Figure 5 shows the range of response for each of the three cases. Note a simple mixture model based on the sample porosity explains some of the variance in the response but 30% of the variance remains for case

Volume 3, Issue 3, 2022
FIG. 4: Stress fields for a representative sample of the pore and inclusion data: (a) elastic-plastic J2 matrix with pores, (b) viscoelastic UPM matrix with pores, and (c) viscoelastic UPM matrix with elastic inclusions. Mesh is colored by tensile stress; blue: $< 0$; red: $> 4$ MPa viscoelastic and $> 700$ MPa elastic-plastic. Deformed configuration is shown at 20% strain for the elastic-plastic simulation and at 50% strain for the viscoelastic simulations. The outline indicates the original configuration.

(a) after this rescaling and 45% for case (b). For case (c) this rescaling increases the variance. A mixture model that takes into account the two phases would be more explanatory but would require knowing the elastic moduli of the two phases.

6. RESULTS

To demonstrate the versatility of the proposed ISV-NODE network we applied it to the tasks of: (a) representing the response of homogeneous inelastic materials, and (b) the homogenization of inelastic samples with microstructure, using data described in the preceding section. The architectures of baseline models corresponding to the homogeneous ISV-NODE (Fig. 1) and heterogeneous ISV-NODE (Fig. 2) are summarized in Table 2.
FIG. 5: Stress-strain response for the pore and inclusion data models. Only the normal 11 components in the tensile direction are plotted and colors are used to distinguish independent samples. (a) Elastic-plastic J2 matrix with pores, (b) viscoelastic UPM matrix with pores, and (c) viscoelastic UPM matrix and elastic inclusions.

6.1 Homogeneous Material

Using multiple loading mode data for homogeneous samples described in Section 5.3, we explored fundamental questions regarding of the proposed framework: (a) is one of the stress models superior to the others, (b) can an optimal latent space be ascertained, (c) how well is the dissipation inequality be satisfied, and (d) can the representation be completely trained with a reasonable amount of data and extrapolate in time. Training these models took roughly 4–8 hr on a GPU or 20–24 hr on a multiprocessor CPU, where training cost increased superlinearly with history duration. Once trained, these models take approximately 15.8 µs/step to evaluate, which is approximately three orders of magnitude faster than the finite-element-based data source models.

6.1.1 Stress Model Formulations

First we compare the component, tensor-basis and potential based variants using a ISV-NODE with $N_F = 3$ layers in the state-evolution model, $N_S = 2$ layers in the stress model, and $N_h = 3$ assumed states. The component-based formulation Eq. (10) directly connects stress components to the outputs of certain NN nodes but lacks inherent equivariance/invariance. The tensor-basis formulation Eq. (9) places part of the burden of functional complexity on the known basis used to represent the stress. In the potential-based formulation Eq. (8) all information used to represent
TABLE 2: Baseline architectures for ISV-NODE models of the (a) homogeneous and (b) heterogeneous (RVE) material datasets. The ISV-NODE consists of two densely connected NN components: (1) \((6 + N_h) \times N_f\) flow rule and (2) \((6 + N_h) \times N_S\) stress rule. The GCNN+ISV-NODE applied to the homogenized RVE adds a third component that processes the initial microstructure. It consists of a stack of \(N_{\text{conv}}\) convolutional layers each with \(N_{\text{filters}}\); the output of this stack is fed into a stack of \(N_{\text{dense}}\) layers which, in turn, augments the hidden ISVs by providing additional initial conditions.

<table>
<thead>
<tr>
<th>Homogeneous</th>
<th>type</th>
<th>(N_h)</th>
<th>(N_f)</th>
<th>(N_S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPM viscoelastic</td>
<td>component</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>tensor-basis</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>potential</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>J2 elastoplastic</td>
<td>potential</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Heterogeneous</th>
<th>ISV-NODE</th>
<th>(N_{\text{filters}})</th>
<th>(N_{\text{conv}})</th>
<th>(N_{\text{dense}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Porous elastoplastic</td>
<td>ReLU</td>
<td>16</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>ELU</td>
<td>16</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Porous viscoelastic</td>
<td>ReLU</td>
<td>16</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>ELU</td>
<td>16</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Filled viscoelastic</td>
<td>ReLU</td>
<td>16</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>ELU</td>
<td>16</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

the stress is collapsed to a single, scalar output before a differentiation with respect to strain provides the stress tensor. It is also the only formulation of the three that has direct access to the power being dissipated.

Figure 6 shows the stress trajectories randomly selected UPM data and the relative differences of predictions for the three formulations. Note the trajectories are plotted with respect to time step as each trajectory has a distinct time step size and, hence, they have different durations. All formulations provide predictions of similar quality with oscillations in error that loosely correlate with the loading; however, the errors of the potential model resemble those of the component, whereas the tensor basis errors are more oscillatory and larger at time zero. This signature of errors is likely due to the inherent complexity of the tensor basis formulation, i.e., the representation is complete to order \(E^2\) regardless of whether the response needs higher-order basis elements. The potential formulation errors have some noticeable kinks with respect to smooth data, which may be due to the fact that a derivative is needed to form \(N_{N_S}\). In fact all formulations have kinks to some degree, which may be due to using relatively small MLPs with ELUs in all the ISV-NODE models. The fact that the maximum errors are generally out-of-sync with the loading seems to support the conjecture that the low-order approximations are the primary source of error. The empirical cumulative distribution function (CDF) of the differences between the predictions and the data, shown in Fig. 7, provides a more quantitative and global view which indicates that the models are comparably accurate. It also appears that the component-based approach has slightly better accuracy over this dataset, which is understandable since loading modes were not shuffled/augmented to teach invariance and, hence, the
A Neural Ordinary Differential Equation Framework

FIG. 6: True stress trajectories (a) and relative trajectory errors for (b) component-based, (c) tensor basis, and (d) potential-based stress variants of the ISV-NODE. Color distinguish different sample trajectories. Since each sample has a distinct time step, the data are shown on a step basis.

FIG. 7: Comparison of CDFs of the per step errors of the three stress formulations of the ISV-NODE component model could specialize to these data. It follows that the errors for the component model would be larger if trajectories with identical invariants but distinct components were included in the validation set (Frankel et al., 2020b). These results suggest that each formulation

Volume 3, Issue 3, 2022
can be comparably accurate; however, the potential-based has the advantage of direct access to dissipation, which will be explored in the following sections.

6.1.2 Latent State Space Size

We investigated whether there is an optimal number of internal state variables, $N_h$, to describe the state space. $N_h$, of a particular material response. The existence of an optimal number of variables is plausible for actual material response and seemingly certain for data generated from a traditional parameterized model. Using the UPM data, we varied the size of the state space $N_h$ and computed the validation error of the predictions. Figure 8 shows the accuracy of both the tensor basis and potential-based formulations for increasing size of the inferred state space $N_h$. It appears that the potential formulation has an optimum at $N_h = 3$, which is the number of isotropic invariants for an additional strain like quantity, while the tensor basis formulation is generally improving with increasing state space size for $N_h \leq 6$. The weak sensitivity to the size of the state space is likely due to the inherent quality of NN to handle redundancy by creating correlated outputs, as was observed in Frankel et al. (2019) where correlated features were extracted from images when more output nodes than were needed were available to the NN. This topic will be revisited with the J2 data in Section 6.1.5.

6.1.3 Hidden Variable Evolution and Dissipation

We devised two constraints to ensure proper dissipation. For the potential-based variant, we have direct access to the power expended by the hidden states and a connection to the second law. This connection is encapsulated by the inequality constraint, Eq. (16). Without this direct connection, we resorted to matching the power expended by the model to that of the data using the equality constraint, Eq. (17).

Figure 9 shows the error in the stress, Eq. (14), and the error in the total power, Eq. (17), for the two formulations with $N_h = 3$ and the UPM data. Clearly there is an optimal value for the penalty for both formulations based on the stress error, but their accuracies are both relatively insensitive to the particular value of the penalty $\varepsilon$. This indicates that the expended power constraint $\| (\mathbf{S} - \hat{\mathbf{S}}) \cdot \dot{\mathbf{E}} \|$, which is essentially the stress error in the direction of the strain rate $\dot{\mathbf{E}}$, is generally compatible with the formulations. In both cases the power error keeps decreasing with increased penalty $\varepsilon$, which is expected. Note that for the tensor basis formulation, this error is

![Figure 8: Sensitivity of the RMSE error and correlation to state space size $N_h$](image)
A Neural Ordinary Differential Equation Framework

FIG. 9: Sensitivity of the stress and power RMSE errors to the penalty parameter $\epsilon$

exactly what is being penalized, whereas for the potential formulation any positive internal state power (signaling second law violations and instability) is penalized.

For potential formulation we examined the effect of the penalization on the internal evolution. Figure 10 shows the state evolution and the internal state power with and without penalization. Recall that the time step for each trajectory is distinct. Clearly the penalization is necessary to prevent positive internal state power and second-law violations. Other than guiding the optimal weights during training, the penalization of nondissipation has no other effect on predictions shown. In fact penalization results in the power expended to increase over a few cycles of displacement loading and then level off as the material reaches a steady state with the loading, which is expected from a viscoelastic material with harmonic loading. For both cases the states follow generally linear growth trends with sinusoidal fluctuations. There is correlation with the state fluctuations, the power expended, and the loading, as expected. Penalization also has the effect of reducing the oscillations relative to the linear trends in the state evolution for the harmonically forced dataset. For this material there is always dissipation with loading so we do not expect to observe phases where $\dot{\mathbf{h}} = 0$. This will be revisited in Section 6.1.5.

6.1.4 Time Extrapolation

As a last investigation with the homogeneous UPM material dataset, we tested whether training eventually becomes complete, in the sense that the extrapolation errors decrease, with increasing duration of the training data. Figure 11 illustrates typical convergence of the sequential training strategy described in Section 4 whereby the ISV-NODE is trained on $N_{\text{steps}} = \{40, 60, 80, 120, 200\}$ in order to allow the NNs to adjust to initial trends before accommodating long-term behavior. Clearly rapid loss reduction is followed by slow improvement for each stage, and there is steady improvement across the sequence of stages. For the last stage we typically train to 1000 epochs (not shown). Figure 12 shows convergence of time extrapolation errors with increasing training steps $N_{\text{steps}}$. Clearly the extrapolation errors become effectively constant in time indicating that the essentials of the material dissipation and internal state evolution have been incorporated into the ISV-NODE model. Ultimately the accuracy of
FIG. 10: State evolution and power without ($\varepsilon = 0$) and with ($\varepsilon > 0$) dissipation constraint for potential formulation. State trajectories from the same sample are plotted with the same color. (a) state $\varepsilon = 0$, (b) state $\varepsilon = 10^2$, (c) power $\varepsilon = 0$, and (d) power $\varepsilon = 10^3$.

the ISV-NODE is limited by the complexity of the flow and stress MLPs. Results are shown for the potential-based ISV-NODE variant; those for the tensor basis formulation are similar.

FIG. 11: Convergence of the sequential training strategy
6.1.5 Conservation and Dissipation

The ISV-NODE model, in particular the potential-based formulation, has the ability to represent both dissipative processes, where the hidden state evolves, and conservative processes, where the hidden state should be constant and the stress potential is only a function of the varying strain. The J2 elastic-plastic data have both aspects, hysteresis loops which begin elastic and then transition to plastic flow when yield is encountered.

Using a ISV-NODE with \( N_F = 3 \) layers in the flow model and \( N_S = 3 \) layers in the stress model, we first determined an optimal number of state variables \( N_h \). The ISV-NODE model of the J2 had similar trends but stronger dependence on the number of state variables than the ISV-NODE model of the UPM data. It appears that the more complex (higher \( N_h \)) models train slower. For a feasible number of sequential training stages and epochs, models with \( N_h = 1, 2 \) are comparably accurate over the initial part of the trajectories. Figure 13 shows the CDF of prediction errors, which are distinctly higher than for UPM data. This is expected due to the difficulty of representing the nonsmooth J2 data with two smooth functions \( f \) and \( S \).

FIG. 13: CDF of the differences between the true and predicted trajectories of potential-based ISV-NODE trained to J2 data
Figure 14 shows the true and predicted stress-strain hysteresis for an ISV-NODE with $N_h = 2$ and trained with penalty $\varepsilon = 10.0$. The ISV-NODE predictions are close to the data even through multiple reversals. The corresponding state evolution and state power are shown in Fig. 15. Consistent with the expectations from ISV theory, there are stages where the states are essentially constant, corresponding to elastic formation, and others where there is dissipation. Given the smoothness of the underlying MLPs in the ISV-NODE, the switching between dissipation and conservation is a smooth approximation of the abrupt transition in the data. Also noticeable in Fig. 14 are spurious oscillations when trajectories cross an elastic-plastic transition (as defined by the underlying mode); we believe this is due to insufficient sampling of this regime, although it also could be due to the limited number of nodes in the MLPs since similar kinks are apparent in Fig. 6. Nevertheless the accuracy of the trajectories appears to recover after these events before degrading after multiple reversals. Clearly there are limitations with this representation of the J2 data; that being said experimental data tend to be generally smoother than the response of the J2 model. A deeper, more complex flow NN $\mathbf{NN}_h$ to capture the flow–no flow transitions more precisely and/or adaptive time stepping to minimize under- and overshoots at transitions should improve the performance. Without these improvements the deployment of this particular ISV-NODE in a structural simulation could be problematic. This topic will be discussed further in Section 7.

6.2 Homogenized Microstructure

To model the volume-averaged stress response of heterogeneous samples we used an ISV-NODE with $N_f = 3$ layers in the state-evolution model, and $N_S = 2$ layers in the stress model including the linear output layer and $N_h = 8$ assumed states. A larger number of internal states was selected based on the assumption that the internal state of a heterogeneous sample will be more

FIG. 14: Comparison of true (black) and predicted (red) stress-strain hysteresis for the J2 data
complex; however, no tuning of $N_h$ was done due to the expense of training the GCNN+ISV-NODE. The size assumed state $N_h$ was augmented by the output $\phi$ of a graph convolutional NN reducing the microstructure $\phi$ to relevant features. The graph convolutional NN subcomponent, refer to Fig. 2(b), processed the binary porosity field $\phi$ on the unstructured meshes with $N_{conv} = 4$ convolutional layers each with $N_{filter} = 16$. This output was reduced by global average pooling, and finally processed by $N_{dense} = 3$ densely connected layers including the linear output layer resulting in the structural features $\phi$ correlated with the output $S(t)$. The 16 structural features were passed into the ISV-NODE as initial conditions for the hidden state $h(0)$. In this study we only had tensile data, so we reduced the input strains, strain rates, and stress to one component. Unlike for the main ISV-NODE, a ReLU activation was used in the GCNN. Also, unlike our previous CNN-RNN architectures (Frankel et al., 2020a, 2021, 2019), the output of reducing the initial microstructure with a GCNN is a true initial condition that augments the state space, as opposed to a constant loading-like input. This feature makes this architecture consistent with homogenization as an initial value problem where microstructure, as encoded in the latent vector $h$, can evolve.

As discussed in Section 5.4 we generated three datasets: (a) porous aluminum with a elasto-plastic J2 material model, (b) porous silicone with a viscoelastic UPM material model, and (c) silicone with elastic glass inclusions. Figure 16 shows the evolution of RMSE of predictions of held-out data over time (note all training/testing data had the same duration and time step). For the porous plastic samples the mean prediction error was 0.0033, and the mean correlation was 0.993 over held-out samples and across time. The predictions for the porous viscoelastic samples were similarly accurate: the mean prediction error was 0.0037, and the mean correlation was 0.998. The predictions for the more challenging composite viscoelastic matrix with stiff elastic inclusion samples were slightly less accurate: the mean prediction error was 0.011, and the mean correlation was 0.978. Figure 17 compares the true and predicted response trajectories for the held-out samples with the minimum, median, and maximum RMSE. In fact, even for the worst cases, the predictions (dashed) and true (solid) trajectories are nearly indistinguishable. Clearly the combined GCNN+ISV-NODE architecture produces accurate predictions that distinguish the effects of varying microstructure and capture their evolution. This also demonstrates that the GCNN+ISV-NODE is comparably accurate on different classes of inelastic response. The
FIG. 16: Evolution of mean error for ISV-NODE models of a J2 matrix with pores, a UPM matrix with pores, and a UPM matrix with elastic inclusions.

FIG. 17: Predictions for response with microstructure. The predictions (dashed lines) with the minimum (blue), median (black), and maximum (red) RMSE error are show with the true trajectories (solid lines with corresponding colors). (a) Elastic-plastic matrix with pores, (b) viscoelastic matrix with pores, and (c) viscoelastic matrix with elastic inclusions.

GCNN+ISV-NODE took approximately 24 hr to train on a GPU in a single stage. Once trained, the GCNN+ISV-NODE takes 1.47 ms/step to evaluate on a given adjacency matrix and loading history on average; this represents about three orders of magnitude speed-up over the traditional model with finite-element mesh and the same loading.
7. CONCLUSION

By basing a neural ODE-like network on classical ISV theory, the proposed ISV-NODE was able to model a variety of inelastic processes and learn internal states in a general and extensible manner. Two primary variants, based on how the stress MLP is formulated, were tested. The potential-based variant allows direct connection with the dissipation requirements of Coleman-Gurtin theory; however, the tensor basis version has inherent polynomial complexity that reduces the modeling burden on the trainable component functions. Both are exactly equivariant by design. This framework precludes the discrepancy/model-form errors that are inherent in calibrating preconceived, traditional models. In addition we demonstrated the application of the ISV-NODE to modeling materials with microstructure where a GCNN processed the initial microstructure to inform the initial state. We expect that the ISV-NODE will be as effective in modeling materials not easily categorized into elastoplastic or viscoelastic as it was in modeling exemplars of those responses. Although the state variables themselves do not currently have a precise physical interpretation, the ISV-NODE has components that are analogs of traditional stress and flow rules. This fact makes ISV-NODE models drop-in replacements for traditional models in existing simulators. We have demonstrated that accurate NN models can be on the order of $1000\times$ faster to evaluate than their traditional counterparts. Furthermore, utilizing a single, general, data-driven constitutive modeling framework will simplify validation of suites of material models needed to simulate full systems. Also uncertainty quantification (UQ) can be embedded in an ISV-NODE with variational inference-based extensions to the two constituent MLPs (Graves, 2011).

By learning directly from data using an ISV-NODE model with a tunable set of hidden variables representing material state, the process of building low-discrepancy models should become relatively straightforward. We expect that this generalized framework for inelastic response will lead to an agile and robust process of constructing constitutive models. As demonstrated, the state space can be built with complexity appropriate for the available data and physical process. The ISV-NODE and its microstructural extension have multiple applications beyond accelerating large-scale simulations such as providing efficient subgrid models of complex microstructures and surrogates for enabling high-dimensional sampling-based UQ, material optimization, and structure-property maps.

Our findings suggest multiple avenues for future work. To fully realize the promise of the ISV-NODE framework we intend to extend it to represent nonisothermal processes and embed all the thermodynamic implications of the Coleman-Gurtin ISV theory (Coleman and Gurtin, 1967). Since complete sampling of loading modes remains a challenge (Fuhr and Bouklas, 2021; Logarzo et al., 2021), as does incorporating limited experimental data, we are pursuing both active learning (Settles, 2009, 2012) and data fusion techniques (Bleiholder and Naumann, 2009; Castanedo, 2013). A related challenge is generating enough microstructure response data to train a model capable of predicting general loading modes. For this issue we are pursuing transfer learning where an ISV-NODE with microstructural inputs is first trained to the homogeneous multimodal data and then to the more limited heterogeneous data. Lastly, one of the more fundamental challenges results from applying the ISV-NODE framework to elastic-plastic response. Effectively we are tasking the flow MLP $\mathbf{NN}_f$ with learning a vector field, which, in the case of elastoplasticity, is not a smooth one (i.e., it is zero in the elastic region and nonzero outside). In the ISV-NODE formulation proposed in this work the primary input is strain. With traditional elastoplastic models, like J2, the elastic limits are simpler to describe in stress space, which suggests feeding the predicted stress back into flow MLP. Other techniques such as basing the flow...
MLP on a dissipation potential, using a derivative of a single output node as in the stress MLP, may yield improved predictions similar to level set technique used in Vlassis and Sun (2020).

ACKNOWLEDGMENTS

We are grateful for helpful discussions with Craig Hamel (Sandia), Kevin Long (Sandia), and Coleman Alleman (Sandia). Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA-0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

REFERENCES


Rubliner, J., On Fading Memory in Materials of Evolutionary Type, *Acta Mech.*, vol. 8, no. 1, pp. 75–81,
1969.


Settles, B., *Active Learning: Synthesis Lectures on Artificial Intelligence and Machine Learning*, Williston,
Based on fundamental developments in tensor function representation theory (Boehler, 1987; Truesdell and Noll, 2004), the stress \( S \) as a function of its tensor arguments has the general representation

\[
S = \sum_i \sigma_i B_i ,
\]  

(A.1)
where the elements of the basis $B_i \in B$ are known powers of the tensor inputs and the coefficients $\sigma_i = \sigma_i(I)$ are functions of the (scalar) invariants $I$ of the inputs. If stress is only a function of strain, $S(E)$, its representation consists of three terms

$$S = \sum_{i=0}^{2} \sigma_i(I)E^i,$$  \hspace{0.5cm} (A.2)

where the basis $B_i$ are powers of strain $E^i$ and $I = \{\text{tr } E, 1/2(\text{tr}^2 E - \text{tr } E^2), \text{det } E\}$ is a complete set of invariants. Beyond the fact that both the second Piola-Kirchhof stress $S$ and the Lagrange strain $E$ are material frame invariant, this representation represents a compact, coordinate-free description. Note that for $S(E)$, $S$, and $E$ are collinear, in that they have the same eigenvectors.

For tensor function of two tensor arguments, such as $S(E, \dot{E})$, Rivlin (1955) showed that ten polynomial invariants,

$$I = \{\text{tr } E, \text{tr } E^2, \text{tr } E^3, \text{tr } \dot{E}, \text{tr } \dot{E}^2, \text{tr } EE, \text{tr } EE^2, \text{tr } E^2 \dot{E}, \text{tr } E^2 E^2\},$$  \hspace{0.5cm} (A.3)

and a nine element tensor basis,

$$B = \{I, E, E^2, \dot{E}, \dot{E}^2, \text{sym } EE, \text{sym } E^2 \dot{E}, \text{sym } E \dot{E}^2, \text{sym } E^2 E^2\},$$  \hspace{0.5cm} (A.4)

are, in general, necessary. See also Boehler (1987) [Ch. 3, Eqs. (9) and (11)], and note here $\dot{E}^2 \equiv \dot{E} \dot{E}$. However, nine basis elements are more than necessary to represent a symmetric tensor with six independent components; hence, the coefficient functions cannot be solved for directly [as in Frankel et al. (2020b)].

For the special case where one argument is the rate of the other it can be shown that only six tensor basis elements,

$$B^* = \{E^0, E^1, E^2, \dot{E}, \dot{E}^2, \dot{E}^3\},$$  \hspace{0.5cm} (A.5)

give a complete and irreducible basis. In addition to reducing the basis this allows for the coefficients of the tensor basis representation to be solved for explicitly. Note Peters et al. (2020) also employed a reduced basis in a TBNN for fluids.

First recognize that the first three elements of $B^*$ are linearly independent and span the eigenspace of $E$,

$$\text{span}\{E^p, p = 0, 1, 2\} = \text{span}\{a_i \otimes a_i, i = 1, 2, 3\} = \text{span } A,$$

where

$$E = \sum \lambda_i a_i \otimes a_i,$$  \hspace{0.5cm} (A.7)

is the spectral decomposition of $E$, with $\lambda_i$ being its eigenvalues and $a_i$ its eigenvectors. This follows from Wang’s Lemma (Bowen and Wang, 2008) as outlined by Gurtin (1982) (Section 37), which implies

$$\sum_{j=0}^{2} c_j E^j = 0 \iff \sum_{j=0}^{2} c_j \lambda_i^j = 0 \iff c_j = 0,$$  \hspace{0.5cm} (A.8)

for $E$ having three distinct eigenvalues $\lambda_i$; the other cases involving multiplicity of the eigenvalues follow by the same logic.
If we now examine the rate, 
\[ \dot{E} = \sum_i \dot{\lambda}_i a_i \otimes a_i + \sum_i \lambda_i (\dot{a}_i \otimes a_i + a_i \otimes \dot{a}_i), \] (A.9)
given the fact that \( \dot{a}_i \cdot a_i = 0 \) since \( a_i \) are unit vectors, \( \|a_i\| = 1 \), it follows that
\[ \text{span}(\dot{a}_i \otimes a_i + a_i \otimes \dot{a}_i) = \text{span} \{ a_j \otimes a_i + a_i \otimes a_j, i \neq j \}, \] (A.10)
and these three additional elements complete the basis (Boehler, 1987, Ch. 3). Unfortunately the eigenvector dyads, \( a_i \otimes a_j \), although being an orthonormal basis, are not a convenient basis since they are expensive to compute and are not permutational invariant like the usual scalar invariants (e.g., ordering the eigenvalues does not preserve continuity in time).

Nevertheless if we examine the five remaining basis elements of the general tensor basis:
(a) 
\[ \dot{E} = \sum_i \dot{\lambda}_i a_i \otimes a_i + \sum_i \lambda_i (\dot{a}_i \otimes a_i + a_i \otimes \dot{a}_i), \] (A.11)
(b) 
\[ (\dot{E})^2 = \left( \sum_i \dot{\lambda}_i a_i \otimes a_i + \sum_i \lambda_i (\dot{a}_i \otimes a_i + a_i \otimes \dot{a}_i) \right)^2 \]
\[ = \sum_i \dot{\lambda}_i^2 a_i \otimes a_i + \sum_i \lambda_i \dot{\lambda}_i (\dot{a}_i \otimes a_i + a_i \otimes \dot{a}_i) + \lambda_i^2 (\dot{a}_i \otimes \dot{a}_i + (\dot{a}_i \cdot \dot{a}_i) a_i \otimes a_i), \] (A.12)
(c) 
\[ \text{sym} \dot{E} \dot{E} = \text{sym} \left[ \sum_i \dot{\lambda}_i a_i \otimes a_i \right] \times \left( \sum_i \dot{\lambda}_i a_i \otimes a_i + \sum_i \lambda_i (\dot{a}_i \otimes a_i + a_i \otimes \dot{a}_i) \right) \]
\[ = \sum_i \dot{\lambda}_i \lambda_i a_i \otimes a_i + \sum_i \lambda_i^2 (\dot{a}_i \otimes a_i + a_i \otimes \dot{a}_i), \] (A.13)
(d) 
\[ \text{sym} E^2 \dot{E} = \sum_i \lambda_i \dot{\lambda}_i a_i \otimes a_i + \sum_i \lambda_i^2 (\dot{a}_i \otimes a_i + a_i \otimes \dot{a}_i), \] (A.14)
we observe that the parts of \( \{ \dot{E}, \text{sym} \, \dot{E}, \text{sym} \, \ddot{E} \} \) not in the span of \( A \) depend on \( \lambda_i, \lambda_i^2, \) and \( \lambda_i^3 \), respectively, Hence, Wang’s Lemma can be used again (with the trivial restriction \( \lambda_i > 0 \)) to state

\[
\text{span}\{ \dot{E}, \text{sym} \, \dot{E}, \text{sym} \, \ddot{E} \} = \text{span}\{ (\dot{a}_i \otimes a_i + a_i \otimes \dot{a}_i), \, 0 \leq i < 3 \}.
\]  

(A.16)

Furthermore, since \( \text{span} \{ E^a \} = \text{span} \{ \dot{E}, \text{sym} \, \dot{E}, \text{sym} \, \ddot{E} \} \), we can claim

\[
\text{span}\{ E^a \}, \, 0 < a \leq 3 = \text{span}\{ \dot{E}, \text{sym} \, \dot{E}, \text{sym} \, \ddot{E} \},
\]  

(A.17)

and finally \( \mathcal{B}^* = \{ E^a \mid 0 \leq a < 3 \} \cup \{ \text{span} \{ E^a \} \mid 0 < a \leq 3 \} \) has the same span as \( \mathcal{B} \) and is linearly independent.

Beyond reducing the complexity of the representation, this reduced basis enables solving for the coefficient functions directly as in Frankel et al. (2020b). Although we do not employ the explicit method of solving for the coefficient functions directly in this work it has advantages in training TBNNs. It does, however, have a complication when there is multiplicity in the eigenvalues of the input tensors. Although in finite precision arithmetic this is rarely encountered, it does degrade the conditioning of the linear system that must be solved. Similarly simple loadings were \( \dot{E} \) and \( E \) are collinear and/or \( \dot{a}_i = 0 \) can lead to rank deficiencies. Gurtin (1982) provided a well-conditioned solution by way of solving a reduced system. For instance, in the case of \( S = S(E) \) and two of the eigenvalues of \( E \) are identical (as in uniaxial tension), \( E \) can be represented as \( E = \lambda_1 a \otimes a + \lambda_2 (I - a \otimes a) \) where \( \lambda_1 \) is the unique eigenvalue and \( \lambda_2 \) is the repeated one. Now instead of solving

\[
\begin{bmatrix}
\varsigma_1 \\
\varsigma_2 \\
\varsigma_3
\end{bmatrix} =
\begin{bmatrix}
1 & \lambda_1 & \lambda_1^2 \\
1 & \lambda_2 & \lambda_2^2 \\
1 & \lambda_3 & \lambda_3^2
\end{bmatrix}
\begin{bmatrix}
\sigma_0 \\
\sigma_1 \\
\sigma_2
\end{bmatrix},
\]  

(A.18)

where \( \varsigma_i \) are the eigenvalues of \( S \), only

\[
\begin{bmatrix}
\varsigma_1 \\
\varsigma_2
\end{bmatrix} =
\begin{bmatrix}
1 & \lambda_1 \\
1 & \lambda_2
\end{bmatrix}
\begin{bmatrix}
\sigma_0 \\
\sigma_1
\end{bmatrix},
\]  

(A.19)

needs to be solved since the coefficient \( \sigma_2 = 0 \). We proposed an alternative scheme in Frankel et al. (2020b).