ARMY RESEARCH LABORATORY

Implementation of a Mechanochemical Model for Dynamic Brittle Fracture in SIERRA

by Adam Sokolow

ARL-CR-0742 **August 2014**

Prepared by

Oak Ridge Institute for Science and Education
ORAU Maryland
4692 Millennium Drive, Suite 101
Belcamp, MD 21017

Under contract: 1120-1120-99

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5066

ARL-CR-0742 **August 2014**

Implementation of a Mechanochemical Model for Dynamic Brittle Fracture in SIERRA

Adam Sokolow

Oak Ridge Institute for Science and Education

Prepared by

Oak Ridge Institute for Science and Education
ORAU Maryland
4692 Millennium Drive, Suite 101
Belcamp, MD 21017

Under contract: 1120-1120-99

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
August 2014	Final	September 2013 – April 2014

4. TITLE AND SUBTITLE
Implementation of a Mechanochemical Model for Dynamic Brittle Fracture in SIERRA

5a. CONTRACT NUMBER
1120-1120-99

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Adam Sokolow

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: RDRL-WMP-B
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-CR-0742

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report discusses the implementation of a mechanochemical model for brittle fracture in the government code SIERRA. It is largely based on damage models for which damage is an internal state variable. The model is introduced first in a linear elastic context and then a similar model is developed using finite deformation theory. The model is then verified and some example applications discussed.

15. SUBJECT TERMS
brittle fracture, mechanochemical, dynamic failure, damage evolution, SIERRA

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Adam Sokolow
Unclassified	Unclassified	Unclassified	UU	52	19b. TELEPHONE NUMBER (*Include area code*) 410-306-2985

Contents

List of Figures

List of Tables

Acknowledgments

We would like to thank Mike Scheidler and Rich Becker for offering their expertise during various aspects of this project and of course Michael Grinfeld for introducing me to this problem.

This research was supported in part by an appointment to the Postgraduate Research Participation Program at the U.S. Army Research Laboratory (ARL) administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and ARL.

INTENTIONALLY LEFT BLANK.

1. Introduction

The fracture and failure of brittle materials has remained a core focus in Army related sciences. Of particular interest is the destabilization of an axisymmetric solution in problems that are completely axisymmetric. This destabilization problem largely motivated Grinfeld and Wright[1] to study a model for dynamic failure that depends on the interplay between two physical effects: internal elastic energy, and the energy associated with breaking chemical bonds. Because of these two considerations, the model is referred to as a *mechanochemical* model for fracture. The motivation of the mechanochemical model is based on Stress Driven Rearrangements Instabilities of phase interfaces Grinfeld[2] and is summarized in Grinfeld[3] and Kassner et al..[4] Motivated by the Stress Driven Rearrangements Instabilities, the model was then used in a different context to form the mechanochemical model, Grinfeld and Wright,[1] which was numerically implemented in MATLAB to handle quasi-static loading cases. Grinfeld et al.[5] used this implementation in MATLAB to established the appearance of radially damaged zones and the destabilization of the axisymmetric solution.

A clear and condensed presentation of damage theory is given by Kachanov,[6] as well by Chaboche,[7, 8] and a short introduction to damage as an internal state variable can also be found in the book by Holzapfel.[9] In the types of damage models considered by Kachanov, the energy density equation is modified by a reduction factor or a *damage function* that depends on an internal state variable, the *damage*. As one might expect, the damage reduces the internal energy thereby reducing the stress response of the damaged material. In the mechanochemical model discussed here, the damage also contributes to the internal energy. This addition to the energy potential creates an interplay between mechanical and chemical constituents and produces the source of the instability.

The original intent was to implement the mechanochemical model discussed in Grinfeld and Wright[1] into the government code SIERRA. The model discussed in Grinfeld and Wright[1] is based on a modified linear elasticity. In this report, however, we focus on a finite deformation, nonlinear formulation of the problem. This change was required by constraints introduced within the finite element solver SIERRA. In SIERRA, constitutive model designs are split between rate integrated and hyperelastic formulations. However, the features of the mechanochemical model make the decision between the two nontrivial. The mechanochemical model largely depends on the internal elastic energy, which makes it a good candidate for the hyperelastic approach. The desired linear elastic response on the other hand, would be simplest to implement in the rate

integrated formulation. At first, a rate integrated version of the mechanochemical model was implemented in SIERRA. This initial implementation determined the linear elastic energy through numerical integration but was prone to error. As a compromise, an alternate formulation of the model was developed that mimicked the linear elastic response using the log-strain tensor. This method benefited from an accurate calculation of elastic energy and has proved to be more stable. The model and its implementation in SIERRA are discussed in this report. Unlike the previous implementation in MATLAB,[5] which could only handle quasi-static loading, this implementation allows for the dynamic problem of failure to be studied on a massively parallel computational architecture.

The report is organized as follows. We discuss the general physical features and introduce the system of equations that describe the damageable material in section 2. This is done in the framework of the linear elastic theory and largely recapitulates the work of Grinfeld and Wright.[10] In section 3 we introduce the log-strain tensor and derive a new formulation of the mechanochemical model for finite deformations. Section 4 presents the numerical details of how the damage evolution is handled and discusses some of its subtleties. In section 5 we verify the implementation of our model in SIERRA using some single element tests. A few example problems that illustrate applications of the mechanochemical model are discussed in section 6, and possible future alterations to the model are discussed in section 7. We make concluding remarks in section 8 and derive the equations of damage evolution in appendix A. Appendix B includes the code used to implement the model in SIERRA.

2. Mechanochemical Model: Linear Elastic Framework

In this section we largely recapitulate the work by Grinfeld and Wright[10] but with additional comments to motivate the following section where we introduce a finite deformation mechanochemical model. We begin by introducing the terms considered in our energy density potential from which we derive the elastic stress and the equations that govern the chemical kinetics.

We take the total energy density to be a function of the mechanical deformation (or strain) and the damage. Specifically, we assume the energy can be decomposed additively into a mechanical part and a chemical part, i.e.,

$$e = e_{\text{mechanical}} + e_{\text{chemical}} \,. \tag{1}$$

Holding the damage fixed, the derivative of the energy with respect to the strain gives the stress,

$$\left.\frac{\partial e}{\partial\, \text{strain}}\right|_{\text{damage constant}} = \text{stress}\,. \tag{2}$$

Similarly, holding the strain constant and taking the derivative of energy with respect to the damage gives the driving force for damage evolution,

$$\left.\frac{\partial e}{\partial \kappa}\right|_{\text{strain constant}} \propto \frac{\partial}{\partial t}\text{damage}\,. \tag{3}$$

Now we define the mechanical and chemical terms for a linear elastic material. We assume the mechanical energy is reduced by a damage function ϕ, which depends on the damage κ so that the mechanical energy is given by

$$e_{\text{mechanical}} = \phi(\kappa)\, e_{\text{elastic}} \,. \tag{4}$$

Here $e_{\text{mechanical}}$ is the mechanical energy of a *damageable material*, and e_{elastic} is the elastic energy associated with a hypothetical *undamageable material* undergoing the same deformation. We also assume the damage function $\phi(\kappa)$ reduces the elastic energy with increasing damage. Specifically, we take our damage function $\phi(\kappa)$ to be

$$\phi(\kappa; \kappa^*, c_{min}) = \begin{cases} 1 - (1 - c_{min})\frac{\kappa}{\kappa^*} & : \kappa \leq \kappa^* \\ c_{min} & : \kappa > \kappa^* \end{cases} \,. \tag{5}$$

The functional form of equation 5 is a simple linear ramp between undamaged and a maximally

damaged material. The parameters κ^* is the maximal damage value. If while updating the value of κ the damage exceeds κ^*, it is simply set to κ^* (see section 4). The parameter c_{min} is the lower limit to the reduction factor of a fully damaged material.

Using a linear elastic theory for the elastic energy term, we obtain the following energy density per unit reference volume:

$$e_{\text{mechanical}} = \mu\phi(\kappa)\left(\frac{\nu}{1-2\nu}u^{i}_{.|i}u^{j}_{.|j} + u_{(i|j)}u^{ij}_{.|.}\right) . \tag{6}$$

Here u_{ij} is the linear elastic strain tensor,

$$u_{ij} \equiv \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) , \tag{7}$$

where u_i is the displacement vector of the deformation.

The second term in the energy density is the chemical energy. This energy is assumed to be quadratic and of the form

$$e_{\text{chemical}} = \frac{\xi}{2}(\kappa - \kappa^o)^2. \tag{8}$$

Here ξ is a chemical constant with dimensions of stress and κ^o the damage associated with minimum energy.

Combining these two terms into an energy density, we have the so-called Kachanov-Lifshitz function.

$$e(u_{i|j}, \kappa) = \mu\phi(\kappa)\left(\frac{\nu}{1-2\nu}u^{i}_{.|i}u^{j}_{.|j} + u_{(i|j)}u^{ij}_{.|.}\right) + \frac{\xi}{2}\left(\kappa - \kappa^o\right)^2 . \tag{9}$$

Thus, by considering changes in the energy density, and either holding the strain or the damage constant, we obtain:

$$\left.\frac{\partial e}{\partial u_{ik}}\right|_{\kappa \text{ constant}} = 2\mu\phi(\kappa)\left[\frac{\nu}{1-2\nu}u_{ll}\delta_{ik} + u_{ik}\right] \equiv \sigma_{ik} \tag{10}$$

and

$$\left.\frac{\partial e}{\partial \kappa}\right|_{u_{ik} \text{ constant}} = \frac{\partial \phi}{\partial \kappa}e_{\text{elastic}} + \xi\left(\kappa - \kappa^o\right) \tag{11}$$

$$= \frac{\partial \phi}{\partial \kappa}\mu\left[\frac{\nu}{1-2\nu}u^{m}_{.\,|m}u^{n}_{.|n} + u_{(m|n)}u^{mn}_{.|.}\right] + \xi\left(\kappa - \kappa^o\right) . \tag{12}$$

The stress tensor retains its usual form, i.e., it is still defined in terms of the derivative of the

energy with respect to the strain tensor; however, it is modified by the damage function $\phi(\kappa)$. Similarly, a *generalized-stress* is derived from the chemical energy. It has units of stress and drives changes in the damage. This generalized-stress is referred to in the literature as a *chemical potential*. The following two equations result from combining the stress equation 10, with momentum balance, and the chemical potential 11 with a kinetic reaction law:

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \frac{\partial \sigma_{ik}}{\partial x_k} = \frac{\partial}{\partial x_k} \left(2\phi(\kappa)\mu \left[\frac{\nu}{1-2\nu} u_{ll}\delta_{ik} + u_{ik} \right] \right) , \quad \text{and} \tag{13a}$$

$$\frac{\partial \kappa}{\partial t} = -K \left(\frac{\partial \phi}{\partial \kappa} \mu \left[\frac{\nu}{1-2\nu} u^m_{\ .|m} u^n_{\ .|n} + u_{(m|n)} u^{mn}_{\ .|.} \right] + \xi \left(\kappa - \kappa^o \right) \right) . \tag{13b}$$

The dimensions of K are:

$$[K] = \frac{1}{[\text{Time}][\text{Stress}]} . \tag{14}$$

The specific choice of equations 13 contain two physically questionable features, which we discuss in later sections. Namely, a degrading bulk modulus and reversible damage. Reversible damage used in this report is not consistent with the work by Grinfeld and Wright[10] where the derivative of the damage is forced to remain positive. See section 7.1 for details regarding the minor alteration in the constitutive model that disables healing.

---

3. Mechanochemical Model: Finite Deformation Formulation

As discussed in the introduciton, the original goal of the research was to implement the mechanochemical model for brittle fracture described in the previous section by equations 5 and 13 as a material model in SIERRA. However, the choice to implement the constitutive model in SIERRA carried with it a number of constraints. In this section we derive a constitutive relation for an isotropic hyperelastic material (hyperelastic for fixed damage) that captures the same physical features described in the previous section. This does not represent the most general nonlinear model, and in fact is a special case that was chosen to mimic the linear elastic mechanochemical model when used in the small strain limit.

3.1 Finite Deformations and the Log-Strain Tensor

The following briefly covers some modern continuum mechanics and introduces the log-strain tensor. Two good references on modern continuum mechanics are the text books by Holzapfel,[9] and Truesdell and Noll.[11] The Cauchy stress tensor $\boldsymbol{T}$ is a linear transformation of a direction vector to a traction force in the current configuration. It can be written in terms of the spherical

part, i.e., the pressure p, and its deviatoric part $\boldsymbol{T}^*$:

$$\boldsymbol{T} = -p\boldsymbol{I} + \boldsymbol{T}^* \,, \qquad p \equiv -\frac{1}{3}\,\mathrm{tr}\,\boldsymbol{T} \,. \tag{15}$$

We use the convention that normal stress components are positive in tension. The deviatoric part of a tensor, denoted by a superscript *, is defined as follows:

$$\boldsymbol{A}^* \equiv \boldsymbol{A} - \frac{1}{3}\mathrm{tr}\,[\boldsymbol{A}]\,\boldsymbol{I} \,, \tag{16}$$

where $\boldsymbol{I}$ is the identity tensor. The deformation gradient $\boldsymbol{F}$ is a linear transformation from the reference configuration to the current configuration, which can be expressed in terms of a properly orthogonal rotation matrix $\boldsymbol{R}$ and positive definite symmetric matrices $\boldsymbol{U}$ or $\boldsymbol{V}$, called the right- and left-stretch tensors, respectively, using the polar decomposition

$$\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U} = \boldsymbol{V}\boldsymbol{R} \,. \tag{17}$$

Both $\boldsymbol{U}$ and $\boldsymbol{V}$ share the same eigenvalues λ_i called the principal stretches. The Jacobian J is the determinant of the deformation gradient,

$$J \equiv \mathrm{det}\boldsymbol{F} = \mathrm{det}\boldsymbol{V} \,, \tag{18}$$

and is equal to the ratio of the current specific volume to the reference specific volume. The right and left Cauchy-Green deformation tensors, respectively, are defined as

$$\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{U}^2 \quad \text{and} \tag{19}$$

$$\boldsymbol{B} = \boldsymbol{F}\boldsymbol{F}^T = \boldsymbol{V}^2 \,. \tag{20}$$

Since $\boldsymbol{C}$ and $\boldsymbol{B}$ are symmetric and positive definite, the eigenvalues of $\boldsymbol{C}$ and $\boldsymbol{B}$ are λ_i^2. The principal invariants of a second-order tensor $\boldsymbol{A}$ are given by

$$I_A = \mathrm{tr}(\boldsymbol{A}) \,, \quad II_A = \mathrm{tr}(\boldsymbol{A}^2) \,, \quad \text{and} \quad III_A = \mathrm{tr}(\boldsymbol{A}^3) \,. \tag{21}$$

If two tensors share the same eigenvalues one can equate the principal invariants of one to the other. Thus, for isotropic elastic materials it can be shown that the energy density can only depend on deformation gradient through the three invariants of $\boldsymbol{B}$ (or equivalently $\boldsymbol{C}$). In this manner the energy density of an elastic material can be expressed as some function,

$$e = e_B(I_B, II_B, III_B) \,. \tag{22}$$

There are many choices of finite deformation strain tensors that are consistent with the linear elastic strain tensor in the infinitesimal limit. These are typically expressed in terms of the stretch tensors, $\boldsymbol{U}$ or $\boldsymbol{V}$, or the Cauchy-Green deformation tensors, $\boldsymbol{C}$ or $\boldsymbol{B}$. We choose to use the log-strain tensor, also called Hencky strain.[12] This particular choice of strain tensor has some desirable features when compared with real materials at moderate strains[13, 14] and is often considered to behave similarly at moderate strains to an infinitesimal strain measure. The log-strain tensor $\boldsymbol{L}$ is defined as,

$$\boldsymbol{L} \equiv \ln \boldsymbol{V} = \frac{1}{2} \ln \boldsymbol{B} \,. \tag{23}$$

Scheidler[15] considered in depth the log-strain tensor and some of its properties. Notably that $\mathrm{tr}\, \boldsymbol{L} = \ln J$ and hence is a measure of the volumetric strain. Furthermore, the deviatoric part, $\boldsymbol{L}^*$, of $\boldsymbol{L}$ is independent of J and is a measure of shear strain. Since $I_L = \mathrm{tr} L = \ln J$ and $\boldsymbol{L}^*$ is traceless, the energy density is some function e'_L of another set of invariants:

$$e = e_L(I_L, II_L, III_L) = e'_L(\ln J, II_{L^*}, III_{L^*}) \,. \tag{24}$$

Scheidler[15] using the log-strain tensor, also showed that the following general relationships for an isotropic hyperelastic material can be derived for the pressure and the deviatoric stresses:

$$p = -\left.\frac{\partial e(J, \boldsymbol{L}^*)}{\partial J}\right|_{\boldsymbol{L}^*} \,, \quad \text{and} \tag{25a}$$

$$J\,\boldsymbol{T}^* = \left.\frac{\partial e(J, \boldsymbol{L}^*)}{\partial \boldsymbol{L}^*}\right|_{J} \,. \tag{25b}$$

The log-strain tensor is work conjugate to the Kirchoff stress, $J\boldsymbol{T}$. Work conjugacy follows from a general kinematic relationship, from which it is shown the diagonal components of the time derivative of the log-strain tensor are equal to the diagonal components of the rate of deformation tensor in the principal basis for the left-stretch tensor[16] and the assumption of an isotropic elastic material. An isotropic elastic implies that the Cauchy stress is also diagonal in the principal basis for the left-stretch tensor, ensuring that the inner product of the Cauchy stress and the rate of deformation tensor is equal to the inner product of the Cauchy stress and the time derivative of the log-strain tensor.

Only the second and third principal invariants of $\boldsymbol{L}^*$ are nonzero, so that the deviatoric stress can

be written:

$$\boldsymbol{T}^* = \frac{2}{J}\left(\frac{\partial e}{\partial II_{L^*}}\right)_{J,III_{L^*}} \boldsymbol{L}^* + \frac{3}{J}\left(\frac{\partial e}{\partial III_{L^*}}\right)_{J,II_{L^*}} [(\boldsymbol{L}^*)^2]^* \,. \tag{26}$$

The deviatoric stresses depend linearly on the deviatoric strains through the second invariant, but their dependence on the third invariant is quadratic in the deviatoric strains.[15] Thus, in the small shear-strain limit and when the shear modulus is independent of density, one can conclude:

$$\boldsymbol{T}^* \approx \frac{2\mu}{J}\boldsymbol{L}^* \,, \quad \text{and} \tag{27a}$$

$$p \approx p(J) \,. \tag{27b}$$

Assuming that the pressure depends only on the Jacobian J, Scheidler[15] showed that the strain energy can be decoupled additively into a function of J only and a function of shear only,

$$e = e_{\text{vol}}(J) + e_{\text{iso}}(\boldsymbol{L}^*) \,. \tag{28}$$

Taking both of these assumptions—specifically, that the pressure depends only on the Jacobian, and the deviatoric stress depends linearly on the deviatoric strains—is an enabling step in the formulation of the finite deformation analog to the linear elastic mechanochemical model. This formulation is similar to the linear elastic theory since we have effectively eliminated the pressure-shear coupling that would otherwise exist.

3.2 Mechanochemical Constitutive Model

We now define the decoupled energy density for the mechanochemical model. As before, we take the energy density per reference volume and additively decouple it into a mechanical and chemical part:

$$e = e_{\text{mechanical}} + e_{\text{chemical}} \,, \tag{29}$$

and express the mechanical energy in terms of a damage function ϕ and the elastic energy:

$$e_{\text{mechanical}} = \phi(\kappa)\, e_{\text{elastic}} \,. \tag{30}$$

The elastic energy is now given by equation 28 and the total energy is

$$e = \phi(\kappa)\,[e_{\text{vol}}(J) + e_{\text{iso}}(\boldsymbol{L}^*)] + e_{\text{chem}}(\kappa) \,. \tag{31}$$

We assume the equation of state $e_{\text{vol}}(J)$ is given by:

$$e_{\text{vol}}(J) = \mathcal{K}\left(J \ln J - J + 1\right) , \tag{32}$$

where $\mathcal{K}$ is the bulk modulus; other equations of state could be used in the future.[†] The energy associated with the deviatoric deformation is taken to be

$$e_{\text{iso}}(\boldsymbol{L}^*) = \mu \,\text{tr}\left[(\boldsymbol{L}^*)^2\right] \tag{33}$$

so that the total energy density function is given by

$$e = \phi(\kappa)\left\{\mathcal{K}\left(J \ln J - J + 1\right) + \mu \,\text{tr}\left[(\boldsymbol{L}^*)^2\right]\right\} + \frac{\xi}{2}\left(\kappa - \kappa^o\right)^2 . \tag{34}$$

Equation 34 is the analog of the Kachanov-Lifshitz function from equation 9 consistent with finite deformations. Using equation 25 and assuming that the shear strains are small gives

$$p = -\phi(\kappa)\mathcal{K} \ln J , \quad \text{and} \tag{35}$$

$$\boldsymbol{T}^* = \phi(\kappa)\frac{2\mu}{J}\,\boldsymbol{L}^* . \tag{36}$$

Thus, the total Cauchy stress tensor is

$$\boldsymbol{T} = \phi(\kappa)\left[\mathcal{K} \ln J \boldsymbol{I} + \frac{2\mu}{J}\,\boldsymbol{L}^*\right] . \tag{37}$$

The dynamical system analogous to equations 13 for the finite deformation model is

$$\rho\frac{\partial^2 \boldsymbol{u}}{\partial t^2} = \nabla \cdot \boldsymbol{T} , \quad \text{and} \tag{38a}$$

$$-\frac{1}{K}\frac{\partial \kappa}{\partial t} = \frac{\partial \phi}{\partial \kappa}\left(e_{\text{elastic}}(J, \boldsymbol{L}^*)\right) + \xi\left(\kappa - \kappa^o\right) . \tag{38b}$$

Note that e_{elastic} again refers to the elastic energy associated with a hypothetical undamageable material, recall equation 30. Again there has been no restriction on the sign of the derivative of $\dot{\kappa}$ so that the damage is reversible.

[†]Typically the $(J \ln J - J + 1)$ term in the expression for the energy that relates to the volumetric contribution does not include the "+1". In fact, in a hyperelastic model, the functional form of the energy is what is most important within an arbitrary constant since the stresses depend on derivatives of the energy. However, since we now would like to relate this energy to a damage growth mechanism, we must ensure that there is no damage growth in the undeformed state from a non-zero energy (i.e., setting $J = 1$ must give a zero energy).

4. Implementing the Damage Evolution Equations

In this section we discuss how the coupled dynamic system given by equation 38 is handled in the code and how it is different from the case where the left hand sides of equation 38 are identically zero, where the damage develops instantaneously. Implementation of the model is in SIERRA SolidMechanics (Presto/Adagio 4.28; Sandia National Laboratory) a Lagrangian finite element solver. SIERRA contains both an explicit solver Presto and an implicit, nonlinear preconditioned conjugant gradient solver, Adagio. The implicit code, Adagio, can run into some difficulties since the solution can become nonunique as the damage increases. In other words, there can be multiple deformations that correspond to the same stress, and the solver will no longer converge. Thus, while the code works with Adagio, its use is not recommended. Since SIERRA is a solid mechanics solver, the damage model must be introduced without disrupting the numerical stability. Therefore, there are some additional assumptions discussed here that are needed to solve the damage evolution alongside the mechanics.

4.1 Dynamic Evolution of Damage

As mentioned in section 2, there are dimensions of time in the kinetic reaction law parameter K. Thus, there are two time scales that need to be considered when solving the equations of motion. The first time scale is related to wave propagation speeds and the characteristic dimensions of the body and the second is how quickly damage accumulates. The stability of the explicit code depends critically on the elastic wave speeds and the minimum element size, which dictates the time step used during the calculations. This places some additional requirements on the calculation of the kinetic reaction law since it too needs to be updated at each time step. If the kinetic reaction law is also implemented in an explicit manner, there would be a second critical time step. Since the solutions to the kinetic reaction law are exponentials (see appendix A), it could become an incredibly stiff system to solve, i.e., sensitive to error. We therefore solve the kinetic reaction law for an incremental change in time so that we can directly solve for the updated damage at each time step.

We find that during a small time interval $t \to t + \Delta t$, the updated damage $\kappa(t + \Delta t)$ can be related to a convolution of the undamaged elastic energy with a derivative of the damage function (see appendix A):

$$\kappa(t+\Delta t) = \kappa(t)e^{-K\xi\Delta t} - \kappa^o\left(e^{-K\xi\Delta t} - 1\right) - K\int_t^{t+\Delta t} \frac{\partial\phi}{\partial\kappa}(\tau)\; e_{\text{elastic}}(\tau)e^{-K\xi(t+\Delta t-\tau)}\mathrm{d}\tau\,. \quad (39)$$

We assume that Δt is chosen by SIERRA such that the elastic energy is varying slowly. This would appear to be a safe assumption since the time step is chosen for stability. Thus, we pull the elastic energy out of the integral. If the damage function is given by equation 5, we can also pull the derivative of the damage function out of the integral and evaluate the result to arrive at the following:

$$\kappa(t+\Delta t) \approx \kappa(t) e^{-K\xi\Delta t} - \kappa^o \left(e^{-K\xi\Delta t} - 1\right) + \frac{1}{\xi}\left(1 - e^{-K\xi\Delta t}\right) \frac{1 - c_{min}}{\kappa^*}\, e_{\text{elastic}}\,. \tag{40}$$

The updated damage κ^{n+1} is then:

$$\kappa^{n+1} = \kappa^n e^{-K\xi\Delta t} - \kappa^o \left(e^{-K\xi\Delta t} - 1\right) + \frac{1}{\xi}\left(1 - e^{-K\xi\Delta t}\right) \frac{1 - c_{min}}{\kappa^*} \left(\mathcal{K}\left(J\ln J - J + 1\right) + \mu\,\text{tr}\left[(\boldsymbol{L}^*)^2\right]\right)^{n+1}\,, \tag{41}$$

where superscripts denote the time step. This updated damage can then be fed into the stress calculation:

$$\boldsymbol{T} = \phi(\kappa)\left(\mathcal{K}\ln J\boldsymbol{I} + \frac{2\mu}{J}\boldsymbol{L}^*\right)\,. \tag{42}$$

The minor alteration to the code to handle the case where the damage is irreversible is discussed in section 7.1.

4.2 Instantaneous Damage

A special case of damage evolution is when the left hand side of equation 38b is zero. This can be achieved by taking the limit as K goes to infinity in equation 40, or by directly solving equation 38b,

$$\kappa = \kappa^o + \frac{1}{\xi}\frac{1 - c_{min}}{\kappa^*}\left(\mathcal{K}\left(J\ln J - J + 1\right) + \mu\,\text{tr}\left[(\boldsymbol{L}^*)^2\right]\right)\,. \tag{43}$$

4.3 Additional Details of the Implementation

SIERRA provides the left-stretch tensor $\boldsymbol{V}$ and rotation tensor $\boldsymbol{R}$ from the polar decomposition of the deformation gradient tensor $\boldsymbol{F}$ at the so-called next time step $n+1$. From this updated configuration, one must determine the Cauchy stress in the unrotated (reference configuration). As discussed in section 3, the model depends on the log-strain, which requires taking the natural logarithm of a tensor. We use built-in functions in SIERRA to perform the spectral decomposition of $\boldsymbol{V}$, from which the log can be taken. The overall algorithm is outlined as follows:

1. Calculate J and $\ln J$ from $\boldsymbol{V}^{n+1}$.
2. Determine eigenvalues/vectors for $\boldsymbol{V}^{n+1}$.

3. Calculate the log of the eigenvalues and rotate $\log \boldsymbol{V}^{n+1}$ from its eigenbasis back to current basis.

4. Calculate $(\boldsymbol{L}^*)^{n+1}$ from $\ln \boldsymbol{V}^{n+1}$.

5. Update the current undamaged elastic energy, e^{n+1}_{elastic}.

6. Solve for the new damage value κ^{n+1}, restrict its range to $0 < \kappa < \kappa^*$ using equation 41 or 43 as selected by the user.

7. Determine the value of the damage function $\phi(\kappa^{n+1})$.

8. Determine $\boldsymbol{T}^{n+1}$ from equation 42.

9. "Unrotate" $\boldsymbol{T}^{n+1}$ to reference configuration by calculating $\boldsymbol{R}^T\boldsymbol{T}\boldsymbol{R}$.

10. Repeat steps 1–9 for all elements in the material.

5. Model Verification

This section compares theoretical predictions of the model against simulation results from single element tests. The three tests considered here are a compression test where the volume is changed and the damage evolves instantaneously to the equilibrium state so that equation 43 is satisfied. The second test is a shear test where the volume does not change and again the damage is always instantly updated. The last test is a compression test where the damage evolution is time dependent, i.e., according to equation 41. In all tests, we chose $\mathcal{K} = 1$ Pa, and $\mu = 0.75$ Pa.

5.1 Instantaneous Damage Evolution, Confined-Compressive Deformation

We compare the results of the simulation on a single element for a confined compression test. The physical components of the deformation gradient $\boldsymbol{F}$, and the log-strain tensor $\ln \boldsymbol{V}$ imposed are

$$[\boldsymbol{F}] = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad [\ln \boldsymbol{V}] = \begin{bmatrix} \ln \alpha & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{44}$$

The Jacobian $J = \det\boldsymbol{F} = \alpha$, so that the deviatoric part of the log-strain tensor is

$$[\boldsymbol{L}^*] = \begin{bmatrix} \frac{2}{3}\ln\alpha & 0 & 0 \\ 0 & -\frac{1}{3}\ln\alpha & 0 \\ 0 & 0 & -\frac{1}{3}\ln\alpha \end{bmatrix}. \tag{45}$$

From J and $\boldsymbol{L}^*$ we can solve for the damage, i.e., solving equation 43 using the deformation given in equation 44. The solution to the damage is

$$\kappa = \min\left[1, \kappa^0 - \frac{\mu}{\xi}\left(\mathcal{K}\left(\alpha\ln\alpha - \alpha + 1\right) + \frac{4\,\mu\ln\alpha}{3}\right)\left(c_{min} - 1\right)\right]. \tag{46}$$

This results in the magnitude of the xx-component of the Cauchy Stress to be given by

$$T_{xx} = \left(\mathcal{K}\ln\alpha + \frac{4\mu\ln\alpha}{3\alpha}\right)\left(\min\left[1, \kappa^0 - \frac{\mu}{\xi}\left(\mathcal{K}\left(\alpha\ln\alpha - \alpha + 1\right) + \frac{4\mu\ln\alpha}{3}\right)\left(c_{min} - 1\right)\right]\left(c_{min} - 1\right) + 1\right). \tag{47}$$

The results of the theoretical predictions are shown in figure 1 as solid lines and the symbols are values taken from simulation results. Panels *a* and *c* plot the damage κ as a function of the strain $\boldsymbol{L}^*_{xx}$, and panels *b* and *d* plot the xx-component of stress as a function of the strain $\boldsymbol{L}^*_{xx}$. In these plots, $\kappa^0 = 0.02$ and $\kappa^* = 1$. The top two plots explore the result of holding $\mu/\xi = 0.1$ and varying c_{min} from 0.2 to 1, while the bottom two plots explore the result of holding $c_{min} = 0.3$ and varying ξ so that $\mu/\xi = 0.2$ to 0.1. In all cases the simulation results agree completely with the theoretical predictions.

The stress-strain curves in panels *b* and *d* show the key feature required for the physical instability. Curves that exhibit a zero slope in the stress followed by a decrease in stress at increased strain are candidates for exploring the physical instability. We leave this largely without discussion in this work since it will be developed further in subsequent work.

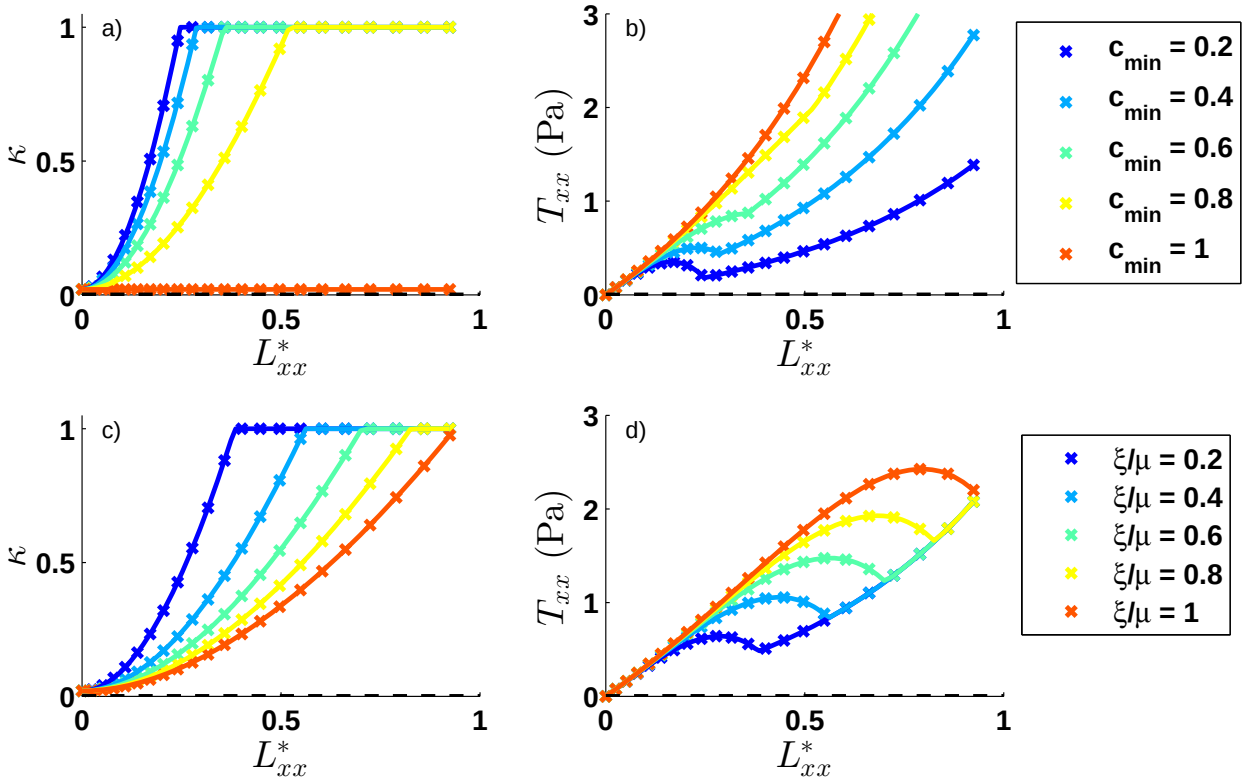

Fig. 1 Single element compression test. Comparison of the theoretical prediction (solid lines) against the simulation results (symbols). Panels *a* and *b* hold $\xi/\mu = 0.1$ and vary c_{min} from 0.2 to 1. Panels *c* and *d* hold $c_{min} = 0.3$ and vary ξ/μ from 0.2 to 1. Stress made positive for viewing purposes although its value is negative in compression..

5.2 Instantaneous Damage Evolution, Isochoric Shear Deformation

Here we compare the results of the simulation on a single element for an isochoric shear deformation in the Y-direction with instantaneous damage evolution. The physical components of the deformation gradient $\boldsymbol{F}$ and left-stretch tensor $\boldsymbol{V}$ are

$$[\boldsymbol{F}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \alpha \\ 0 & 0 & 1 \end{bmatrix}, \qquad [\boldsymbol{V}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{2+\alpha^2}{\sqrt{4+\alpha^2}} & \frac{\alpha}{\sqrt{4+\alpha^2}} \\ 0 & \frac{\alpha}{\sqrt{4+\alpha^2}} & \frac{2}{\sqrt{4+\alpha^2}} \end{bmatrix}. \tag{48}$$

Using MuPad (MathWorks), one can calculate the deviatoric log-strain tensor $\boldsymbol{L}^*$, the damage κ, and the stress $\boldsymbol{T}$ from the previous expressions for $\boldsymbol{F}$ and $\boldsymbol{V}$. Since the explicit forms of these variables are quite complicated, they are not reproduced here. Instead, figure 2 compares the simulation results (symbols) against the theoretical prediction (solid lines). This figure is very similar to the previous one with the exception that the component of the stress and strain that is plotted is the yz-component. The figure clearly shows agreement between theory and simulation and that the physical instability can manifest itself in shear as well.[†]

[†]The level of shear in figure 2 likely exceeds the range of validity for equation 27a.

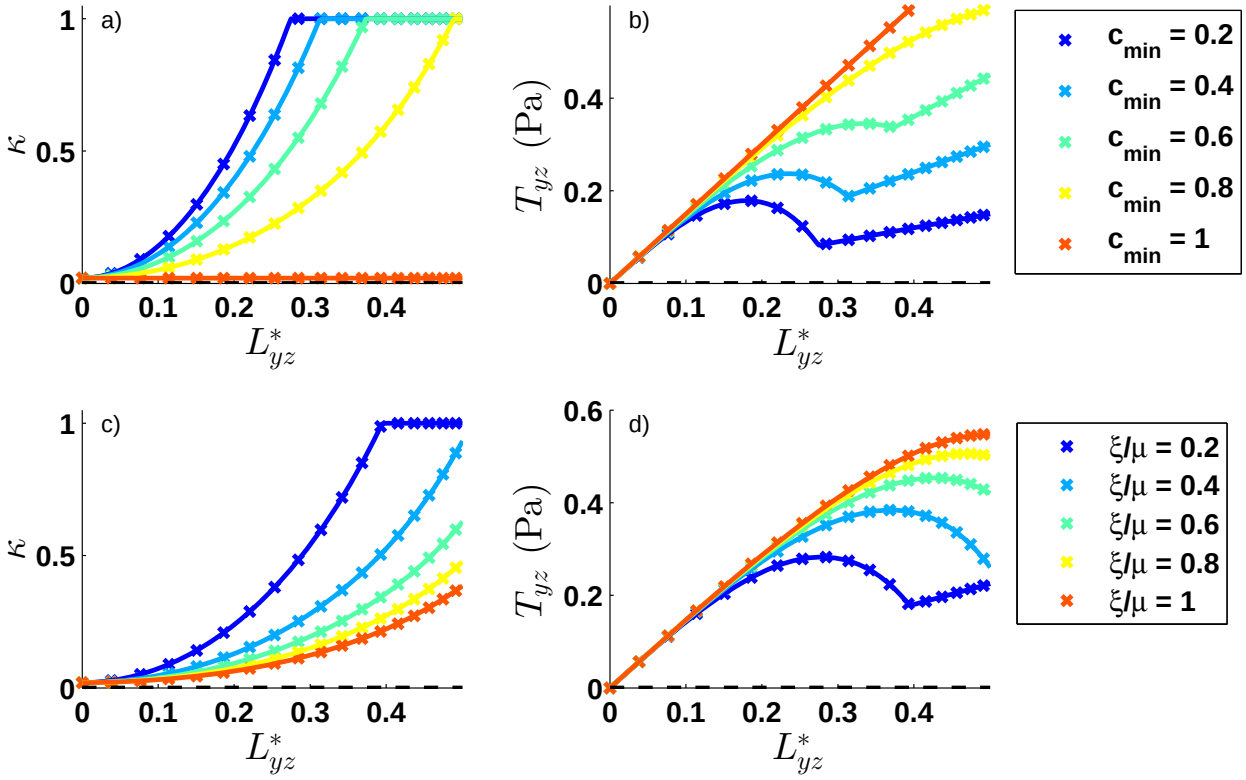

Fig. 2 Single element shear test. Comparison of the theoretical prediction (solid lines) against the simulation results (symbols). Panels *a* and *b* hold $\xi/\mu = 0.1$ and vary c_{min} from 0.2 to 1. Panels *c* and *d* hold $c_{min} = 0.3$ and vary ξ/μ from 0.2 to 1..

5.3 Dynamic Damage Evolution, Confined-Compressive Deformation

The last verification step presented here considers the time-dependent damage evolution. We impose a very rapid compression, identical to that in section 5.1 except that the damage is now evolved dynamically according to equation 41. Here the initial value of the damage is equal to the equilibrium damage, i.e., $\kappa(0) = \kappa^0 = 0.02$, and the kinetic coefficient K = 0.0001 (1/Pa·s).

For the following analytical calculation, we ignore the initial transient behavior. We assume the body is initially deformed and that the damage starts off at the damage associated with minimum energy $\kappa(0) = \kappa^0$. Under these conditions we find the solution to the damage evolution to be

$$\kappa(t) = \min\left[1, \kappa(0)\mathrm{e}^{-K\,t\,\xi} - \kappa^0\,\left(\mathrm{e}^{-K\,t\,\xi} - 1\right) - \frac{\mu\,(c_{min} - 1)}{\xi}\left(1 - \mathrm{e}^{-K\,t\,\xi}\right)\left(\frac{4}{3}\mu\log\alpha + \mathcal{K}\,(\alpha\log\alpha - \alpha + 1)\right)\right] \tag{49}$$

with a corresponding magnitude of the xx component of the stress evolution given by:

$$T_{xx} = \left(\mathcal{K}\log\alpha + \frac{4\mu\log\alpha}{3\alpha}\right)\phi(\kappa(t))\,. \tag{50}$$

Figure 3 shows good agreement between the simulation and the theoretical prediction. The only discrepancies arise from the initial transient behavior that we ignored (see the symbols near the vertical axis panels *b* and *d*).

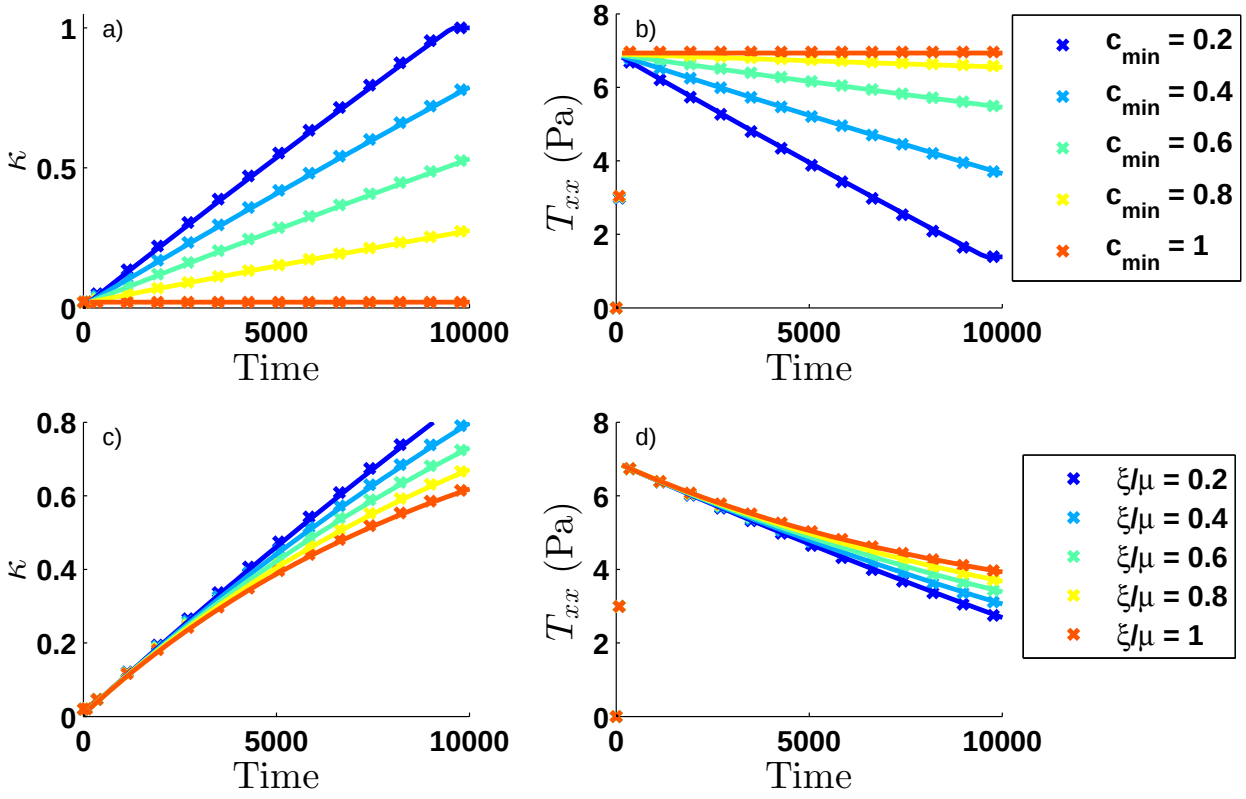

Fig. 3 Single element dynamic compression test. Comparison of the theoretical prediction (solid lines) against the simulation results (symbols). Panels *a* and *b* hold $\xi/\mu = 0.1$ and vary c_{min} from 0.2 to 1. Panels *c* and *d* hold $c_{min} = 0.3$ and vary ξ/μ from 0.2 to 1..

6. Example Problems

The previous sections introduced the equations of motion and the verifications of the model's implementation. Here, we present some examples of how the model could be used.

6.1 One-Dimensional Dynamic Confined-Compression

In this section we consider a one-dimensional (1-D) problem of a linear alignment of material that is slowly (compared to the elastic wave speeds) compressed until it becomes unstable. This problem is investigated by fixing one end of the material at $Z = L$ while the end at $Z = 0$ is slowly displaced according to the function:

$$\delta(t) = \begin{cases} \delta_{max}\frac{t}{t_c} & : t \leq t_c \\ \delta_{max} & : t > t_c \end{cases} . \tag{51}$$

This displacement function has a discontinuity in its derivative at 0 and at t_c. The dimensions and material parameters used for this simulation are presented in table 1.

Table 1 Summary of material parameters used in the 1-D stability example..

L (m)	ρ (kg/m^3)	$\mathcal{K}$ (Pa)	μ (Pa)	ξ (Pa)	K (1/Pa·s)	c_{min}	κ^0	κ^*	t_c (s)	δ_{max} (m)
1	1e6	10	7.5	1	0.1	0.3	0.02	1	15000	0.29

Figure 4 shows the results of this simulation. Each panel of the figure plots the damage distribution versus location for a given time. The difference in time between each panel is not the same. At first, in panels *a–e*, the damage evolution is similar to that of what we would expect in the quasi-static case, i.e., it is proportional to the strain applied to the system and shows no localization of damage. However, in panels *f* and *g* the damage has started to accumulate at the end that is displaced. This localization of damage culminates by panels *h* and *i* where the saturation of damage to κ^* in panel *i* sends off a relaxation wave in the damage. This wave propagates toward the fixed end ($Z = L$) where a reflection of the wave at the boundary causes two additional localizations of damage evolution, panels *j* and *k*. At this point the damage evolution slows down as the system reaches a new stable equilibrium, panels *l–o*.

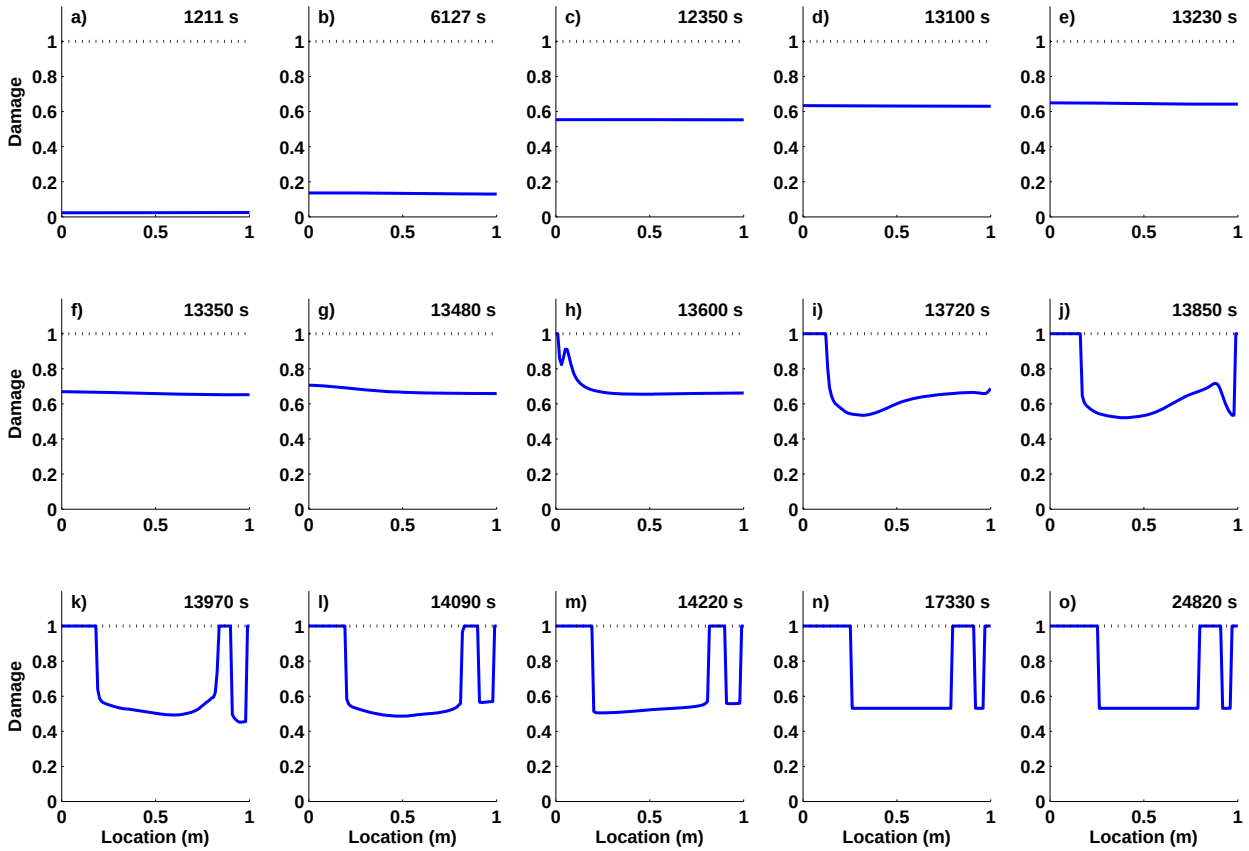

Fig. 4 A One-Dimensional alignment is slowly compressed from its $Z = 0$ end. Each panel plots the damage vs. position within the alignment, where the plots are on the initial configuration. The time for each panel is shown in the upper right corner and the time difference between each panel is not constant..

One subtle feature of the constitutive model is that the chemical energy term introduces a chemical potential that drives the damage toward κ^0. This means that the material can effectively heal and that the damage is reversible. This feature is present in this example and can be seen by comparing panels *e* and *o*. In panel *e*, the material is fairly uniformly damaged with a value slightly larger than 0.6. In panel *o*, the material is split into fully damaged material $\kappa = 1$ and material that is only partially damaged $\kappa \approx 0.5$. In some cases this self-healing feature may be undesirable, and a modification to the model is discussed in section 7.1.

6.2 Two-Dimensional Radial Damage Bands

In this section we consider a two-dimensional (2-D) problem of an annular disc that is slowly (compared to the elastic wave speeds) compressed until it becomes unstable and forms radial damage bands. This is similar to the problem considered by Grinfeld et al.,[5] except that the mechanical system is not evolved quasi-statically. The outer radius $R_o = 1$ of an annular disc is

slowly displaced radially toward the center of the disc according to the piecewise function given in equation 51. The inner boundary at $R_i = 0.1$ is stress free and the disc is fixed in the axial coordinate. This type of deformation concentrates stresses near the inner radius. The simulation used the material parameters given in table 2. Because of the choices of material parameters and the imposed deformation, the maximal displacement of the inner radius is only 0.004 m towards the center.

Table 2 Summary of material parameters used in the 2-D radial damage band instability example..

R_i (m)	R_o	ρ (kg/m^3)	$\mathcal{K}$ (kPa)	μ (kPa)	ξ (Pa)	K (1/Pa·s)	c_{min}	κ^0	κ^*	t_c (s)	δ_{max} (m)
0.1	1	1e6	10	7.5	1	0.1	0.3	0.02	1	3000	0.05

Each panel in figure 5 shows the damage κ in false-color plotted on the initial configuration throughout time. The time elapsed between each panel is 175 s. Initially, the damage uniformly accumulates throughout the material. This can be seen by noting the uniform color distribution in panels *a–f*. The solution to this problem without damage is rotationally symmetric and concentrates energy at the small opening. This gives insight into why the damage begins to localize near the center and where the material instability is first manifested. The accumulation of damage causes a local weakening near the opening of the disc and the germination of radial damage bands (panels *g* and *h*)[†]. Six radial damage bands clearly form (panel *i*) leaving the surrounding material intact. The local weakening around the damage bands causes a strain concentration ahead of the existing band. This subsequently causes the instability to propagate and causes the band to grow predominantly in the radial direction as can be seen in panels *h–m*. The length of the band grows at a finite velocity, where one can estimate the speed of the tip of the vertical ($90°$) band in panels *h* and *l*, 0.001 m/s. The damage is arrested as the stress is alleviated and the system finds a new stable configuration. No additional damage accumulates in panels *m–o*. The damage bands alleviate hoop stresses that are concentrated at the center opening and that develop from the imposed deformation. In future developments it may be necessary to introduce a mechanism for hoop damage bands to develop from radial instabilities. Two possible alterations are discussed in sections 7.1 and 7.2.

[†]In these damage bands the material is still intact. In an abstract way these bands can be thought of as a proxy for radial cracks.

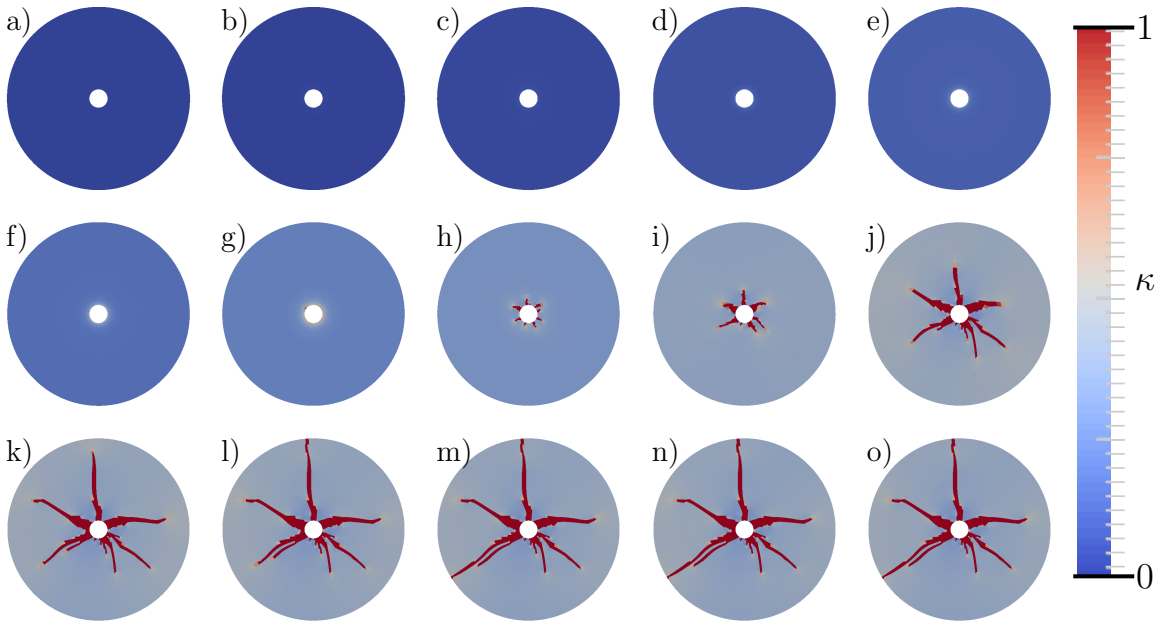

Fig. 5 Two-Dimensional instability and the development of radial damage bands. A 2-D annular disc is slowly compressed from its outside edge $r = R_o$. Each panel plots the damage in false-color as it varies with position within the disc. Plots are on the initial configuration. The elapsed time between each panel is 175 s. At first the damage accumulates symmetrically before a material instability produces the localization of damage along radial lines..

7. Future Considerations

In the previous sections we introduced and verified the implementation of a mechanochemical model for failure. In this section we consider some possible alternatives to the choices of the damage function ϕ and the energy.

7.1 Removing Healing From the Mechanochemical Model

In section 6.1 we commented that the quadratic term in the chemical energy can result in a healing process. This healing process is driven by the chemical potential and may not be physically realistic. Analogous to Grinfeld and Wright,[10] we now restrict the sign of $\dot{\kappa}$ so that damage is irreversible. Here we consider some ways to modify the current model and implementation to correct for this feature.

The kinetic evolution of the damage is given by the rate equation 38b. Repeated here for clarity:

$$-\frac{1}{K}\frac{\partial \kappa}{\partial t} = \frac{\partial \phi}{\partial \kappa}\left(e_{\text{elastic}}(J, \boldsymbol{L}^*)\right) + \xi\left(\kappa - \kappa^o\right) . \tag{52}$$

For our specific choice of ϕ (and likely in more general cases), the derivative of the damage function with respect to the damage is negative. Since $e_{\text{elastic}} > 0$, $\dot{\kappa}$ will be negative if

$$e_{\text{elastic}}(J, \boldsymbol{L}^*) < \frac{\xi\kappa^*\left(\kappa - \kappa^o\right)}{1 - c_{min}} . \tag{53}$$

This would certainly become an issue in a test where the loading is applied very rapidly and there are elastic waves propagating back and forth. A wave propagating past would initially cause damage, but as the wave continued on, the damaged material might be in a lower elastic energy state and begin to self-heal (recall the results from section 6.1).

We can quite trivially remove the self-healing feature. The algorithm outlined in section 4 calculates the mechanical energy associated with the deformation. This energy could then used to evaluate the inequality in equation 53. If the inequality holds true, then applying the traditional update equation 41 would result in a smaller damage value in the next time step. However, instead of using the update equation, one could set $\partial\kappa/\partial t = 0$, and

$$\kappa^{n+1} = \kappa^n . \tag{54}$$

This type of modification might be sufficient to observe radial and hoop damage bands a dynamic test where healing would have been an issue before.

7.2 Decoupling Bulk Damage and Deviatoric Damage Functions

Here, we consider a simple extension to the current model where the only modification is that there are two damage functions that can affect the volumetric and isochoric responses separately:

$$e = \phi_1(\kappa) e_{\text{vol}}(J) + \phi_2(\kappa) e_{\text{iso}}(\boldsymbol{L}^*) + e_{\text{chem}}(\kappa)\,. \tag{55}$$

The stress is no longer given by equation 42, but instead

$$\boldsymbol{T} = \phi_1(\kappa)\mathcal{K} \ln J \boldsymbol{I} + \phi_2(\kappa)\frac{2\mu}{J}\boldsymbol{L}^*\,. \tag{56}$$

The damage still evolves in time; however, derivatives of ϕ_1 and ϕ_2 with respect to κ will enter into the kinetic equation separately:

$$-\frac{1}{K}\frac{\partial \kappa}{\partial t} = \frac{\partial \phi_1}{\partial \kappa} e_{\text{vol}}(J) + \frac{\partial \phi_2}{\partial \kappa} e_{\text{iso}}(\boldsymbol{L}^*) + \xi\left(\kappa - \kappa^o\right)\,. \tag{57}$$

This would be particularly useful if the material was more sensitive to damage in shear than in its bulk response. One possible choice of damage functions is

$$\phi_1(\kappa) = 1 - (1 - c_{min})\frac{\kappa}{\kappa^*} \quad \text{and} \tag{58}$$

$$\phi_2(\kappa) = \max\left[0, 1 - \alpha\frac{\kappa}{\kappa^*}\right]\,. \tag{59}$$

A dimensionless parameter $\alpha > 0$ sets how rapidly the damage function ϕ_2 drops to zero.* The maximum operator is needed here since the original notion of κ^* is lost by introducing α which would otherwise make ϕ_2 negative when $\alpha\kappa > \kappa^*$. In this example c_{min} is dropped from ϕ_2 so that the material completely loses its deviatoric response and behaves more like a soft, nearly incompressible material when it is maximally damaged.

An alternate approach to this type of modification would be to introduce a second damage parameter κ_2, with its own damage evolution and maximal damage κ_2^*. In this case there would be two internal state variables and two damage evolution equations $\dot{\kappa}_i$ to solve. This also would require the notion that damage carries a direction instead of simply being a scalar.

*There is still no restriction on the sign of the time derivative of the damage. Thus, bulk and deviatoric damage can still heal and that α merely adjusts the rate of damage accumulation to be preferentially bulk or preferentially shear.

7.3 Damage Growth Asymmetry in Compression and Tension

We introduce a pathway to add an asymmetric growth of the damage in tension and compression. This type of model might be useful to ensure that a material does not lose strength in compression and to promote damage growth in tension. The modified energy density is

$$e = \phi(\kappa)\left[\alpha_t H(J-1) + \alpha_c H(1-J)\right] e_{\text{vol}}(J) + \phi(\kappa) e_{\text{iso}}(\boldsymbol{L}^*) + e_{\text{chem}}(\kappa)\,, \tag{60}$$

where H is a Heaviside function:

$$H(x) = \begin{cases} 1 & : x \geq 0 \\ 0 & : x < 0 \end{cases}\,. \tag{61}$$

The construction in the square brackets in equation 60 acts as a switch between the two energy scale factors α_t and α_c. $H(J-1)$ takes on the value 1 when the material is in tension, and if $\alpha_t > \alpha_c$, the energy is increased by $\alpha_t - \alpha_c$. Assuming this form of the energy density gives the desired behavior in the kinetic law

$$-\frac{1}{K}\frac{\partial \kappa}{\partial t} = \frac{\partial \phi}{\partial \kappa}\left[\alpha_t H(J-1) + \alpha_c H(1-J)\right] e_{\text{vol}}(J) + \frac{\partial \phi}{\partial \kappa} e_{\text{iso}}(\boldsymbol{L}^*) + \xi\left(\kappa - \kappa^o\right)\,. \tag{62}$$

The deviatoric stress is

$$J\,\boldsymbol{T}^* = \left.\frac{\partial e}{\partial \boldsymbol{L}^*}\right|_J = 2\mu\,\boldsymbol{L}^*, \tag{63}$$

and the pressure can be obtained from applying equation 25a to equation 60,

$$p = -\left.\frac{\partial e}{\partial J}\right|_{\boldsymbol{L}^*} = -\frac{\partial}{\partial J}\left(\left[\alpha_t H(J-1) + \alpha_c H(1-J)\right] \mathcal{K}\left(J \ln J - J + 1\right)\right)\,. \tag{64}$$

Evaluating the derivatives and simplifying gives:

$$p = \left[\alpha_t H(J-1) + \alpha_c H(1-J)\right] \mathcal{K} \ln J\,. \tag{65}$$

Now the slope of the pressure has a discontinuity as we cross $J = 1$ from above or below:

$$\frac{\partial p}{\partial J} = \left[\alpha_t H(J-1) + \alpha_c H(1-J)\right] \mathcal{K}\frac{1}{J} + \left[\alpha_t\,\delta(J-1) - \alpha_c\,\delta(1-J)\right] \mathcal{K} \ln J\,, \tag{66}$$

where

$$\lim_{J \to 1^-} \frac{\partial p}{\partial J} = \alpha_c\,\mathcal{K} \tag{67}$$

and

$$\lim_{J \to 1^+} \frac{\partial p}{\partial J} = \alpha_t \, \mathcal{K} \, . \tag{68}$$

This is only a minor concern since the derivative of the stress response already has a discontinuity from the damage function (see figures 1 and 2).

8. Conclusions

In conclusion we have developed a mechanochemical model for dynamic failure using a finite deformation formulation analogous to the model developed by Grinfeld and Wright.[1] We have conducted single element tests to verify the model is working. We have also outlined some example problems that the model was implemented to study. We also discussed some of the limitations and possible future alterations that could be made to the model.

9. References

1. Grinfeld, M. A.; Wright, T. W. Thermodynamics of Solids: Recent Progress With Applications to Brittle Fracture and Nanotechnology. In *23rd U.S. Army Science Conference, Orlando, FL*, 2002.

2. Grinfeld, M. A. Instability of the Interface Between a Nonhydrostatically Stressed Elastic Body and a Melt. *Akademiia Nauk SSSR Doklady* **1986**, *290*, 1358-1363.

3. Grinfeld, M. A. *Thermodynamic Methods in the Theory of Heterogeneus Systems*; Longman Scientific and Technical: New York, 1991.

4. Kassner, K.; Misbah, C.; Müller, J.; Kappey, J.; Kohlert, P. Phase-Field Modeling of Stress-Induced Instabilities. *Physical Review E.* **2001**, *63* (3), 036117.

5. Grinfeld, M. A.; McCauley, J. W.; Schoenfeld, S. E.; Wright, T. W. Failure Pattern Formation in Brittle Ceramics and Glasses. In *23rd International Symposium on Ballistics, April 2007*, Tarragona, Spain, 2007.

6. Kachanov, L. M. *Introduction to Continuum Damage Mechanics*; Springer: New York, 1986.

7. Chaboche, J. L. Continuum Damage Mechanics: Part I General Concepts. *Journal of Applied Mechanics* **1988**, *55* (1), 59–64.

8. Chaboche, J. L. Continuum Damage Mechanics: Part II Damage Growth, Crack Initiation, and Crack Growth. *Journal of Applied Mechanics* **1988**, *55* (1), 65–72.

9. Holzapfel, G. A. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*; John Wiley & Sons Ltd.: New York, 2000.

10. Grinfeld, M. A.; Wright, T. W. *Thermodynamics of Brittle Fracture*; ARL-TR-3659; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 2005.

11. Truesdell, C.; Noll, W. *The Non-Linear Field Theories of Mechanics*; Springer: New York, 2004.

12. Hencky, H. The Law of Elasticity for Isotropic and Quasi-Isotropic Substances by Finite Deformations. *Journal of Rheology* **1931**, *2*, 169.

13. Anand, L. On H. Hencky's Approximate Strain-Energy Function for Moderate Deformations. *Journal of Applied Mechanics* **1979**, *46*, 78.

14. Anand, L. Moderate Deformations in Extension-Torsion of Incompressible Isotropic Elastic Materials. *Journal of the Mechanics and Physics of Solids* **1986**, *34* (3), 293–304.

15. Scheidler, M. On the Coupling of Pressure and Deviatoric Stress in Hyperelastic Materials. In *Proceedings of the 13th Army Symposium on Solid Mechanics, Aug 1993*, Plymouth, MA, 1993.

16. Scheidler, M. Time Rates of Generalized Strain Tensors Part I: Component Formulas. *Mechanics of Materials* **1991**, *11* (3), 199–210.

Appendix A. Analytical Solution to Kinetic Equation

In the main text, we derived a kinetic law relating the evolution of the damage to the elastic energy of the material,

$$\frac{\partial \kappa}{\partial t} = -K\left(\frac{\partial \phi}{\partial \kappa} e_{\text{elastic}} + \xi\left(\kappa - \kappa^o\right)\right) . \tag{A-1}$$

For the special case discussed in the main text, equation A-1 readily integrates. This appendix considers a more general approach that could be used for other damage functions ϕ.

For notational simplicity we write,

$$\frac{\partial \phi}{\partial \kappa} = g(\kappa(t)) . \tag{A-2}$$

and rewrite equation A-1 as

$$\frac{\mathrm{d}\kappa}{\mathrm{d}t} = -K\left(g(\kappa(t))\, e_{\text{elastic}}(t) + \xi\left(\kappa - \kappa^o\right)\right) . \tag{A-3}$$

Recall that the elastic energy is the hypothetical elastic energy of an undamageable material, i.e., the energy that only depends on the displacements and the elastic moduli. In general, g might not be a simple function of the damage so a method of Laplace transforms is useful when attempting to solve this differential equation. In the following, $\mathcal{L}\left[f(t)\right] = F(s)$ denotes the Laplace transform of $f(t)$. Similarly, $\mathcal{L}^{-1}\left[F(s)\right] = f(t)$ denotes the inverse Laplace transform of $F(s)$.

Applying the Laplace transform to both sides of equation A-3 gives

$$s\mathcal{L}\left[\kappa(t)\right] - \kappa(0) = -K\mathcal{L}\left[g(\kappa(t))\, e_{\text{elastic}}(t)\right] - K\xi\mathcal{L}\left[\kappa(t)\right] + \frac{K\xi\kappa^o}{s} , \tag{A-4}$$

which is an algebraic equation that can be solved for $\mathcal{L}\left[\kappa(t)\right]$,

$$\mathcal{L}\left[\kappa(t)\right] = \frac{\kappa(0)}{s + K\xi} + \frac{1}{K\xi}\left(\frac{K\xi\kappa^o}{s} - \frac{K\xi\kappa^o}{s + K\xi}\right) - \frac{K\mathcal{L}\left[g(\kappa(t))\, e_{\text{elastic}}(t)\right]}{s + K\xi} . \tag{A-5}$$

Applying the inverse Laplace transform gives:

$$\kappa(t) = \mathcal{L}^{-1}\left[\frac{\kappa(0)}{s + K\xi}\right] + \mathcal{L}^{-1}\left[\frac{\kappa^o}{s}\right] - \mathcal{L}^{-1}\left[\frac{\kappa^o}{s + K\xi}\right] - \mathcal{L}^{-1}\left[\frac{K\mathcal{L}\left[g(\kappa(t))\, e_{\text{elastic}}(t)\right]}{s + K\xi}\right] , \tag{A-6}$$

$$\kappa(t) = \kappa(0)e^{-K\xi t} - \kappa^o\left(e^{-K\xi t} - 1\right) - K\mathcal{L}^{-1}\left[\mathcal{L}\left[g(\kappa(t))\, e_{\text{elastic}}(t)\right]\mathcal{L}\left[e^{-K\xi t}\right]\right] . \tag{A-7}$$

The last term is the inverse Laplace transform of the product of two Laplace transforms, which is the convolution of the two original functions:

$$\kappa(t) = \kappa(0)e^{-K\xi t} - \kappa^o\left(e^{-K\xi t} - 1\right) - K\int_0^t \frac{\partial \phi}{\partial \kappa}(\tau)\, e_{\text{elastic}}(\tau) e^{-K\xi(t-\tau)}\mathrm{d}\tau . \tag{A-8}$$

Since, in general, $\partial\phi/\partial\kappa$ will depend on κ, this is an integral equation for $\kappa(t)$. It can be used to update the damage parameter for any damage function as a convolution of $\partial\phi/\partial\kappa$ and e_{elastic}. Considering a small time step Δt, however, one can derive:

$$\kappa(t+\Delta t) = \kappa(t)e^{-K\xi\Delta t} - \kappa^o\left(e^{-K\xi\Delta t} - 1\right) - K\int_t^{t+\Delta t} \frac{\partial\phi}{\partial\kappa}(\tau)\, e_{\text{elastic}}(\tau)e^{-K\xi(t+\Delta t-\tau)}\mathrm{d}\tau\,. \quad \text{(A-9)}$$

If Δt is chosen for stability in the explicit codes, the change in elastic energy will likely be small over the time interval $t \to t+\Delta t$. Thus, we take the elastic energy outside of the convolution integral.* We also assume that the damage is not changing rapidly during this time interval so that $\partial\phi/\partial\kappa$ is slowly varying and can also be taken outside the integral. In the special case of ϕ given by equation 5 this holds trivially, since $\partial\phi/\partial\kappa$ is constant. Thus, we can write the incremental change in the damage as:

$$\kappa(t+\Delta t) \approx \kappa(t)e^{-K\xi\Delta t} - \kappa^o\left(e^{-K\xi\Delta t} - 1\right) - \frac{1}{\xi}\left(1 - e^{-K\xi\Delta t}\right)\frac{\partial\phi}{\partial\kappa}\, e_{\text{elastic}} \quad \text{(A-10)}$$

Thus, the updated damage $\kappa(t+\Delta t)$ depends on the current damage $\kappa(t)$ the equilibrium damage κ^o and the elastic energy.

Using a superscript n denotes that the variable is evaluated at the current time step, and $n+1$ the future time step gives

$$\kappa^{n+1} = \kappa^n e^{-K\xi\Delta t} - \kappa^o\left(e^{-K\xi\Delta t} - 1\right) - \frac{1}{\xi}\left(1 - e^{-K\xi\Delta t}\right)\left(\frac{\partial\phi}{\partial\kappa}\right)^n e_{\text{elastic}}^{n+1}\,. \quad \text{(A-11)}$$

When ϕ is given by equation 5,

$$\frac{\partial\phi}{\partial\kappa} = \frac{1-c_{min}}{\kappa^*}\,, \quad \text{(A-12)}$$

and

$$\kappa^{n+1} = \kappa^n e^{-K\xi\Delta t} - \kappa^o\left(e^{-K\xi\Delta t} - 1\right) + \frac{1}{\xi}\left(1 - e^{-K\xi\Delta t}\right)\frac{1-c_{min}}{\kappa^*}\, e_{\text{elastic}}^{n+1}\,. \quad \text{(A-13)}$$

In the text, the elastic energy term is used in two different ways. First, that of a linear elastic material and later, a hyper-elastic material.

*In future models where $\partial\phi/\partial\kappa$ is not constant, additional state variables may be needed in the constitutive model, which keeps track of past elastic energies and past damage values. These might be used in some Simpson quadrature rule to simplify the convolution so that it could be evaluated numerically.

INTENTIONALLY LEFT BLANK.

Appendix B. SIERRA Implementation

```
#include <models/BrittleElastic.h>
#include <cmath>
#include <kinematics/lame_general_methods.h>
#include <kinematics/LameUtil.h>
#include <kinematics/Kinematics.h>
using namespace std;
namespace lame {

 Material * BrittleElastic::createMaterial( const MatProps & props ) {
   return new BrittleElastic( props );
 }
 //*****************************************************************
 // There are three state variables (although only one changes right now)
 // The first two are basic, BULK_MODULUS and SHEAR_MODULUS
 // The third which evolves in time is the damage parameter KAPPA
 // To make this model work, however, you need to specify
 // XI_CHEMICAL damage chemical constant, K_KINETIC, KAPPA_ZERO the steady state damage
 // Also will need C_MIN and KAPPA_STAR which appear in the damage function phi.
 // These are all constant throughout the run and are properties.
 // This is the constructor for the BrittleElastic model.  The properties
 // it reads in are the BULK_MODULUS and SHEAR_MODULUS.  These
 // are stored in the properties array.
 //
 //*****************************************************************
BrittleElastic::BrittleElastic(const MatProps & props) :
   Material(props) {
  setFlag(USE_LEFT_STRETCH);
   setHyper();
   //
   // Material Property Definitions
   // What is read in from the input dec
   // YOUNGS_MODULUS, POISSONS_RATIO,  XI_CHEMICAL, K_KINETIC, KAPPA_ZERO, C_MIN, KAPPA_STAR, ↩
       INSTANT  - 8 total properties

   mat_name = "BRITTLEELASTIC";
   num_material_properties = 8;
   initializeProperties();

   setMaterialProperty(0,"YOUNGS_MODULUS",props);
   setMaterialProperty(1,"POISSONS_RATIO",props);
   // The default behavior is linear elastic without damage
   setMaterialPropertyDefault(2,0.0);
   setMaterialProperty(2,"XI_CHEMICAL",props);
   setMaterialPropertyDefault(3,0.0);
   setMaterialProperty(3,"K_KINETIC",props);
   setMaterialPropertyDefault(4,0.0);
   setMaterialProperty(4,"KAPPA_ZERO",props);
   setMaterialPropertyDefault(5,0.0);
   setMaterialProperty(5,"C_MIN",props);
   setMaterialPropertyDefault(6,1.0);
   setMaterialProperty(6,"KAPPA_STAR",props);
   setMaterialPropertyDefault(7,0);
   setMaterialProperty(7,"INSTANTANEOUS",props);

```

```
54    LAME_FORTRAN(elastic_property_check)( num_material_properties,
55                 &properties[0] );
56
57    //
58    // State Variable Definitions
59    // We are using 5 state variables. The bulk and shear are two. then damage kappa. then two ↩
          energies
60    //
61
62    num_state_vars = 5;
63
64    set_state_variable_alias("YOUNGS_MODULUS", 0);
65    set_state_variable_alias("POISSONS_RATIO", 1);
66    set_state_variable_alias("KAPPA_INTERNAL", 2);
67    set_state_variable_alias("MECHANICAL_ENERGY", 3);
68    set_state_variable_alias("CHEMICAL_ENERGY", 4);
69
70  }
71
72  int BrittleElastic::initialize( matParams * p ) {
73    const double ym = properties[0];
74    const double nu = properties[1];
75
76   originalYoungs = properties[0];
77   originalNu = properties[1];
78
79        xi = properties[2];
80        bigk = properties[3];
81   kzero = properties[4];
82   cmin = properties[5];
83   kstar = properties[6];
84   instant = properties[7];
85
86    const double kappa = properties[4]; //initially we are neutral/steadystate damaged
87    double * stateOld = p->state_old;
88    double * stateNew = p->state_new;
89    for (int i(0); i < p->nelements; ++i) {
90      //
91      //  Set the default state to the block elastic constants
92      //
93      stateOld[0] = stateNew[0] = ym;
94      stateOld[1] = stateNew[1] = nu;
95      stateOld[2] = stateNew[2] = kappa;
96      stateOld[3] = stateNew[3] = 0.0;
97      stateOld[4] = stateNew[4] = 0.0;
98      stateOld += 5;
99      stateNew += 5;//5 internal state variables
100   }
101
102   return 0;
103 }
104
105 //***************************************************************
106 //
107 // The getStress method returns the stress given the strain
```

```
108  // and the time step.
109  //
110  //*****************************************************************
111
112  int BrittleElastic::getStress( matParams * p ) {
113    const double dt = p->dt;
114  //  const double * strainRate = p->strain_rate;
115    double * rotation = p->rotation;
116    const double * leftStretch = p->left_stretch;
117    const double * stressOld = p-> stress_old;
118    const double * stateOld = p-> state_old;
119    double * stressNew = p-> stress_new;
120    double * stateNew = p-> state_new;
121
122   const double onehalf = 1.0/2.0;
123   const double onethird = 1.0/3.0;
124
125        double devlnV [6];
126        double evals [3];
127   double evecs [9];
128   double LQT [9];
129   double resultmat [9];
130   double tracelnV;
131
132    for (int i(0); i < p->nelements; ++i) {
133
134    const double ym = originalYoungs; //stateOld[0] < 0 ? 0 : stateOld[0];     // ym must be ↩
         positive
135         const double pr = originalNu; //stateOld[1] < -0.9999999 ? -0.9999999 :
136   const double bulk = ym/3.0/(1.0-2.0*pr);
137   const double shear = ym/2.0/(1.0+pr);
138      const double kappa = stateOld[2] > kstar ? kstar :
139            stateOld[2] < 0 ? 0 : stateOld[2]; //keep kappa less than kstar and greater than 0
140
141   const double J = leftStretch[xx]*(leftStretch[yy]*leftStretch[zz]-leftStretch[yz]*↩
       leftStretch[yz])
142            -leftStretch[xy]*(leftStretch[xy]*leftStretch[zz]-leftStretch[zx]*leftStretch[yz])
143            +leftStretch[zx]*(leftStretch[xy]*leftStretch[yz]-leftStretch[yy]*leftStretch[zx]);
144   const double lnJ = FastLog(J);
145   const double sheartwooverJ = shear*2.0/J;
146   const double lnJdivJ = lnJ/J;
147
148   devlnV[xx]=leftStretch[xx];
149   devlnV[yy]=leftStretch[yy];
150   devlnV[zz]=leftStretch[zz];
151   devlnV[xy]=leftStretch[xy];
152   devlnV[yz]=leftStretch[yz];
153   devlnV[zx]=leftStretch[zx];
154
155   LAME_FORTRAN(lame_eigen)(1,1,devlnV,evals,evecs);//calculate eigen values and vectors
156   //Then use these values to calculate logV in its eigenbasis.
157   //then convert basis back to where we are using: QλQ^T = V
158   evals[0]=FastLog(evals[0]);
159   evals[1]=FastLog(evals[1]);
160   evals[2]=FastLog(evals[2]);
```

```
161
162     //make Lambda*Q^T, name is LQT .
163     // this is taking the rows of Q^T and mult by lambda_i for each row
164     LQT[fxx] = evals[0]*evecs[fxx];//
165     LQT[fxy] = evals[0]*evecs[fyx];//fyx because it is Q^T, not Q
166     LQT[fxz] = evals[0]*evecs[fzx];
167
168     LQT[fyy] = evals[1]*evecs[fyy];
169     LQT[fyz] = evals[1]*evecs[fzy];
170     LQT[fyx] = evals[1]*evecs[fxy];
171
172     LQT[fzz] = evals[2]*evecs[fzz];
173     LQT[fzx] = evals[2]*evecs[fxz];
174     LQT[fzy] = evals[2]*evecs[fyz];
175
176     //now Q matrix is just the evecs matrix, so multiply
177     // lnV = evecs * LQT
178 //this lame call to fortran does not work, so I replaced it with a direct calculation instead
179 // LAME_FORTRAN(lame_tensor_product3636)(1,1,evecs,LQT,resultmat);//multiply evecs*LQT
180     //result mat should be symmetric since it is now ln V
181
182     resultmat[fxx] = evecs[fxx]*LQT[fxx]+evecs[fxy]*LQT[fyx]+evecs[fxz]*LQT[fzx];
183     resultmat[fxy] = evecs[fxx]*LQT[fxy]+evecs[fxy]*LQT[fyy]+evecs[fxz]*LQT[fzy];
184     resultmat[fxz] = evecs[fxx]*LQT[fxz]+evecs[fxy]*LQT[fyz]+evecs[fxz]*LQT[fzz];
185
186     resultmat[fyx] = evecs[fyx]*LQT[fxx]+evecs[fyy]*LQT[fyx]+evecs[fyz]*LQT[fzx];
187     resultmat[fyy] = evecs[fyx]*LQT[fxy]+evecs[fyy]*LQT[fyy]+evecs[fyz]*LQT[fzy];
188     resultmat[fyz] = evecs[fyx]*LQT[fxz]+evecs[fyy]*LQT[fyz]+evecs[fyz]*LQT[fzz];
189
190     resultmat[fzx] = evecs[fzx]*LQT[fxx]+evecs[fzy]*LQT[fyx]+evecs[fzz]*LQT[fzx];
191     resultmat[fzy] = evecs[fzx]*LQT[fxy]+evecs[fzy]*LQT[fyy]+evecs[fzz]*LQT[fzy];
192     resultmat[fzz] = evecs[fzx]*LQT[fxz]+evecs[fzy]*LQT[fyz]+evecs[fzz]*LQT[fzz];
193
194     tracelnV = resultmat[fxx]+resultmat[fyy]+resultmat[fzz];
195
196     devlnV[xx]=resultmat[fxx]-onethird*tracelnV;
197     devlnV[yy]=resultmat[fyy]-onethird*tracelnV;
198     devlnV[zz]=resultmat[fzz]-onethird*tracelnV;
199     devlnV[xy]=resultmat[fxy];
200     devlnV[yz]=resultmat[fyz];
201     devlnV[zx]=resultmat[fzx];
202
203     //update the mechanical energy
204     stateNew[3]=bulk*(J*lnJ-J+1)+shear*(devlnV[xx]*devlnV[xx]+devlnV[yy]*devlnV[yy]+devlnV[zz]*↩
          devlnV[zz]+
205           2.0*devlnV[xy]*devlnV[xy]+2.0*devlnV[yz]*devlnV[yz]+2.0*devlnV[zx]*devlnV[zx]);
206
207     //update the new damage
208     const double expbxt = (instant == 1) ? 0 : exp(-bigk*xi*dt);
209     double kappaNew = kappa*expbxt-kzero*(expbxt-1.0)-(1.0-expbxt)*(cmin-1.0)/kstar*stateNew[3]/↩
          xi; // alphaNew has a mu in it
210        kappaNew = kappaNew > kstar ? kstar :
211              kappaNew < 0.0 ? 0.0 : kappaNew;
212
213     const double phi = 1.0 - (1.0 - cmin)*kappaNew/kstar;
```

```
214
215   //finally calculate the cauchy stress
216      stressNew[0] =  phi*bulk*lnJ + phi*sheartwooverJ*devlnV[xx]; //xx
217      stressNew[1] =  phi*bulk*lnJ + phi*sheartwooverJ*devlnV[yy]; //yy
218      stressNew[2] =  phi*bulk*lnJ + phi*sheartwooverJ*devlnV[zz]; //zz
219      stressNew[3] =                 phi*sheartwooverJ*devlnV[xy]; //xy
220      stressNew[4] =                 phi*sheartwooverJ*devlnV[yz]; //yz
221      stressNew[5] =                 phi*sheartwooverJ*devlnV[zx]; //zx
222
223   //rotate (or unrotate) back to reference R^T T R
224   //in Sierra T is called sigma
225   //use resultmat to temporarily store the info
226     unRotate(stressNew,rotation);
227
228      stateNew[0] = ym*phi;//original youngs modulus is reduced by phi
229      stateNew[1] = pr;//poissons ratio is not affected by damage
230      stateNew[2] = kappaNew;
231   stateNew[3]=stateNew[3]*phi;
232   stateNew[4] = onehalf * (kappaNew - kzero)*(kappaNew - kzero)*xi;
233   rotation +=9;
234      stressOld += 6;
235      stressNew += 6;
236   leftStretch +=6;
237      stateOld += 5;
238      stateNew +=5;
239    }
240    return 0;
241  }
242 //-----------------------------------------------------------------------
243  int  BrittleElastic::getInitialElasticModuli( matParams * p ) {
244    const double * stateOld = p-> state_old;
245    for (int i(0); i < p->nelements; ++i) {
246      p->pcVars_old[PC_YOUNGS_MODULUS][i] = stateOld[0]; // 9.0*stateOld[0]*stateOld[1]/(3.0*↩
           stateOld[0]+stateOld[1]);
247      p->pcVars_old[PC_POISSONS_RATIO][i] = stateOld[1]; //(3.0*stateOld[0]-2.0*stateOld[1])↩
           /(2.0*(3.0*stateOld[0]+stateOld[1]));
248    }
249    return 0;
250  }
251
252 int  BrittleElastic::pcElasticModuli( matParams * p )
253 {
254    const double * stateOld = p-> state_old;
255    double * stateNew = p-> state_new;
256   double * oldE = p->pcVars_old[PC_YOUNGS_MODULUS];
257   double * newE = p->pcVars_new[PC_YOUNGS_MODULUS];
258   double * oldNu = p->pcVars_old[PC_POISSONS_RATIO];
259   double * newNu = p->pcVars_new[PC_POISSONS_RATIO];
260
261    for (int i(0); i < p->nelements; ++i)
262   {
263   oldE[i] = stateOld[0];
264   newE[i] = stateNew[0];
265   oldNu[i] = stateOld[1];
266   newNu[i] = stateNew[1];
```

```
267       stateOld += 5;
268       stateNew +=5;
269     }
270     return 0;
271 }
272 void BrittleElastic::unRotate(double * stressNew, double * rotation)
273 {
274       double resultmat [6];
275         // R^T S R
276         resultmat[xx] = rotation[fxx]*(rotation[fxx]*stressNew[xx] + rotation[fyx]*stressNew[xy↩
                ] + rotation[fzx]*stressNew[zx]) + rotation[fyx]*(rotation[fxx]*stressNew[xy] + ↩
                rotation[fyx]*stressNew[yy] +
277 rotation[fzx]*stressNew[yz]) + rotation[fzx]*(rotation[fyx]*stressNew[yz] + rotation[fxx]*↩
      stressNew[zx] + rotation[fzx]*stressNew[zz]);
278         resultmat[yy] = rotation[fxy]*(rotation[fxy]*stressNew[xx] + rotation[fyy]*stressNew[xy↩
                ] + rotation[fzy]*stressNew[zx]) + rotation[fyy]*(rotation[fxy]*stressNew[xy] + ↩
                rotation[fyy]*stressNew[yy] +
279 rotation[fzy]*stressNew[yz]) + rotation[fzy]*(rotation[fyy]*stressNew[yz] + rotation[fxy]*↩
      stressNew[zx] + rotation[fzy]*stressNew[zz]);
280         resultmat[zz] = rotation[fxz]*(rotation[fxz]*stressNew[xx] + rotation[fyz]*stressNew[xy↩
                ] + rotation[fzz]*stressNew[zx]) + rotation[fyz]*(rotation[fxz]*stressNew[xy] + ↩
                rotation[fyz]*stressNew[yy] +
281  rotation[fzz]*stressNew[yz]) + rotation[fzz]*(rotation[fyz]*stressNew[yz] + rotation[fxz]*↩
      stressNew[zx] + rotation[fzz]*stressNew[zz]);
282
283         resultmat[xy] = rotation[fxx]*(rotation[fxy]*stressNew[xx] + rotation[fyy]*stressNew[xy↩
                ] + rotation[fzy]*stressNew[zx]) + rotation[fyx]*(rotation[fxy]*stressNew[xy] + ↩
                rotation[fyy]*stressNew[yy] +
284  rotation[fzy]*stressNew[yz]) + rotation[fzx]*(rotation[fyy]*stressNew[yz] + rotation[fxy]*↩
      stressNew[zx] + rotation[fzy]*stressNew[zz]);
285         resultmat[yz] = rotation[fxy]*(rotation[fxz]*stressNew[xx] + rotation[fyz]*stressNew[xy↩
                ] + rotation[fzz]*stressNew[zx]) + rotation[fyy]*(rotation[fxz]*stressNew[xy] + ↩
                rotation[fyz]*stressNew[yy] +
286  rotation[fzz]*stressNew[yz]) + rotation[fzy]*(rotation[fyz]*stressNew[yz] + rotation[fxz]*↩
      stressNew[zx] + rotation[fzz]*stressNew[zz]);
287         resultmat[zx] = rotation[fxz]*(rotation[fxx]*stressNew[xx] + rotation[fyx]*stressNew[xy↩
                ] + rotation[fzx]*stressNew[zx]) + rotation[fyz]*(rotation[fxx]*stressNew[xy] + ↩
                rotation[fyx]*stressNew[yy] +
288  rotation[fzx]*stressNew[yz]) + rotation[fzz]*(rotation[fyx]*stressNew[yz] + rotation[fxx]*↩
      stressNew[zx] + rotation[fzx]*stressNew[zz]);
289
290         for(unsigned int j=0; j<6;++j)
291                 stressNew[j]=resultmat[j];
292 }
293 } // lame
```

INTENTIONALLY LEFT BLANK.

1 (PDF) DEFENSE TECHNICAL INFORMATION CTR DTIC OCA

2 (PDF) DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL IMAL HRA MAIL & RECORDS MGMT

1 (PDF) GOVT PRINTG OFC A MALHOTRA

4 (PDF) DIR USARL RDRL DPW R COATES P FROUNFELKER R SPINK M TEGTMEYER

1 (PDF) DIR USARL RDRL ROP L F GREGORY

2 (PDF) DIR USARL RDRL CIH C B HENZ M VINDIOLA

10 (PDF) DIR USARL RDRL HR P FRANASZCZUK RDRL HRS C S GORDON W HAIRSTON B LANCE K MCDOWELL J MCARDLE K OIE A PASSARO M PETERSON J VETTEL

7 (PDF) DIR USARL RDRL SLB W D BOOTHE A BREUER N EBERIUS P GILLICH C KENNEDY A KULAGA W MERMAGEN

44 (PDF) DIR USARL RDRL WM S KARNA RDRL WML C T PIEHLER RDRL WML H B SCHUSTER RDRL WMM B B LOVE RDRL WMM G L PIEHLER N ZANDER RDRL WMP S SCHOENFELD RDRL WMP B A DAGRO A DILEONARDI A DWIVEDI W EVANS C GUNNARSSON C HOPPEL Y HUANG M LYNCH J MCDONALD P MCKEE B SANBORN S SATAPATHY A SOKOLOW C WEAVER T WEERASOORIYA S WOZNIAK T ZHANG K ZIEGLER RDRL WMP C S BILYK T BJERKE D CASEM J CLAYTON D DANDEKAR M GREENFIELD B LEAVY M RAFTENBERG RDRL WMP D R DONEY J RUNYEON RDRL WMP E P SWOBODA RDRL WMP F E FIORAVANTE A FRYDMAN N GNIAZDOWSKI R GUPTA R KARGUS RDRL WMP G

R BANTON
N ELDREDGE
S KUKUCK