ARL-TR-9776 • Sep 2023

DEVCOM ARMY RESEARCH LABORATORY

Numerical Models for Constitutive Behaviors of Ferrous Metals

by John D Clayton and Jeffrey T Lloyd

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.

ARL-TR-9776 ● SEP 2023

Numerical Models for Constitutive Behaviors of Ferrous Metals

John D Clayton and Jeffrey T Lloyd
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
September 2023	Technical Report	October 2021–September 2023

4. TITLE AND SUBTITLE
Numerical Models for Constitutive Behaviors of Ferrous Metals

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
John D Clayton and Jeffrey T Lloyd

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLA-TB
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-9776

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
primary author's email: <john.d.clayton1.civ@army.mil>
ORCID ID: John D Clayton 0000-0003-4107-6282

14. ABSTRACT
A mechanical user material subroutine (UMAT) developed for ferrous metals is documented. The UMAT includes a number of subroutines for different constitutive behaviors characteristic of ferrous materials. The present focus is on constitutive models used in recent research on dynamic responses of phase-transforming steels and pure iron (Fe). Models can account for finite deformation, nonlinear thermoelasticity, plasticity from dislocations and twinning, damage from voids, and solid-solid phase transitions driven by pressure, shear stress, temperature, and magnetic fields. Two different kinds of equation of state are encompassed. General yield and flow functions of history and local state capture the net deviatoric stress-strain behavior. Two different kinetic functions for phase transitions are included. A generalized void growth law with degradation of tangent elastic coefficients and material strength is given. A separate, coupled magnetic user material subroutine (MAGUMAT) is implemented to update the magnetization and magnetic field-dependent properties. A material point simulator (MPS) software that calls both the UMAT and MAGUMAT is described. Verification examples invoking the MPS are reported.

15. SUBJECT TERMS
ferrous metals, equation of state, plasticity, phase transitions, damage, magnetohydrodynamics, user material, Terminal Effects

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			John D Clayton
Unclassified	Unclassified	Unclassified	UU	100	19b. TELEPHONE NUMBER (Include area code) 410-278-6146

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures v

List of Tables v

Acknowledgments vi

1. Introduction 1

2. Descriptions of the Models 5

2.1 CMF1 8

2.1.1 EOS 8

2.1.2 Plastic Flow 9

2.1.3 Phase Transformations 12

2.1.4 Damage Mechanics 15

2.1.5 Magnetization 15

2.1.6 Stress and Stress Rate 18

2.1.7 Temperature 19

2.2 CMF2 20

2.2.1 EOS 21

2.2.2 Plastic Flow 22

2.2.3 Phase Transformations 22

2.2.4 Damage Mechanics 24

2.2.5 Stress and Stress Rate 24

2.2.6 Temperature 25

3. Input Constants 26

3.1 CMF1 26

3.2 CMF2 31

3.3 Syntax for Input Files 36

4. State Variables 37

5. Code Maturity **40**

5.1 CMF1 40

5.2 CMF2 41

6. Standalone Simulator (MPS) **42**

6.1 Procedures 43

6.2 Syntax for Input File with Loading Conditions 44

6.3 Output Files 48

7. Verification Examples **49**

7.1 CMF1 49

7.2 CMF2 52

8. Conclusion **55**

9. References **56**

Appendix A. Input Files for Verification Examples **62**

Appendix B. MPS Source Code (`mps.f`) **69**

List of Symbols, Abbreviations, and Acronyms **90**

Distribution List **92**

List of Figures

Fig. 1 Hydrostatic compression and decompression of pure Fe, with model results obtained from CMF1. Top image shows results of ϵ phase ξ vs. pressure p for null applied magnetic field compared to experimental DAC data. Bottom left image shows predicted ξ for field strengths of 0, 25, and 50 T for loading-unloading cycles to 25.5 GPa; bottom right image shows density ratio $J = \rho_0/\rho$ vs. p for the same cycles, with hysteresis enclosed by the curves. 50

Fig. 2 Tension of a ferrous alloy, with model results obtained from CMF1. Respective top left and right images show engineering stress without and with applied field of strength 0.9 T. Bottom image shows volume fraction of α phase including experimental data from EBSD. 51

Fig. 3 Adiabatic uniaxial-strain and isothermal hydrostatic compression of a TRIP steel, with model results obtained from CMF2. Top image compares predictions with shock compression data on three stainless steels. Respective bottom left and right images show temperature and shear stress. 53

Fig. 4 Uniaxial-stress compression of a TRIP steel, with model results obtained from CMF2. Respective top left and right images show quasi-static and dynamic compression. Results of simulations and experiments obtained at room temperature and elevated temperature. Bottom image shows volume fraction of α phase including experimental data from EBSD. 54

List of Tables

Table 1 Input constants 1–32 for CMF1 27

Table 2 Input constants 33–64 for CMF1 28

Table 3 Input constants 65–96 for CMF1 29

Table 4 Input constants 1–32 for CMF2 32

Table 5 Input constants 33–64 for CMF2 33

Table 6 Input constants 65–96 for CMF2 34

Table 7 User-defined state variables for CMF1 and CMF2 38

Table 8 Successful prior research implementations of CMF1 and CMF2 40

Table 9 Primary input and output variables for UMAT and MAGUMAT subroutines 44

Table 10 Mechanical loading options and hardcoded strain ϵ and stresses P, τ for the MPS 47

Acknowledgments

The authors acknowledge support from the US Army Combat Capabilities Development Command Army Research Laboratory.

1. Introduction

Ferrous metals include iron (Fe), steels, and other Fe-based alloys. When subjected to large mechanical forces, these materials demonstrate a variety of deformation mechanisms depending on their chemistries and microstructures: nonlinear elasticity, dislocation plasticity, deformation twinning, solid-solid martensitic phase transitions, ductile fracture from voids, and brittle fracture from microcracks. Crystalline phases within the microstructure are usually body centered cubic (BCC), body centered tetragonal (BCT), face centered cubic (FCC), or hexagonal close packed (HCP). Under loading pertinent to dynamic impact, the extreme conditions of high pressures, high shear stresses, high strain rates, and high temperatures can be attained. Each of these conditions serves as a possible driving force for inelastic deformation mechanisms, albeit in a different way.

The predilection for dislocation slip versus deformation twinning versus phase transformation is often attributed to stacking fault energy (SFE),[1–3] with a higher SFE noted for those alloys that deform mostly by dislocation slip, referred to herein as slip-dominated plasticity (SLIP) steels. Transformation-induced plasticity (TRIP) steels characterized by ease of martensitic transformations tend to have a low SFE. Twinning-induced plasticity (TWIP) steels tend to have SFEs in between those of SLIP and TRIP steels.

Ferrous metals often demonstrate magnetic behavior. For example, α-Fe (BCC) is ferromagnetic, whereas ϵ-Fe (HCP) is weakly paramagnetic or antiferromagnetic.[4–7] Because the free energy of a magnetic material changes with the local magnetic field, an externally applied magnetic field will affect driving forces and kinetics for martensitic transformations between phases with different magnetic properties.[6,7] Magnetic fields can also affect fundamental thermodynamic properties of ferrous solids such as specific heat.[8,9]

Recent research by the coauthors[10–12] has developed novel constitutive theories and numerical implementations for ferrous metals. These continuum mechanical models capture the homogenized thermomechanical response of a polycrystalline material element consisting of up to two coexisting solid phases. Sufficient numbers of randomly oriented grains of each phase are assumed such that material isotropy is justified. Two classes of constitutive model framework (CMF) are documented in this report. Both frameworks satisfy established continuum balance laws and non-

negative dissipation restrictions.

The first model framework to be described, denoted as CMF1, was theoretically established in Clayton and Lloyd,[11] including analytical solutions for hydrostatic and shock compression of pure Fe, with and without external magnetic fields. A numerical implementation of CMF1 was achieved by Clayton et al.[12] Therein, numerical solutions were obtained for the response of pure Fe and a ferrous alloy with several heat treatments; calculations considered homogeneous magnetic fields and were restricted to a single material point (e.g., represented by an integration point within a single finite element [FE]).

The second framework to be described, denoted as CMF2, was theoretically established earlier in Clayton and Lloyd.[10] Numerical implementation in a single-point FE context was also achieved in that reference. Calculated homogeneous responses of a SLIP alloy, a TWIP alloy, and a TRIP alloy were obtained for different mechanical and thermal loading protocols. Multi-element simulations of a Taylor rod impact experiment on these same alloys were documented in a subsequent technical report.[3] The original theory and implementation of CMF2 did not explicitly consider magnetic effects.

Features of CMF1 as implemented numerically[12] are summarized as follows:

- Nonlinear pressure-volume-temperature equation of state (EOS) based on a logarithmic volumetric strain measure;
- Additive decomposition of spatial deformation rate tensor with plasticity, twinning, and deviatoric transformation strain rates combined;
- Composite flow rule with shear strength accounting for all inelastic deformation mechanisms in a physically justified manner;
- Relaxation equation for phase transition kinetics with metastable states driven by the Gibbs free energy difference that depends on pressure, shear, temperature, and magnetic field;
- Continuum damage mechanics with void-based kinetics and linear elastic moduli degradation;

- Objective stress rate (i.e., true Jaumann rate) consistent with logarithmic hyperelasticity derived under the assumptions of small deviatoric elastic stretch and stretch rate;
- Free energy, specific heat, and phase transition kinetic parameters dependent on magnetic field.

Features of CMF2 as implemented numerically[10] are summarized as follows:

- Nonlinear pressure-volume-temperature EOS based on an Eulerian volumetric strain measure;
- Additive decomposition of spatial deformation rate tensor with plasticity, twinning, and deviatoric transformation strain rates combined;
- Composite flow rule with shear strength accounting for all inelastic deformation mechanisms in a physically justified manner;
- Direct differential equation for phase transition kinetics driven by pressure, shear, and temperature;
- Continuum damage mechanics with void-based kinetics and self-consistent elastic moduli degradation;
- Objective stress rate (i.e., modified Jaumann rate) consistent with Eulerian hyperelasticity derived under the assumptions of small deviatoric elastic stretch and stretch rate.

Key equations and further supporting references for each CMF are given in Section 2 of this report. CMF1 is more recent and includes more physics, but it has been implemented to date only in material point simulations.[12] CMF2 is more mature, including implementation in a larger-scale host code for Taylor impact simulations,[3] but it does not consider effects of electromagnetism. A phase transition framework similar to that in CMF1 was exercised in multi-element plane-wave and impact simulations,[13] but that work did not use the EOS and strength models of CMF1 or CMF2. Neither CMF1 nor CMF2 explicitly considers non-Schmid effects[14,15] or residual lattice dilatation from dislocation core and nonlinear elastic

fields (e.g, anharmonicity),[15–17] either or both of which could be important for BCC phases of ferrous materials.

The purpose of the current report is to document numerical implementations of both frameworks in a user material subroutine (UMAT). This subroutine is intended to be called by the constitutive update algorithm of a host FE code[18–21] such as `ABAQUS`, `ALE3D`, `ALEGRA`, or `EPIC`. The mechanical constitutive subroutine follows the UMAT syntax of `ABAQUS`. It therefore should be compatible with other host codes that support the `ABAQUS` UMAT, including the other three host codes just mentioned.

For coupled electromagnetic behaviors, a separate, supplementary magnetic user material subroutine (MAGUMAT) is used to calculate the magnetization, magnetic field energy, and specific heat. The MAGUMAT is given as input a local magnetic flux density or a local magnetic field as the pointwise magnetic loading condition. This routine also includes the capability to calculate the dissipated energies from the local heat flux and electric current, both of which affect the temperature rate. Whether or not this algorithmic breakdown is supported by the host code depends on the structure of the host code, namely its magnetohydrodynamics solver. It may be necessary to restructure the mechanical UMAT to contain or internally call the MAGUMAT, rather than rely on the host code to make separate calls to the UMAT and MAGUMAT. This task is left to the user.

The UMAT and MAGUMAT subroutines do not prohibit user invocation of different aspects of each CMF. For example, the phase transition kinetics of CMF1 could be used in conjunction with the Eulerian EOS of CMF2. However, mixed combinations such as this have not undergone verification testing. The current report only documents CMF1 and CMF2 with suggested inputs for those ensembles of features that have been successfully used in previous works[3,10,12] or that are verified in this report. Some non-standard feature combinations may not be mathematically, physically, or thermodynamically consistent, so such combinations should only be attempted at the user's discretion, with further verification strongly recommended to be performed by the user.

This report also documents a material point simulator (MPS) that imposes macroscopically homogeneous deformation and magnetic states, noting that states among individual phases in the mixed-phase domain may still differ within the material

point. The MPS calls the UMAT and MAGUMAT separately and sequentially, and the input syntax for material properties of the MPS follows the `ABAQUS` style. Verification problems are solved in this report using the MPS with each of CMF1 and CMF2. All three routines, namely the UMAT, MAGUMAT, and MPS, are written in the FORTRAN programming language. The source code for the MPS is included in an appendix. For brevity, source code for the UMAT and MAGUMAT are not included in this report. The authors may supply these files to interested parties upon future request, to be considered on a case-by-case basis. Along these lines, any errors discovered in the code or in the present documentation should be reported to the authors for subsequent correction.

The layout of this report is as follows. Contents include those recommended for standardized UMAT documentation by Gerlach et al.[22] with the exception of exclusion of the UMAT source code. Section 2 includes descriptions of the theoretical features and key governing equations for each CMF. Section 3 describes the input constants and input syntax for each CMF. Section 4 describes the state variables for each CMF. Section 5 discusses code maturity and robustness. Section 6 documents the MPS. Section 7 contains two classes of verification example problems for each CMF. Section 8 is the conclusion. Following the references, Appendix A contains the input files for the verification problems. Appendix B contains the FORTRAN source code for the MPS. A list of acronyms and mathematical notation follows the Appendices.

2. Descriptions of the Models

All models are consistent with the fundamental laws of continuum physics.[23–26] These include conservation of mass, conservation of linear and angular momentum, conservation of energy, the Clausius-Duhem inequality, and Maxwell's equations in the Galilean invariant approximation, all of which are summarized hereafter.

The models allow for coexistence of up to two solid phases at any material point with reference coordinates $\boldsymbol{X}$. At time t, the local volume fraction of the product (i.e., transformed) phase is $\xi^{(1)} = \xi$, and the local volume fraction of the parent phase is $\xi^{(0)} = 1 - \xi$. The initial value of $\xi(\boldsymbol{X}, t)$ is $\xi_0 = \xi(\boldsymbol{X}, 0)$. The initial mass density of the parent phase is $\rho_0^{(0)}$, and the initial mass density of the product phase is $\rho_0^{(1)}$. The volume change associated with complete forward transformation $0 \to 1$ is δ^{ξ}. The damage variable is $\phi(\boldsymbol{X}, t) \in [0, 1]$ with initial condition $\phi_0 = \phi(\boldsymbol{X}, 0) =$

0. Dilatation from damage is modulated by a material parameter $c^\phi \in [0,1]$. The product $c^\phi \phi$ is physically interpreted as the local void volume fraction.

Local mass conservation requires[10,12]

$$\rho_0 = \rho J; \qquad \rho_0 = (1-\xi_0)\rho_0^{(0)} + \xi_0 \rho_0^{(1)}; \tag{1}$$

$$J = \det \boldsymbol{F} = J^E J^\xi J^\phi, \qquad \dot{J} = J \nabla \cdot \boldsymbol{v}; \tag{2}$$

$$\frac{\dot{J^\xi}}{J^\xi} = \frac{\delta^\xi}{1+(\xi-\xi_0)\delta^\xi}\dot{\xi}, \qquad \delta^\xi = \frac{\rho_0^{(0)}}{\rho_0^{(1)}} - 1; \tag{3}$$

$$\frac{\dot{J^\phi}}{J^\phi} = \frac{c^\phi \dot{\phi}}{1-c^\phi \phi}. \tag{4}$$

The Jacobian determinant of the deformation gradient $\boldsymbol{F}(\boldsymbol{X},t)$ is J. The initial and current mass densities of the mixture of phases are ρ_0 and ρ. The thermoelastic volume change is measured by J^E. The scalars J^ξ and J^ϕ describe, if different from unity, the respective volume changes from phase transitions and induced voids. The particle velocity vector is $\boldsymbol{v}(\boldsymbol{X},t)$. Superposed dots are material time derivatives.

Denote the spatial magnetic flux density by $\boldsymbol{B}$ and the magnetization vector per unit spatial volume by $\boldsymbol{M}$. Local linear and angular momentum conservation require[11,12]

$$\nabla \cdot \boldsymbol{t} + \tilde{\boldsymbol{f}} + \boldsymbol{f} = \rho \dot{\boldsymbol{v}}, \qquad \boldsymbol{t} - \boldsymbol{t}^{\mathrm{T}} = \boldsymbol{B}\otimes\boldsymbol{M} - \boldsymbol{M}\otimes\boldsymbol{B}. \tag{5}$$

The Cauchy stress tensor is $\boldsymbol{t}$, the mechanical body force vector per unit spatial volume is $\boldsymbol{f}$, and the electromechanical body force per unit spatial volume is $\tilde{\boldsymbol{f}}$:

$$\tilde{\boldsymbol{f}} = \boldsymbol{J}\times\boldsymbol{B} + (\nabla\boldsymbol{B})\cdot\boldsymbol{M}, \tag{6}$$

with $\boldsymbol{J}$ the electric current density. The symmetric part of the Cauchy stress tensor is

$$\boldsymbol{\sigma} = \boldsymbol{t} - \tfrac{1}{2}(\boldsymbol{B}\otimes\boldsymbol{M} - \boldsymbol{M}\otimes\boldsymbol{B}). \tag{7}$$

For the isotropic constitutive models implemented in this report, Cauchy stress is always symmetric, meaning $\boldsymbol{t} = \boldsymbol{\sigma}$, but this simplification does not hold in the general theory for materials of arbitrary symmetry.[11,23] Also for the present constitutive models, the Cauchy stress $\boldsymbol{t}$ and Cauchy pressure p are decomposed into a symmetric lattice part $(\cdot)^E$ induced by thermoelastic deformation and an electromagnetic

contribution:

$$\boldsymbol{t} = \boldsymbol{t}^E + \boldsymbol{B} \otimes \boldsymbol{M}, \quad p = -\tfrac{1}{3}\mathrm{tr}\,\boldsymbol{t} = -\tfrac{1}{3}\mathrm{tr}\,\boldsymbol{\sigma} = -\tfrac{1}{3}(\mathrm{tr}\,\boldsymbol{t}^E + \boldsymbol{B}\cdot\boldsymbol{M}) = p^E - \tfrac{1}{3}\boldsymbol{B}\cdot\boldsymbol{M}. \quad (8)$$

Denote by U the internal energy of the mixture per unit reference volume, $\boldsymbol{q}$ the spatial heat flux vector, and r the local heat source per unit spatial volume. Denote the electric current $\mathsf{J} = \boldsymbol{J} - \boldsymbol{v}\nabla\cdot\boldsymbol{D}$ and electric field $\mathsf{E} = \boldsymbol{E} + \boldsymbol{v}\times\boldsymbol{B}$ vectors entering the Galilean-invariant formulation for non-polarizable conductors.[11,23] Then the local balance of energy is[11,12]

$$J^{-1}\dot{U} = \boldsymbol{t} : \nabla\boldsymbol{v} + \mathsf{J}\cdot\mathsf{E} - \boldsymbol{M}\cdot\dot{\boldsymbol{B}} - \nabla\cdot\boldsymbol{q} + r, \qquad (\boldsymbol{t} : \nabla\boldsymbol{v} = t_{ji}\partial_j v_i). \quad (9)$$

Isotropic models of constant thermal conductivity $\kappa \geq 0$ and constant electrical conductivity $\Sigma \geq 0$ are implemented in some algorithms discussed later, whereby

$$\boldsymbol{q} = -\kappa\nabla\theta, \qquad \mathsf{J} = \Sigma\mathsf{E}. \quad (10)$$

The local absolute temperature is θ, the local Helmholtz free energy density is ψ, and the local entropy density per unit reference volume is η:

$$\psi = U - \theta\eta. \quad (11)$$

The local Clausius-Duhem inequality is

$$\dot{\eta} \geq (J/\theta)(r - \nabla\cdot\boldsymbol{q} + \{\boldsymbol{q}/\theta\}\cdot\nabla\theta), \quad (12)$$

whereby from Eqs. 9 and 11,

$$-(1/J)(\dot{\psi} + \eta\dot{\theta}) + \boldsymbol{t} : \nabla\boldsymbol{v} + \mathsf{J}\cdot\mathsf{E} - \boldsymbol{M}\cdot\dot{\boldsymbol{B}} - \{\boldsymbol{q}/\theta\}\cdot\nabla\theta \geq 0. \quad (13)$$

For electromagnetics, the present implementation uses rationalized MKS units.[24] Vacuum permittivity is $\epsilon_0 = 8.854\cdot 10^{-12}$F/m, vacuum permeability is $\mu_0 = 4\pi\cdot 10^{-7}$H/m, and light speed is $\mathsf{c} = 3\cdot 10^8$ m/s, related by $\epsilon_0\mu_0\mathsf{c}^2 = 1$.

Let $\boldsymbol{D}$ be the electric displacement vector and $\boldsymbol{H}$ the spatial magnetic field. For non-

polarizable materials,

$$\boldsymbol{D} = \epsilon_0 \boldsymbol{E}, \qquad \boldsymbol{B} = \mu_0(\boldsymbol{H} + \boldsymbol{M}). \tag{14}$$

Quasi-magnetostatic conditions[23] are assumed, with free charges absent. Maxwell's equations at any spatial point $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{X}, t)$, expressed with respect to a fixed spatial frame, are

$$\nabla \cdot \boldsymbol{E} = 0, \qquad \nabla \cdot \boldsymbol{B} = 0, \qquad \nabla \times \boldsymbol{E} = -\partial_t \boldsymbol{B}, \qquad \nabla \times \boldsymbol{H} = \boldsymbol{J}. \tag{15}$$

Time differentiation at fixed $\boldsymbol{x}$ is $\partial_t(\cdot)$. Alternative forms of Maxwell's equations can be derived in the Galilean-invariant approximation, which presupposes that particle velocity is small compared to the speed of light in a vacuum. The reader is referred to the monograph of Maugin[23] and the more succinct presentation in Clayton and Lloyd[11] for the pertinent derivations as well as boundary or jump conditions on surfaces, stationary or moving. Omitted in the present context are higher-order balance laws pertinent to spin-exchange effects that have been modeled elsewhere using more elaborate, generalized continuum theories.[27,28] To obtain numerical solutions, Maxwell's equations in the quasi-static approximation are typically manipulated to arrive at a magnetic diffusion equation.[11,29]

2.1 CMF1

Essential aspects of the formulation implemented numerically by Clayton et al.[12] are reported here. This formulation (i.e., CMF1) contains several alterations of the original, more rigorous theory of Clayton and Lloyd[11] to enable efficient numerical implementation in hydrocodes. These differences are discussed at length in a follow-up paper.[12]

2.1.1 EOS

A logarithmic EOS relates lattice pressure p^E to thermoelastic volume change J^E and temperature θ. This EOS can be derived from third-order nonlinear elasticity theory with a dependence of free energy on the material logarithmic strain tensor[25,30]:

$$p^E = -\frac{1}{J}\left[\mathsf{B} \ln J^E - \tfrac{1}{2}\mathsf{B}_0(\mathsf{B}_0' - 2)(\ln J^E)^2 - c_V \gamma_0 \Delta\theta\right]. \tag{16}$$

The initial bulk modulus, pressure derivative of the bulk modulus, and Grüneisen parameter are respective constants B_0, B_0', and γ_0. The specific heat per unit refer-

ence volume at constant volume is a mixture of the specific heats of phases:

$$c_V(\xi, \theta, |\boldsymbol{H}|) = (1-\xi)c_V^{(0)}(\theta, |\boldsymbol{H}|) + \xi c_V^{(1)}(\theta, |\boldsymbol{H}|). \tag{17}$$

The tangent bulk modulus B degrades linearly with damage for tensile states as follows, and the tangent shear modulus G entering later equations degrades linearly regardless of local deformation state:

$$\mathsf{B} = \mathsf{B}_0\{1 - \phi \mathrm{H}(\ln J^E)\}, \qquad \mathsf{G} = \mathsf{G}_0\{1-\phi\}. \tag{18}$$

The right-continuous Heaviside step function is $\mathrm{H}(\cdot)$. Dependence of elastic coefficients on temperature (e.g., modeled in other works on dynamic crystal plasticity[31–35] or macroscopic plasticity[36]) is omitted.

2.1.2 Plastic Flow

The Eulerian velocity gradient is decomposed into an elastic part $\boldsymbol{l}^E$ and a total plastic part $\boldsymbol{d}^P$. For isotropic polycrystals, the latter is presumed symmetric and accounts for plastic slip, deformation twinning, and the deviatoric contributions of phase transitions in its isochoric (i.e., traceless) part $\bar{\boldsymbol{d}}^P$:

$$\nabla \boldsymbol{v} = \boldsymbol{l}^E + \boldsymbol{d}^P; \qquad \boldsymbol{d}^P = \tfrac{1}{3}(\dot{J}^\xi/J^\xi + \dot{J}^\phi/J^\phi)\mathbf{1} + \bar{\boldsymbol{d}}^P, \quad \mathrm{tr}\,\bar{\boldsymbol{d}}^P = 0. \tag{19}$$

Associated deviatoric plasticity with a flow potential Ω^P and von Mises-equivalent yield and flow surface K^P is invoked:

$$\Omega^P = \sqrt{\tfrac{2}{3}} J(\bar{\sigma}^V - K^P), \qquad K^P = K^P(e^P, \dot{e}^P, \xi, \phi, \theta, \boldsymbol{H}). \tag{20}$$

The local von Mises stress is $\bar{\sigma}^V$, and the total deviatoric plastic strain rate is $\dot{e}^P$:

$$\bar{\sigma}^V = \sqrt{\tfrac{3}{2}}|(\boldsymbol{t} + p\mathbf{1}) : (\boldsymbol{t} + p\mathbf{1})|^{1/2}, \qquad \dot{e}^P = \sqrt{\tfrac{2}{3}}|\bar{\boldsymbol{d}}^P : \bar{\boldsymbol{d}}^P|^{1/2}. \tag{21}$$

The total plastic strain rate $\dot{e}^P$ is work conjugate to $\bar{\sigma}^V$. The cumulative scalar plastic strain is $e^P(\boldsymbol{X}, t) = \int \dot{e}^P(\boldsymbol{X}, \tau)\mathrm{d}\tau$.

The composite yield stress K^P, first introduced by the coauthors in prior work,[10] is used to modulate the total inelastic strain rate $\bar{\boldsymbol{d}}^P$ in the two-phase material, similarly to Grujicic and Sankaran.[37] Total strength K^P depends on the total flow resistance of each phase, phase interactions, and damage incurred in the solid. Mecha-

nistic resistances depend on strain history, strain rate, and temperature. Dependence of K^P on local magnetic field $\boldsymbol{H}$, which itself can be expressed as a function of other state variables, is also admitted in the theory, though this feature has yet to be implemented in the UMAT.

Let superscripts $(\alpha) = (0), (1)$ distinguish the two phases, recalling forward transformation is $\alpha : 0 \to 1$. Let K_0^P denote the total composite flow resistance in the absence of damage. A rule of mixtures is

$$K^P = K_0^P \cdot (1 - c^\phi \phi) = \{\xi K^{P(1)} + (1 - \xi) K^{P(0)}\} \cdot (1 - c^\phi \phi). \tag{22}$$

The degradation function $1 - c^\phi \phi$ is justified in prior research[10,38,39] by equivalent plastic work arguments.

For each phase α, slip and twinning resistances are embedded in strength function $K^{\chi(\alpha)}$, whereas transition flow resistance is embedded in $K^{\xi(\alpha)}$. Since slip, twinning, and phase transformations may operate in parallel,[37]

$$1/K^{P(\alpha)} = 1/K^{\chi(\alpha)} + 1/K^{\xi(\alpha)}. \tag{23}$$

A smooth double-well function of ξ is used to modulate $K^{\xi(\alpha)}$, with zeroes at $\xi - \xi_0 = 0, 1$:

$$z^{\xi(\alpha)}(\xi; \theta, \boldsymbol{H}) = 4\iota^{(\alpha)}(\theta, \boldsymbol{H}) \cdot (\xi - \xi_0)(1 - \{\xi - \xi_0\}); \tag{24}$$

$$K^{\xi(\alpha)} = K^{\chi(\alpha)} \left(1/z^{\xi(\alpha)} - 1\right) \quad \Rightarrow \quad K^{P(\alpha)} = K^{\chi(\alpha)} \left(1 - z^{\xi(\alpha)}\right). \tag{25}$$

The scalar function $\iota^{(\alpha)} \in (0, 1]$ when transformation strain accommodation induces softening in phase α. This permits prediction of sigmoidal stress-strain curves witnessed in some TRIP steels.[40,41]

Values of $\iota^{(0,1)}$, which affect composite strength only for $\xi \neq \xi_0$ and hence depict phase interactions, can most generally depend on the local temperature and magnetic field.[12] However, the present software implementation omits temperature dependence, sets ι the same for each phase, and presumes linear dependence on the local field magnitude:

$$\iota(|\boldsymbol{H}|) = \iota^{(0)}(|\boldsymbol{H}|) = \iota^{(1)}(|\boldsymbol{H}|) = \iota_0 + \iota_1 \mu_0 |\boldsymbol{H}|, \tag{26}$$

where ι_0 and ι_1 are constants. The second well in Eq. 24 is never attained in practice if $\xi_0 > 0$; in that case, some accommodation is permanent when $\xi \to 1$. The form in Eq. 24 is intended for situations when $\xi \geq \xi_0$ as in Clayton and Lloyd[10]; conditional modifications are necessary to account for scenarios in which reverse transformation renders $\xi < \xi_0$.

Now consider $K^{\chi(\alpha)}$, the resistance to slip and deformation twinning not comprised by phase transitions. Let $\dot{e}_0$ be a reference strain rate, and $m^{(\alpha)}, p^{(\alpha)}$ be material constants for each phase $\alpha = 0, 1$. Denote by $k^{(\alpha)}$ the slip-twinning resistance at an applied total deviatoric plastic strain rate $\dot{e}^P = \dot{e}_0$ and datum temperature $\theta = \theta_R$. Let $\bar{\sigma}^{V(\alpha)}$ be the von Mises stress in phase α. When plastic flow is occurring, a local viscoplastic flow rule is operative, where $\dot{e}^P$ is assumed the same in each phase:

$$\dot{e}^P = \dot{e}_0 \cdot \langle 1 - \bar{\theta}^{p^{(\alpha)}} \rangle^{-1/m^{(\alpha)}} \left[\bar{\sigma}^{V(\alpha)} / k^{(\alpha)} \right]^{1/m^{(\alpha)}}, \qquad \bar{\theta} = \frac{\langle \theta - \theta_R \rangle}{\theta_M - \theta_R}. \tag{27}$$

Thermal softening is measured by $p^{(\alpha)} \geq 0$. The melt temperature at ambient pressure is θ_M, and θ_R is treated as a universal constant.* Angled brackets denote the operation that parses positive values: $\langle x \rangle = \frac{1}{2}(x + |x|)$. Inversion of Eq. 27 gives

$$K^{\chi(\alpha)} = k^{(\alpha)} \cdot \left(\dot{e}^P / \dot{e}_0 \right)^{m^{(\alpha)}} \langle 1 - \bar{\theta}^{p^{(\alpha)}} \rangle = \bar{\sigma}^{V(\alpha)} \tag{28}$$

at local yield. The strain rate sensitivity for plastic flow in phase α is $m^{(\alpha)} = \partial \ln \bar{\sigma}^{V(\alpha)} / \partial \ln \dot{e}^P$. The von Mises-equivalent strength function $k^{(\alpha)}$ depends on the cumulative work-conjugate inelastic strain e^P as[2]

$$\begin{aligned} k^{(\alpha)}(e^P) = \sigma_0^{(\alpha)} + \frac{\Theta_0^{(\alpha)}}{2} \left[e^P - \frac{1}{\delta^{(\alpha)}} \ln\{\cosh[\delta^{(\alpha)}(e^P - \epsilon^{(\alpha)})]\} \right] \\ + \frac{\Theta_0^{(\alpha)}}{2\delta^{(\alpha)}} \ln\{\cosh(-\delta^{(\alpha)}\epsilon^{(\alpha)})\}. \end{aligned} \tag{29}$$

The datum-state yield stress for each phase is $\sigma_0^{(\alpha)}$. The strain hardening coefficient with dimensions of stress is $\Theta_0^{(\alpha)}$, and $\delta^{(\alpha)}$ and $\epsilon^{(\alpha)}$ dictate the hardening profile[2]:

$$\frac{\mathrm{d}k^{(\alpha)}}{\mathrm{d}e^P} = \frac{\Theta_0^{(\alpha)}}{2} \left[1 - \tanh\{\delta^{(\alpha)}(e^P - \epsilon^{(\alpha)})\} \right]. \tag{30}$$

*In the present UMAT software, θ_R is hardcoded to 300 K; in prior work[10] it was 293 K.

The net dissipation per unit reference volume from deviatoric flow is

$$\mathfrak{D}^P = J\hat{\beta}\bar{\sigma}^V\{\dot{e}^P - \sqrt{\tfrac{1}{6}}\gamma^\xi(\bar{\sigma}^V, \dot{e}^P)\dot{\xi}\} \geq 0. \tag{31}$$

The Taylor-Quinney factor $\hat{\beta} \in [0, 1]$ is assumed constant and identical in each phase, and γ^ξ is the signed transformation shear.

An augmentation of Eq. 28 was introduced by Clayton and Lloyd[10] to allow for increased strength at very high rates of loading,[42–44] for example manifesting from increased viscous and phonon drag of dislocations. This augmentation is available in the present software implementation of CMF1:

$$K^{\chi(\alpha)} = k^{(\alpha)} \cdot \left(\dot{e}^P/\dot{e}_0\right)^{m^{(\alpha)}} \langle 1 - \bar{\theta}^{p^{(\alpha)}}\rangle[1 + k_s\langle \ln(\dot{e}^P/\dot{e}_s)\rangle]. \tag{32}$$

Material constants are k_s and $\dot{e}_s > 0$, where the latter is generally a large strain rate for metals (e.g., on the order of 10^4/s). When $k_s > 0$, increased strength occurs for $\dot{e}^P > \dot{e}_s$.

2.1.3 Phase Transformations

To ensure non-negative net dissipation from the transformation rate $\dot{\xi}$, the kinetic law for transitions is required to satisfy[11,12]

$$\mathfrak{D}^\xi = -\Delta^*\mathbb{G} \cdot \dot{\xi} \geq 0. \tag{33}$$

The total Gibbs free energy difference, specifically the local energy density of the potentially transformed phase (1) minus that of the initial phase (0), is $\Delta^*\mathbb{G} = \mathbb{G}^{(1)} - \mathbb{G}^{(0)}$. For forward transitions, $\dot{\xi} > 0$ necessitates that the Gibbs free energy must not increase: $-\Delta^*\mathbb{G} \geq 0 \Leftrightarrow \mathbb{G}^{(1)} \leq \mathbb{G}^{(0)}$. Analogously, the Gibbs free energy must not increase for reverse transitions as well, where signs are reversed.

The theory of Boettger and Wallace and Lloyd et al.[9,13] is extended in works by the coauthors[11,12] to allow $\Delta^*\mathbb{G}$ to depend on shear (deviatoric) stress, magnetic flux density, and other state variables in addition to just the classical pressure and temperature dependencies of $\mathbb{G}^{(\alpha)}$ used in prior theory.[9,13] The local metastable state value of ξ, written $\xi^m(\boldsymbol{X}, t)$, is obtained from $\mathrm{d}\xi = (1-\xi)\mathrm{d}\bar{\mathbb{F}}$, where the dimensionless forward transition driving force is $\bar{\mathbb{F}} = -\Delta^*\mathbb{G}/\beta^F$. The proportionality factor β^F for forward transformation may generally depend on the local state, as can the

activation energy barrier α^F, and $\alpha_0^F/\beta_0^F = \alpha^F/\beta^F$ is prescribed so integration produces a metastable volume fraction ξ^m:

$$\int_0^{\xi^m} \frac{\mathrm{d}\xi}{1-\xi} = \int_{\alpha_0^F/\beta_0^F}^{-\Delta^*\mathbb{G}/\beta^F} \mathrm{d}\bar{\mathbb{F}} \quad \Rightarrow \quad \xi^m(\boldsymbol{X},t) = 1 - \exp\left[\frac{\alpha^F + \Delta^*\mathbb{G}(\boldsymbol{X},t)}{\beta^F}\right]. \tag{34}$$

The trial $(\cdot)^t$ forward transformation rate is given by linear relaxation kinetics[9,45]:

$$-\Delta^*\mathbb{G} > \alpha^F \text{ and } \xi(\boldsymbol{X},t) < \xi^m(\boldsymbol{X},t) \quad \mapsto \quad \dot{\xi}^t = (\xi^m - \xi)/\tau^F > 0, \tag{35}$$

with $\tau^F > 0$ a scalar with dimensions of time.

In reverse transformations, $\dot{\xi}^t < 0$ and $\mathrm{d}\xi = -\xi \mathrm{d}\bar{\mathbb{F}}$, wherein $\bar{\mathbb{F}} = \Delta^*\mathbb{G}/\beta^R$. Let β^R and α^R be the proportionality constant and activation energy barrier for reverse transformation with $\alpha_0^R/\beta_0^R = \alpha^R/\beta^R$, and let $\tau^R > 0$ be the time scale for kinetics. Then

$$\int_{\xi^m}^{0} \frac{\mathrm{d}\xi}{\xi} = -\int_{\Delta^*\mathbb{G}/\beta^R}^{\alpha_0^R/\beta_0^R} \mathrm{d}\bar{\mathbb{F}} \quad \Rightarrow \quad \xi^m(\boldsymbol{X},t) = \exp\left[\frac{\alpha^R - \Delta^*\mathbb{G}(\boldsymbol{X},t)}{\beta^R}\right]; \tag{36}$$

$$\Delta^*\mathbb{G} > \alpha^R \text{ and } \xi(\boldsymbol{X},t) > \xi^m(\boldsymbol{X},t) \quad \mapsto \quad \dot{\xi}^t = (\xi^m - \xi)/\tau^R < 0. \tag{37}$$

When conditions contrary to Eqs. 35 and 37 hold, no transformation is possible:

$$\left.\begin{array}{ll} -\alpha^F \leq \Delta^*\mathbb{G} \leq \alpha^R & \text{and/or} \\ -\Delta^*\mathbb{G} < \alpha^F \text{ and } \xi > \xi^m & \text{or} \\ \Delta^*\mathbb{G} > \alpha^R \text{ and } \xi < \xi^m & \end{array}\right\} \mapsto \dot{\xi} = \dot{\xi}^t = 0. \tag{38}$$

Quantities $\alpha^F, \alpha^R, \beta^F, \beta^R, \tau^F$, and τ^R can all potentially depend on local material state in the theory. However, in the current software implementation, τ^F and τ^R are fixed constants, and the phase transition barriers $\alpha^F, \alpha^R, \beta^F$, and β^R are permitted to depend only on local magnetic field strength $|\boldsymbol{H}|$ in a linear manner:

$$\alpha^{F/R} = \alpha_0^{F/R} + \alpha_1\mu_0|\boldsymbol{H}|, \qquad \beta^{F/R} = \beta_0^{F/R} + \beta_1\mu_0|\boldsymbol{H}|. \tag{39}$$

Six material constants are $\alpha_0^{F/R}, \alpha_1, \beta_0^{F/R}$, and β_1. Note that the linear field dependence is assumed identical for forward and reverse metastable states. Generally, α^F and α^R are unrestricted in sign, but $-\alpha^F \leq \alpha^R$, $\beta^F > 0$, and $\beta^R > 0$ are physical restrictions.

The shear strain rate due to phase transitions should not exceed the total deviatoric plastic strain rate, the latter of which includes the sum of contributions from slip, twinning, and phase transitions. Mathematically, this requirement leads to $2\dot{\epsilon}^P \geq \gamma^\xi \dot{\xi}$ and ensures that Eq. 31 always holds. An additional constraint on $\dot{\xi}$ is imposed to enforce this condition. The transformation shear of Eq. 31 is defined as a function of local von Mises stress state $\bar{\sigma}^V$ as in earlier models of TRIP steels,[40,41] and it becomes zero if the total deviatoric plastic strain rate $\dot{\epsilon}^P = \sqrt{\frac{3}{2}}\dot{e}^P$ is zero:

$$\dot{\xi} = \begin{cases} \min(\dot{\xi}^t, 2\dot{\epsilon}^P/\gamma^\xi), \\ \dot{\xi}^t \qquad \text{if } \gamma^\xi = 0; \end{cases} \qquad \gamma^\xi = \begin{cases} \gamma_0^\xi \cdot \min(1, \bar{\sigma}^V/\sigma_0^{(0)}), \\ 0 \qquad \text{if } \dot{\epsilon}^P = 0; \end{cases} \qquad \gamma_0^\xi \geq 0. \quad (40)$$

The transformation shear strain for local stress exceeding the datum yield stress is the material constant γ_0^ξ. The datum yield stress of the parent phase is $\sigma_0^{(0)} \geq 0$, introduced explicitly in Eq. 29. Since $\dot{\epsilon}^P \geq 0$ and $\gamma^\xi \geq 0$ by definition, the first of Eq. 40 only restricts forward transformations (i.e., those for which $\dot{\xi}^t > 0$). If $\gamma_0^\xi = 0$ (i.e., null transformation shear[9,13]), this constraint vanishes. Inequality Eq. 33 is guaranteed by Eqs. 34–40.

The driving force $\Delta^*\mathbb{G}$ is divided into mechanical (pressure and shear), magnetostatic, and chemical (compositional and thermal) contributions:

$$\begin{aligned} \Delta^*\mathbb{G} &= \Delta^*\hat{\mathbb{G}} + \Delta^*\tilde{\mathbb{G}} + \Delta^*\bar{\mathbb{G}} && \text{(total)}, \\ \Delta^*\hat{\mathbb{G}} &= pJ\delta^\xi/[1 + (\xi - \xi_0)\delta^\xi] - \sqrt{\tfrac{1}{6}}J\bar{\sigma}^V\gamma^\xi && \text{(mechanical)}, \\ \Delta^*\tilde{\mathbb{G}} &= \Delta^*\Phi && \text{(magnetostatic)}, \\ \Delta^*\bar{\mathbb{G}} &= -\Delta^* A[\mathsf{B}_0 \ln J^E \Delta\theta] - \Delta^*\hat{c}_V[\theta \ln(\theta/\theta_0) - \Delta\theta] \\ &\qquad - (\lambda_T/\theta_T)(\theta - \theta_T) + \psi_0 && \text{(chemical)}. \end{aligned} \quad (41)$$

The difference in thermal expansion coefficients (transformed phase minus initial phase) is $\Delta^* A = A^{(1)} - A^{(0)}$, with $A^{(\alpha)}$ the volumetric coefficient of thermal expansion of phase α, treated as a constant. The magnetic field-independent part of specific heat is $\hat{c}_V$. The latent heat coefficient of transformation is λ_T, and θ_T is a material constant called the transformation temperature. Energy density ψ_0 is the free energy of the transformed phase relative to the free energy of the initial phase at the reference temperature θ_0, at null lattice deformation (i.e., at no mechanical or thermal loads), and at null applied magnetic field. Further details are explained by

Clayton et al.[11,12]

2.1.4 Damage Mechanics

Dissipation from damage is constrained to be non-negative, leading to the following inequality:

$$\mathfrak{D}^{\phi} = \hat{\Xi}\dot{\phi} = -\{Jp[c^{\phi}/(1-c^{\phi}\phi)] - \hat{\omega}\}\dot{\phi} \geq 0; \quad \hat{\omega} = -\partial\psi/\partial\phi \geq 0. \tag{42}$$

Thus, the rate of irreversible damage should be constrained as follows noting $J > 0$:

$$c^{\phi}p > 0 \mapsto \dot{\phi} = 0; \qquad c^{\phi}p \leq 0 \mapsto \dot{\phi} \geq 0. \tag{43}$$

Because $c^{\phi} \in [0,1]$, dilatational damage is possible only when pressure is tensile. When $c^{\phi} = 0$, any kinetic equation with $\dot{\phi} \geq 0$ is admissible. In Clayton and Lloyd,[11] a generalized Tuler-Butcher spall model[46–48] was prescribed that obeys Eq. 43. Here,[12] Cocks-Ashby void growth kinetics[38,39,49] is implemented for general loading protocols, with slight modifications as outlined by Clayton and Lloyd[10]:

$$\dot{\phi} = \begin{cases} \sqrt{\frac{3}{2}}\sinh\left[\frac{2(2\hat{m}-1)}{2\hat{m}+1}\tilde{\Sigma}\right]\left[\frac{1}{(1-\phi)^{\hat{m}}} - \{1-(\phi+c_0)\}\right]\dot{e}^{P} & (\text{if } \tilde{\Sigma} \geq 0), \\ 0 & (\text{if } \tilde{\Sigma} \leq 0 \text{ or } \bar{\sigma}^{V} = 0). \end{cases} \tag{44}$$

The initial condition is $\phi(t=0) = \phi_0 = 0$. Two constant parameters are $\hat{m} \geq \frac{1}{2}$ related to viscoplastic rate sensitivity and $c_0 > 0$ to enable damage growth without assignment of an initial pore fraction. In Eq. 44, damage can only increase under conditions in which the triaxiality $\tilde{\Sigma} = -p/\bar{\sigma}^{V} > 0$ (i.e., tension) and plastic flow rate $\dot{e}^{P} > 0$. It can be verified using Eqs. 43 and 44 that damage is irreversible and Eq. 42 is unconditionally obeyed.

2.1.5 Magnetization

In the present theory,[11,12] like that of Daniel et al.,[50] the magnetic field $\boldsymbol{H}$ is presumed identical among coexisting phases at space-time location $\boldsymbol{x}(\boldsymbol{X},t)$. In the homogenized description, coexisting phases at $\boldsymbol{X}$ experience the same motion $\boldsymbol{x}$ and the same deformation gradient $\boldsymbol{F} = \nabla_0 \boldsymbol{x}$.

The magnetization vector $\boldsymbol{M}$ is the local volume average of the local magnetization

in each phase (α). Thus,

$$\boldsymbol{B}^{(\alpha)} = \mu_0(\boldsymbol{H} - \boldsymbol{M}^{(\alpha)}), \quad \boldsymbol{M} = J^{-1}[(1-\xi)\boldsymbol{M}_0^{(0)} + \xi\boldsymbol{M}_0^{(1)}]. \tag{45}$$

The magnetization per unit reference volume is $\boldsymbol{M}_0^{(\alpha)} = J\boldsymbol{M}^{(\alpha)}$. For an isotropic solid, $\boldsymbol{B}^{(\alpha)}||\boldsymbol{H}$.[29] It follows from Eq. 45 that $\boldsymbol{M}^{\alpha}||\boldsymbol{H}$, meaning

$$\begin{aligned}
&\boldsymbol{H} = H\boldsymbol{i}, \quad \boldsymbol{B}^{(\alpha)} = B^{(\alpha)}\boldsymbol{i}, \quad \boldsymbol{M}^{(\alpha)} = M^{(\alpha)}\boldsymbol{i} = J^{-1}M_0^{(\alpha)}\boldsymbol{i};\\
&H = \boldsymbol{H}\cdot\boldsymbol{i}, \quad B^{(\alpha)} = \boldsymbol{B}^{(\alpha)}\cdot\boldsymbol{i}, \quad M^{(\alpha)} = \boldsymbol{M}^{(\alpha)}\cdot\boldsymbol{i};\\
&\boldsymbol{i} = \boldsymbol{H}/H = \boldsymbol{B}^{(\alpha)}/B^{(\alpha)} = \boldsymbol{M}^{(\alpha)}/M^{(\alpha)} = \boldsymbol{i}^{(0)} = \boldsymbol{i}^{(1)};\\
&\boldsymbol{M} = (1-\xi)\boldsymbol{M}^{(0)} + \xi\boldsymbol{M}^{(1)} = M\boldsymbol{i}, \qquad \boldsymbol{B} = \mu_0(\boldsymbol{H}+\boldsymbol{M}) = B\boldsymbol{i}.
\end{aligned} \tag{46}$$

In other words, the vectors of magnetic field, local magnetic flux, and local magnetization are parallel among all coexisting phases at $(\boldsymbol{X}, t)$, as are their macroscopic counterparts. This description simplifies the more rigorous frame-indifferent formulation of Clayton and Lloyd.[11] Signed magnitude H, which can be positive or negative, is the same in each coexisting phase, but $M^{(\alpha)}$, and therefore $B^{(\alpha)}$, can differ locally among coexisting phases. Since $\boldsymbol{B}||\boldsymbol{M}$, verification is straightforward that Cauchy stress $\boldsymbol{t}$ is symmetric in Eq. 5:

$$\boldsymbol{t} = \boldsymbol{t}^{\mathrm{T}} = \boldsymbol{\sigma} = \boldsymbol{t}^{E} + (MB)\boldsymbol{i}\otimes\boldsymbol{i}. \tag{47}$$

Denote the saturation magnetization by M^S. In typical ferromagnetic ferrous alloys, the contribution of magnetostriction energy to magnetization[11] is at most on the order of $10^{-2}M^S$ and is deemed negligible in CMF1.

The magnitude of the local magnetization per unit reference volume, $M_0^{(\alpha)}$, of a phase of an alloy with complex chemistry is furnished as a combined function, not necessarily analytical, of magnetic field H and temperature θ:

$$M_0^{(\alpha)} = M_0^{(\alpha)}(H,\theta) \quad \Rightarrow \quad \boldsymbol{M}^{(\alpha)}(\boldsymbol{H},\theta,J) = J^{-1}M_0^{(\alpha)}(H,\theta)\,\boldsymbol{i}. \tag{48}$$

Quadratic functions used by Clayton et al.[12] are obtained from modeling with theoretical formulae, elemental mixture rules, and thermodynamic databases.[6,7] The

magnetostatic energy is likewise a function of local field and temperature:

$$\Phi^{(\alpha)}(H,\theta) = -\mu_0 \int_{\boldsymbol{0}}^{\boldsymbol{H}} \boldsymbol{M}_0^{(\alpha)} \cdot \mathrm{d}\hat{\boldsymbol{H}} = -\mu_0 \int_0^H M_0^{(\alpha)}(\hat{H},\theta)\,\mathrm{d}\hat{H}. \tag{49}$$

As a standard example, assume a ferromagnetic phase at temperature θ is saturated with magnetization magnitude $M^S(\theta) \geq 0$. In this case, with $\mathrm{sgn}(\cdot)$ denoting the signum function, the local signed magnetization is $M_0^{(\alpha)}(H,\theta) = M^{S(\alpha)}(\theta)\mathrm{sgn}(H)$, directed parallel to and in the same sense as $\boldsymbol{H}$. Then Eq. 49 produces a field energy $\Phi^{(\alpha)}(\theta) = -\mu_0 H M^{S(\alpha)}(\theta) \leq 0$, consistent with other theories.[4,50,51]

Two local magnetic loading protocols are addressed in algorithms applied at a material point with space-time coordinates $(\boldsymbol{x}(\boldsymbol{X},t),t)$. In both situations, assume ξ, $\boldsymbol{F}$ and θ are locally known.

In the first, $\boldsymbol{H} = H\boldsymbol{i}$ is imposed pointwise, where both H and $\boldsymbol{i}$ are given explicitly. Since $\boldsymbol{i}$ is given, $B = \mu_0(H+M)$, though none of B, H, M is required to be non-negative. In this case, Eq. 48 is directly solved for magnetization in each phase α. Then the last line of Eq. 46 furnishes the average (global) magnetization $\boldsymbol{M}$ and total magnetic flux density $\boldsymbol{B}$. The magnetostatic energy is found directly from Eq. 49.

In the second protocol, $\boldsymbol{B} = B\boldsymbol{i}$ is imposed, where both B and $\boldsymbol{i}$ are given explicitly. Again, $B = \mu_0(H+M)$ and none of B, H, M is required to be non-negative. Then inserting Eq. 48 into the second of Eq. 14 and taking the scalar product with $\boldsymbol{i}$ produces an implicit equation to be solved for H:

$$H(B,\theta,\xi,J) = \mu_0^{-1}B - J^{-1}\{(1-\xi)M_0^{(0)}(H,\theta) + \xi M_0^{(1)}(H,\theta)\}. \tag{50}$$

For general constitutive functions in Eq. 48, this equation is solved numerically. For simple functions (e.g., $M_0 = M^S =$ constant), analytical solutions are possible. Once H is found from Eq. 50, local magnetization in each phase is found from Eq. 48 and local magnetostatic energy from Eq. 49. Average magnetization $\boldsymbol{M}$ is then obtained from the last line of Eq. 46.

Average specific heat is prescribed similarly to Eqs. 48 and 49, with $\hat{c}_V^{(\alpha)}$ the specific

heat of phase α at reference temperature θ_0 and null $\boldsymbol{H}$:

$$c_V(H,\theta,\xi) = (1-\xi)\left(\hat{c}_V^{(0)} - \theta\frac{\partial^2\Phi^{(0)}(H,\theta)}{\partial\theta^2}\right) + \xi\left(\hat{c}_V^{(1)} - \theta\frac{\partial^2\Phi^{(1)}(H,\theta)}{\partial\theta^2}\right). \quad (51)$$

In Clayton et al.,[12] specific heats are obtained from modeling with theoretical formulae, elemental mixture rules, and thermodynamic databases.[6,7] Derivatives of magnetic energy density $\partial\Phi^{(\alpha)}/\partial\theta$ ($\alpha = 0, 1$) are obtained the same way as c_V and are likewise used in the temperature rate equation that follows in Section 2.1.7.

In the present software implementation (i.e., the MAGUMAT), properties that depend on magnetic field are approximated by quadratic polynomials at $\theta = \theta_0 = 300$ K, namely total specific heat per unit reference volume, magnetization per unit reference volume, magnetostatic energy per unit reference volume, and the temperature derivative of magnetostatic energy per unit reference volume:

$$\begin{aligned}
c_V^{(\alpha)} &= c_{V0}^{(\alpha)} + c_{V1}^{(\alpha)}\mu_0|H| + c_{V2}^{(\alpha)}(\mu_0)^2|H|^2,\\
M_0^{(\alpha)} &= m_0^{(\alpha)} + m_1^{(\alpha)}\mu_0|H| + m_2^{(\alpha)}(\mu_0)^2|H|^2,\\
\Phi^{(\alpha)} &= \Phi_0^{(\alpha)} + \Phi_1^{(\alpha)}\mu_0|H| + \Phi_2^{(\alpha)}(\mu_0)^2|H|^2,\\
\Phi'^{(\alpha)} &= \partial\Phi^{(\alpha)}/\partial\theta = \Phi_0'^{(\alpha)} + \Phi_1'^{(\alpha)}\mu_0|H| + \Phi_2'^{(\alpha)}(\mu_0)^2|H|^2.
\end{aligned} \quad (52)$$

The right side of each equation above contains three material constants, with each such constant having a subscript 0, 1, or 2. Since each material point is idealized as an isotropic polycrystal typically consisting of a mixture of coexisting phases, average magnetization $\boldsymbol{M}$ is assumed to vanish in the absence of an external aligning field. In such cases, ferromagnetic domains are randomly oriented to minimize total energy.[23,29]

2.1.6 Stress and Stress Rate

The lattice-originated part of the Cauchy stress, $\boldsymbol{t}^E$ in Eq. 8, consists of the lattice pressure p^E from the EOS of Eq. 16, the deviatoric thermoelastic contribution from the elastic shear modulus $\bar{\boldsymbol{t}}^e$, and the magnetostriction contribution $\bar{\boldsymbol{t}}^M$, where the latter is deviatoric by construction[11,50]:

$$\boldsymbol{t}^E = -p^E\mathbf{1} + \bar{\boldsymbol{t}}^e + \bar{\boldsymbol{t}}^M = \boldsymbol{\sigma} - (MB)\boldsymbol{i}\otimes\boldsymbol{i}, \quad (53)$$

$$\bar{\boldsymbol{t}}^M = -J^{-1}\mathsf{G}_0\left[(1-\xi)e_{||}^{M(0)} + \xi e_{||}^{M(1)}\right](3\boldsymbol{i}\otimes\boldsymbol{i} - \mathbf{1}). \quad (54)$$

The initial shear modulus G_0 is identical in each phase, and magnetostriction strain constants for each phase are $e_{||}^{M(\alpha)}$.

The dynamic numerical implementation of the UMAT updates the deviatoric spatial elastic stress tensor $\bar{\boldsymbol{t}}^e$ by integrating an objective rate form over time. The mathematically consistent objective rate is a Jaumann rate derived from logarithmic thermoelasticity[30] under the assumptions of small deviatoric elastic stretch and small deviatoric elastic stretch rate[12]:

$$\overset{\nabla}{\bar{\boldsymbol{t}}}^e = 2\hat{\mathsf{G}}[\bar{\boldsymbol{d}} - \bar{\boldsymbol{d}}^P] = \dot{\bar{\boldsymbol{t}}}^e - \boldsymbol{\omega}\bar{\boldsymbol{t}}^e + \bar{\boldsymbol{t}}^e\boldsymbol{\omega} + \left[\frac{\dot{\phi}}{1-\phi} + \mathrm{tr}\boldsymbol{d}\right]\bar{\boldsymbol{t}}^e; \tag{55}$$

$$\hat{\mathsf{G}} = \frac{1}{J}(1-\phi)\mathsf{G}_0. \tag{56}$$

The tangent shear modulus is $\hat{\mathsf{G}}$. Spin tensor $\boldsymbol{\omega}$ is the skew part of $\nabla\boldsymbol{v}$, which is equal to the skew part of $\boldsymbol{l}^E$ of Eq. 19. Denoted by $\boldsymbol{d} = \nabla\boldsymbol{v} - \boldsymbol{\omega}$ is the deformation rate tensor with deviatoric part $\bar{\boldsymbol{d}}$, and the deviatoric plastic deformation rate is $\bar{\boldsymbol{d}}^P$.

The local yield condition with magnetic coupling entering Eq. 20 is obtained from similar small deviatoric elasticity assumptions as[12]

$$\sqrt{\tfrac{3}{2}}(\bar{\boldsymbol{t}}^e : \bar{\boldsymbol{t}}^e)^{1/2} \leq \{\xi K^{P(1)} + (1-\xi)K^{P(0)}\} \cdot (1 - c^\phi\phi) + MB \geq 0. \tag{57}$$

Algorithms assume that magnetostriction stress is small compared to the yield stress.[11] Also assumed negligible in the UMAT is the product of deviatoric thermoelastic strain with MB.

2.1.7 Temperature

The local temperature rate derived by Clayton et al.,[12] simplified for cases considered in the present software implementation, is

$$\begin{aligned} c_V\dot{\theta} =& \mathfrak{D}^P + \mathfrak{D}^\xi + \mathfrak{D}^\phi + J[\Sigma|\mathbf{E}|^2 + \kappa\nabla^2\theta] \\ & - \theta c_V\gamma_0\frac{\dot{J}^E}{J^E} + \theta\left[\frac{\partial(\Delta^*\Phi)}{\partial\theta} - (\Delta^*A)\mathsf{B}_0\ln J^E - \Delta^*\hat{c}_V\ln\frac{\theta}{\theta_0} - \frac{\lambda_T}{\theta_T}\right]\dot{\xi}. \end{aligned} \tag{58}$$

The first three terms on the right of Eq. 58 are dissipation from plasticity, phase transitions, and damage kinetics. In the latter, contributions of deviatoric strain en-

ergy to $\hat{\omega}$ of Eq. 42 are omitted in the numerics, consistent with the small deviatoric elastic stretch assumptions used in Section 2.1.6. The next two terms are dissipation from isotropic electrical and thermal conduction. Local point heat sources are absent: $r = 0$. On the subsequent line, contributions from thermoelastic coupling (i.e., temperature change from a nonzero Grüneisen parameter γ_0) and phase differences in thermal-magnetostatic energy, thermal expansion coefficients, specific heat, and latent heat follow sequentially.

2.2 CMF2

The CMF2 implementation is based on the theory of Clayton and Lloyd,[10] which omits explicit magnetization, meaning $\boldsymbol{M} = \boldsymbol{0}$ for each phase and for the mixture. Isotropic heat conduction and electric conduction, the latter not addressed by the original model of Clayton and Lloyd,[10] are both supported by the present software implementation within the MAGUMAT.

Governing equations listed in the preliminary part of Section 2 all still hold, but are reduced as follows. Equations 1–4 apply verbatim. Equations 5–8 become

$$\nabla \cdot \boldsymbol{\sigma} + \tilde{\boldsymbol{f}} + \boldsymbol{f} = \rho \dot{\boldsymbol{v}}, \qquad \boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathrm{T}}, \qquad \tilde{\boldsymbol{f}} = \boldsymbol{J} \times \boldsymbol{B}; \tag{59}$$

$$\boldsymbol{\sigma} = \bar{\boldsymbol{\sigma}} - p\boldsymbol{1}, \qquad p = -\tfrac{1}{3}\mathrm{tr}\,\boldsymbol{\sigma}, \tag{60}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress with deviatoric part $\bar{\boldsymbol{\sigma}}$ and p is the Cauchy pressure. Equations 10, 11, and inequality 12 remain the same (i.e., no simplifications). Equation 9 and inequality 13 reduce to, respectively,

$$J^{-1}\dot{U} = \boldsymbol{\sigma} : \nabla \boldsymbol{v} + \mathbf{J} \cdot \mathbf{E} - \nabla \cdot \boldsymbol{q} + r, \qquad (\boldsymbol{\sigma} : \nabla \boldsymbol{v} = \sigma_{ji} \partial_j v_i); \tag{61}$$

$$-(1/J)(\dot{\psi} + \eta\dot{\theta}) + \boldsymbol{\sigma} : \nabla \boldsymbol{v} + \mathbf{J} \cdot \mathbf{E} - \{\boldsymbol{q}/\theta\} \cdot \nabla\theta \geq 0. \tag{62}$$

Maxwell's Eqs. 15 are unchanged, but the second of Eq. 14 reduces to $\boldsymbol{B} = \mu_0 \boldsymbol{H}$. In the terminology of CMF1, stress and pressure of CMF2 are simply

$$\boldsymbol{t} = \boldsymbol{t}^E = \boldsymbol{\sigma}, \qquad p = p^E. \tag{63}$$

2.2.1 EOS

An Eulerian EOS relates pressure p to thermoelastic volume change J^E and temperature θ. This EOS can be derived from third-order nonlinear elasticity theory with a dependence of free energy on the material Eulerian strain[25,52]:

$$\begin{aligned} p = \frac{J^E}{J}[\tfrac{3}{2}\mathsf{B}_0(J^{E\,-7/3} - J^{E\,-5/3}) \\ \cdot \{\zeta^B - \tfrac{3}{4}(\mathsf{B}_0' - 4)(1 - J^{E\,-2/3})\} + J^{E\,-5/3} c_V \gamma_0 \Delta\theta]. \end{aligned} \tag{64}$$

The initial bulk modulus, pressure derivative of the bulk modulus, and Grüneisen parameter are respective constants B_0, B_0', and γ_0. The specific heat per unit reference volume at constant volume is a constant[10] (i.e., differences in specific heats of phases, temperature, and magnetic field dependence are ignored in CMF2):

$$c_V = \hat{c}_V = c_V^{(0)} = c_V^{(1)} = \text{constant}. \tag{65}$$

The tangent bulk modulus B degrades with damage for tensile states and the tangent shear modulus G degrades regardless of local deformation state, where ζ^B and ζ^G are non-negative functions of ϕ:

$$\mathsf{B}(\phi, J^E) = \zeta^B(\phi, J^E)\mathsf{B}_0, \qquad \mathsf{G}(\phi) = \zeta^G(\phi)\mathsf{G}_0. \tag{66}$$

The Poisson's ratio of the undamaged material at its initial state is

$$\nu_0 = (3\mathsf{B}_0 - 2\mathsf{G}_0)/(6\mathsf{B}_0 + 2\mathsf{G}_0). \tag{67}$$

Dilatational and deviatoric strain energy densities are affected by voids using coefficients obtained from the analysis of Mackenzie[53]:

$$\kappa^B = \frac{3(1-\nu_0)}{(1+\nu_0)\phi + 2(1-2\nu_0)}, \qquad \kappa^G = \frac{15(1-\nu_0)}{7-5\nu_0}. \tag{68}$$

Notice that κ^B is not a constant since it depends on ϕ. The following degradation functions[54] are then used for $0 \le \phi < 1$:

$$\zeta^B(\phi, \hat{q}(J^E)) = \begin{cases} 1 - \kappa^B\phi & \text{if } \hat{q} \ge 0 \text{ and } \phi < 1/\kappa^B, \\ 0 & \text{if } \hat{q} \ge 0 \text{ and } \phi \ge 1/\kappa^B, \\ 1 & \text{if } \hat{q} < 0; \end{cases} \tag{69}$$

$$\zeta^G(\phi) = \begin{cases} 1 - \kappa^G \phi & \text{if } \phi < 1/\kappa^G, \\ 0 & \text{if } \phi \geq 1/\kappa^G. \end{cases} \tag{70}$$

The value of function $\hat{q} = J^E - 1$ indicates elastic compression ($\hat{q} < 0$) versus tension ($\hat{q} > 0$) or neutral loading ($\hat{q} = 0$). If $\kappa^B \geq 1$ or $\kappa^G \geq 1$, the corresponding minimum of ζ^B or ζ^G is set to zero when $\phi \geq 1/\kappa^B$ or $\phi \geq 1/\kappa^G$, respectively.

2.2.2 Plastic Flow

The composite plastic flow theory for CMF1 in Section 2.1.2 is likewise invoked in CMF2, for which it originated in Clayton and Lloyd,[10] with the hardening function of Eq. 29 credited to Lloyd et al.[2] The following simplifications arise for CMF2 relative to CMF1. First, the Cauchy stress rather than Kirchhoff stress is used in the yield and flow potential Ω^P, and possible dependence of yield and flow stress on $\boldsymbol{H}$ is omitted, meaning Eq. 20 is

$$\Omega^P = \sqrt{\tfrac{2}{3}}(\bar{\sigma}^V - K^P), \qquad K^P = K^P(e^P, \dot{e}^P, \xi, \phi, \theta). \tag{71}$$

Second, since $\boldsymbol{M} = \boldsymbol{0}$ in CMF2, the local von Mises stress in Eq. 21 is now

$$\bar{\sigma}^V = \sqrt{\tfrac{3}{2}}|\bar{\boldsymbol{\sigma}} : \bar{\boldsymbol{\sigma}}|^{1/2} = \sqrt{\tfrac{3}{2}}|(\boldsymbol{\sigma} + p\boldsymbol{1}) : (\boldsymbol{\sigma} + p\boldsymbol{1})|^{1/2}, \qquad \boldsymbol{\sigma} = \boldsymbol{t}^E. \tag{72}$$

Finally, any possible dependence of phase interaction factor ι of Eq. 24 on $\boldsymbol{H}$ and θ is likewise omitted, leading to

$$\iota = \iota^{(0)} = \iota^{(1)} = \text{constant}. \tag{73}$$

2.2.3 Phase Transformations

A notably different phase transition model is used in CMF2 versus that of CMF1 in Section 2.1.3. The model of CMF2, established by Clayton and Lloyd,[10] is based on earlier thermodynamic ideas of Turteltaub and Suiker.[55,56] Like the theory of CMF1 and Clayton et al.,[11,12] the theory of CMF2 is thermodynamically consistent, meaning that dissipation from transition kinetics is always non-negative. However, the theory of CMF2 does not explicitly use the Gibbs free energy difference $\Delta^*\mathbb{G}$ as the driving force, nor is a linear relaxation kinetic equation toward a metastable state employed.

Rather, the sum of Helmholtz free energy difference and stress power of transformation act as driving forces[10] and a nonlinear differential equation is prescribed directly for the transformation rate in CMF2. Since magnetization and specific heat differences between phases are ignored in CMF2, the Helmholtz free energy difference only includes the latent heat (e.g., thermal) contribution in addition to an athermal resistance that can be related to defect content[10] or more generally to ψ_0 of CMF1. Finally, the theory and implementation of CMF2 only address forward transitions:

$$\dot{\xi} \geq 0. \tag{74}$$

The condition in Eq. 74 are adequate for modeling deformation-driven forward transitions in TRIP steels.[10,40,41] The kinetic rules that follow have yet to be thoroughly tested for modeling transitions driven purely by temperature and pressure differences in the absence of deviatoric flow, that is, hydrostatic conditions when transformation shear is zero. Restrictions of Eq. 74 do not apply for CMF1 of Section 2.1.3.

The dissipation per unit reference volume from phase transitions in CMF2 is[10]

$$\mathfrak{D}^{\xi} = [f_{\sigma} + f_{\theta} + f_0]\dot{\xi} \geq 0. \tag{75}$$

Mechanical, thermal, and athermal driving forces are, respectively,

$$f_{\sigma} = -pJ\delta^{\xi}/[1 + (\xi - \xi_0)\delta^{\xi}] + \sqrt{\tfrac{1}{6}}J\bar{\sigma}^{V}\gamma^{\xi}, \tag{76}$$

$$f_{\theta} = (\lambda_T/\theta_T)(\theta - \theta_T), \qquad f_0 = -\psi_0 < 0. \tag{77}$$

The sum $f_{\sigma} + f_{\theta} + f_0$ equals $-\Delta^{*}\mathbb{G}$ of CMF1 and Section 2.1.3 when magnetostatic energy and specific heat differences between phases vanish in Eq. 41. Since only forward transitions are permitted, the datum free energy of the transformed phase ψ_0 must be positive; otherwise, spontaneous reverse transformation should occur, and these are precluded by Eq. 74.

The kinetic law for transformation per unit volume ensuring that $\mathfrak{D}^{\xi} \geq 0$ is*:

$$\frac{\dot{\xi}}{1-\xi} = \dot{\xi}_m \tanh\left\langle \frac{1}{\mu_{\xi}} \frac{f_{\sigma} + f_{\theta} + f_0}{f_0} \right\rangle \cdot \left[1 + \mathrm{H}(\dot{e}^P - \dot{e}_1)\{\left(\dot{e}^P/\dot{e}_1\right)^{-R} - 1\}\right]. \tag{78}$$

*The term in brackets in Eq. 78 corrects a misprint in Eq. 74 and Box 1 of Clayton and Lloyd.[10]

The maximum transformation rate is $\dot{\xi}_m \geq 0$, and a dimensionless viscosity constant is $\mu_\xi > 0$. Note that $\tanh(0) = 0$ and $\lim_{x\to\infty} \tanh(x) = 1$. Denoted by $\dot{e}_1$ is constant reference strain rate above which the time scale for transformation kinetics limits the maximum effective transformation rate relative to the total plastic strain rate $\dot{e}^P$. Dimensionless constant $R \geq 0$ controls the magnitude of this rate-limiting effect.

The maximum transformation rate $\dot{\xi}_m$ is *not* always constant; it occurs when the deviatoric transformation strain rate approaches the total plastic strain rate[10]:

$$\dot{\xi}_m = \begin{cases} 2\dot{\epsilon}^P/\gamma^\xi & (\gamma^\xi > 0 \text{ and } \dot{\epsilon}^P > 0), \\ \dot{\xi}_0 = \text{constant} \geq 0 & (\gamma^\xi = 0 \text{ and/or } \dot{\epsilon}^P = 0). \end{cases} \tag{79}$$

The material constant $\dot{\xi}_0$ modulates transition conditions that have no shear strain and/or for which total deviatoric plastic flow is nonexistent (e.g., purely thermal or hydrostatic loadings). The total effective inelastic strain rate $\dot{\epsilon}^P = \sqrt{\frac{3}{2}}\dot{e}^P$ consists of summed contributions from slip, twinning, and phase transformations (deviatoric part only). Relations in Eq. 79 are necessary conditions that prohibit the transformation contribution from exceeding this sum. The product of other terms multiplying $\dot{\xi}_m$ in Eq. 78 is always less than or equal to unity according to definitions of each variable. In the present numerical implementation of CMF2, the twinning shear γ^ξ is a user-defined, non-negative constant.

2.2.4 Damage Mechanics

The implementation of damage kinetics for CMF2 is nearly identical to that of CMF1 described in Section 2.1.4. Equations 42–44 apply for CMF2. The only distinction is that for CMF2, $\hat{\omega} = -\partial\psi/\partial\phi$ is different than that for CMF1, since the elastic strain energy density degrades differently in the two frameworks.

In CMF2, $\hat{\omega}$ follows from differentiation of Eq. 66 with respect to ϕ, while in CMF1, $\hat{\omega}$ follows from differentiation of Eq. 18 with respect to ϕ. Details are contained in the original references.[10–12] In both frameworks, $\mathfrak{D}^\phi \geq 0$ and $\dot{\phi} \geq 0$ ensure physical and thermodynamic irreversibility of damage.

2.2.5 Stress and Stress Rate

The total Cauchy stress is $\boldsymbol{\sigma} = \bar{\boldsymbol{\sigma}} - p\mathbf{1}$, where the pressure p is given by the Eulerian EOS in Eq. 64. Unlike CMF1, the theory of CMF2 does not consider magnetostric-

tion, so $\bar{\boldsymbol{\sigma}}$ results solely from deviatoric elasticity (i.e., the effect of the tangent shear modulus $\hat{\mathsf{G}}$) in CMF2.

The dynamic numerical implementation of the UMAT updates the deviatoric spatial stress tensor $\bar{\boldsymbol{\sigma}}$ by integrating an objective rate form over time. The objective rate is a modified Jaumann-type rate derived from Eulerian thermoelasticity[52] under the assumptions of small deviatoric elastic stretch and small deviatoric elastic stretch rate[10]:

$$\overset{\nabla}{\bar{\boldsymbol{\sigma}}} = 2\hat{\mathsf{G}}[\bar{\boldsymbol{d}} - \bar{\boldsymbol{d}}^P] = \dot{\bar{\boldsymbol{\sigma}}} - \boldsymbol{\omega}\bar{\boldsymbol{\sigma}} + \bar{\boldsymbol{\sigma}}\boldsymbol{\omega} - \left[\frac{\dot{\phi}}{\zeta^G}\frac{\mathrm{d}\zeta^G}{\mathrm{d}\phi} - \tfrac{5}{3}\mathrm{tr}\boldsymbol{d} + \tfrac{2}{3}\mathrm{tr}\boldsymbol{d}^P\right]\bar{\boldsymbol{\sigma}}; \tag{80}$$

$$\hat{\mathsf{G}} = \frac{J^{E-4/3}}{J}\mathsf{G} = \frac{J^{E-4/3}}{J}\zeta^G\mathsf{G}_0. \tag{81}$$

The same notation used in Eq. 55 applies for Eq. 80. The tangent shear modulus is $\hat{\mathsf{G}}$. Spin tensor $\boldsymbol{\omega}$ is the skew part of $\nabla\boldsymbol{v}$ and $\boldsymbol{l}^E$ of Eq. 19, $\boldsymbol{d} = \nabla\boldsymbol{v} - \boldsymbol{\omega}$ is the deformation rate tensor with deviatoric part $\bar{\boldsymbol{d}}$, and the deviatoric plastic deformation rate is $\bar{\boldsymbol{d}}^P$.

2.2.6 Temperature

The local temperature rate derived by Clayton and Lloyd,[10] modified to allow for cases considered in the present software implementation, is

$$c_V\dot{\theta} = \mathfrak{D}^P + \mathfrak{D}^\xi + \mathfrak{D}^\phi + J[\Sigma|\mathsf{E}|^2 + \kappa\nabla^2\theta] - \theta c_V\gamma_0\frac{\dot{J}^E}{J^{E\,5/3}} - \theta\frac{\lambda_T}{\theta_T}\dot{\xi}. \tag{82}$$

The first three terms* on the right of Eq. 82 are dissipation per unit reference volume from plasticity, phase transitions, and damage kinetics. In $\mathfrak{D}^\phi$, contributions of deviatoric strain energy to $\hat{\omega}$ are omitted in the algorithms, consistent with the small deviatoric elastic stretch assumptions used in Section 2.2.5. The next two terms are dissipation from isotropic electrical and thermal conduction. Local point heat sources are absent: $r = 0$. Contributions from thermoelastic coupling and latent heat follow subsequently.

*These terms were missing a factor of J in some instances (e.g., Box 1) of the equations in the original publication,[10] but numerical calculations in that work correctly included this factor. The complete set of correct thermodynamic equations is documented by Clayton et al.[12]

3. Input Constants

Input constants for CMF1 are tabulated in Section 3.1, and input constants for CMF2 are tabulated in Section 3.2. The syntax for an ABAQUS-style UMAT input file is given in Section 3.3. The current software implementation accepts 96 input constants for each CMF, though not all entries are used. Some constants are dummy variables carried over from legacy code with other modeling options not documented herein. The input constants are labeled by number from 1 to 96 in forthcoming tables of Sections 3.1 and 3.2 in the sequence in which they appear in the input file syntax of Section 3.3.

Temperature is always in units of Kelvin (K). In the mechanical UMAT, pressure units (i.e., stress or energy per unit volume) are arbitrary, labeled by P. Time units are likewise arbitrary in the mechanical UMAT, labeled by t. Note that SI units in the MKS system are enforced for electromagnetic quantities in the MAGUMAT, and thus for certain entries in the input file of syntax in Section 3.3. Transformations to other systems of units are avoided since these are cumbersome.[23] Regarding units for Σ, the conjugate electric field $\mathbf{E}$ to electric current $\mathbf{J}$ is assumed to be provided to the MAGUMAT by the host code in unscaled SI units of $\mathrm{V{\cdot}m^{-1}}$. Regarding units for κ, the gradient and divergence of temperature, respectively $\nabla\theta$ and $\nabla^2\theta$, are assumed to be provided by the host code in respective unscaled SI units of $\mathrm{K{\cdot}m^{-1}}$ and $\mathrm{K{\cdot}m^{-2}}$.

Mass density ρ_0 is not a required input constant for the syntax of Section 3.3, but it will be required by any dynamic host code in order to calculate stress wave speeds and inertial forces. In such cases, mass density is typically input by other means.

3.1 CMF1

Input constants 1 through 32 for CMF1 are identified in Table 1, constants 33 through 64 in Table 2, and constants 65 through 96 in Table 3. The first (i.e., leftmost) column numbers each constant according to the scheme just described. The second column labels each constant by its name in the FORTRAN source code of the UMAT subroutine. The third column gives the data type: floating point number (F) or integer (I). The fourth column gives the mathematical symbol used in the theoretical formulation of Section 2.1 and/or the original references.[11,12] The fifth column gives the units. The sixth (i.e., rightmost) column gives the definition.

Table 1 Input constants 1–32 for CMF1

Number	Name	Type	Symbol	Units	Definition
1	lambda	F	λ_0	P	Lamé modulus $\lambda_0 = \mathsf{B}_0 - \frac{2}{3}\mathsf{G}_0$
2	mu0	F	G_0	P	undamaged reference shear modulus
3	theta0	F	θ_0	K	reference temperature (not necessarily initial θ)
4	tempmelt	F	θ_M	K	melting temperature
5	hsltype	I	-	-	set to 3 for plastic potential of CMF1 and CMF2
6	hsl1	F	$\sigma_0^{(0)}$	P	initial yield strength of parent phase
7	hsl2	F	$\Theta_0^{(0)}$	P	affects strain hardening of parent phase
8	hsl3	F	$\delta^{(0)}$	-	affects strain hardening of parent phase
9	hsl4	F	$\epsilon^{(0)}$	-	affects strain hardening of parent phase
10	hsl5	F	$\dot{e}_0$	t^{-1}	reference normalization strain rate
11	hsl6	F	$m^{(\alpha)}$	-	strain rate sensitivity; same for all phases
12	hsl7	F	$p^{(\alpha)}$	-	thermal softening; same for all phases
13	hsl8	F	$\dot{e}_{\min}$	t^{-1}	minimum allowed value of $\dot{e}^P$ in Eq. 28
14	dmgtype	I	-	-	damage model type*
15	d1	F	$\hat{m}$	-	damage growth parameter
16	d2	F	c_0	-	damage nucleation parameter
17	d3	F	-	-	not used; set to 0.0
18	d4	F	-	-	not used; set to 0.0
19	d5	F	-	-	not used; set to 0.0
20	d6	F	-	-	not used; set to 0.0
21	d7	F	-	-	not used; set to 0.0
22	d8	F	ϕ_c	-	max local damage at failure*
23	d9	F	ϕ_m		enforces* $c^\phi = \phi_m/\phi$ for $\phi \geq \phi_m$
24	d10	F	$p_{\min}$	P	limit tensile pressure if $\phi \geq \phi_c$
25	eostype	I	-	-	set to 3 for EOS of Eq. 16; -3 for $\dot{\theta} = 0$
26	cv0	F	$\hat{c}_{V0}$	$\mathrm{P{\cdot}K}^{-1}$	initial specific heat per unit volume
27	b0	F	B_0	P	undamaged reference bulk modulus
28	dbdp	F	B_0'	-	pressure derivative of bulk modulus
29	cte	F	A	K^{-1}	vol. thermal expansion; $\gamma_0 = A\mathsf{B}_0/\hat{c}_{V0}$
30	gruna	F	-	-	not used; set to 0.0
31	ecolds	F	-	-	not used; set to 0.0
32	ecold0	F	-	-	not used; set to 0.0

* see text

Table 2 Input constants 33–64 for CMF1

Number	Name	Type	Symbol	Units	Definition
33	ecold1	F	-	-	not used; set to 0.0
34	ecold2	F	-	-	not used; set to 0.0
35	beta	F	$\hat{\beta}$	-	constant Taylor-Quinney factor
36	objrate	I	-	-	objective rate formulation type*
37	vf0	F	ξ_0	-	initial fraction of second (product) phase
38	pttype	I	-	-	set to 4 for $\dot{\xi}$ of Eq. 40; set to 0 for $\dot{\xi} = 0$
39	pt1	F	δ^{ξ}	-	transformation volume change
40	pt2	F	γ_0^{ξ}	-	tranformation shear strain
41	pt3	F	α_0^F	P	forward transition barrier
42	pt4	F	α_0^R	P	reverse transition barrier
43	pt5	F	β_0^F	P	forward transition factor
44	pt6	F	β_0^R	P	reverse transition factor
45	pt7	F	ψ_0	P	free energy offset of product phase
46	pt8	F	λ_T	P	latent heat parameter
47	pt9	F	θ_T	K	transition temperature
48	pt10	F	ι_0	-	transition strength accommodation
49	pt11	F	$\sigma_0^{(1)}$	P	initial yield strength of product
50	pt12	F	$\Theta_0^{(1)}$	P	affects strain hardening of product
51	pt13	F	$\delta^{(1)}$	-	affects strain hardening of product
52	pt14	F	$\epsilon^{(1)}$	-	affects strain hardening of product
53	pt15	F	τ^F	10^{-9}t	forward relaxation time (ns, SI units)
54	pt16	F	τ^R	10^{-9}t	reverse relaxation time (ns, SI units)
55	hsl9	F	k_s	-	strength amplification, viscous drag
56	hsl10	F	$\dot{e}_s$	t^{-1}	normalizing strain rate, viscous drag
57	mskip	I	-	-	set to 1 for $\|M\| \geq 0$; if 0, $\|M\| = 0$*
58	stressfac	F	-	-	conversion factor = Pa/P to SI unit*
59	econd	F	Σ	10^6A(Vm)$^{-1}$	electrical conductivity
60	tcond	F	κ	W(Km)$^{-1}$	thermal conductivity
61	em1	F	$e_{\|\|}^{M(0)}$	10^{-6}	magnetostriction μ-strain, parent*
62	em2	F	$e_{\|\|}^{M(1)}$	10^{-6}	magnetostriction μ-strain, product*
63	hsat1	F	$\mu_0 H^{S(0)}$	T	$\|H\| \geq H^{S(0)} \Rightarrow M_0^{(0)} \rightarrow M^{S(0)*}$
64	hsat2	F	$\mu_0 H^{S(1)}$	T	$\|H\| \geq H^{S(1)} \Rightarrow M_0^{(1)} \rightarrow M^{S(1)*}$

* see text

Table 3 Input constants 65–96 for CMF1

Number	Name	Type	Symbol	Units	Definition
65	msat1	F	$\mu_0 M^{S(0)}$	T	saturation magnetization, parent
66	msat2	F	$\mu_0 M^{S(1)}$	T	saturation magnetization, product
67	temp0	F	θ_i	K	initial temperature $\theta(t=0)$
68	pt17	F	$\Delta^* A$	K^{-1}	difference in thermal expansion
69	pt18	F	α_1	$\mathrm{P{\cdot}T}^{-1}$	linear H-dependence of $\alpha^{F/R}$
70	pt19	F	β_1	$\mathrm{P{\cdot}T}^{-1}$	linear H-dependence of $\beta^{F/R}$
71	pt20	F	ι_1	T^{-1}	linear H-dependence of ι
72	magmod	I	-	-	quadratic spline if 1; saturation if 0*
73	ma1	F	$m_0^{(0)}$	T	constant mag term, parent phase
74	mb1	F	$m_1^{(0)}$	-	linear mag term, parent phase
75	mc1	F	$m_2^{(0)}$	T^{-1}	quadratic mag term, parent phase
76	ma2	F	$m_0^{(1)}$	T	constant mag term, product phase
77	mb2	F	$m_1^{(1)}$	-	linear mag term, product phase
78	mc2	F	$m_2^{(1)}$	T^{-1}	quadratic mag term, product phase
79	ga1	F	$\Phi_0^{(0)}$	P	constant mag energy term, parent
80	gb1	F	$\Phi_1^{(0)}$	$\mathrm{P{\cdot}T}^{-1}$	linear mag energy term, parent
81	gc1	F	$\Phi_2^{(0)}$	$\mathrm{P{\cdot}T}^{-2}$	quadratic mag energy term, parent
82	ga2	F	$\Phi_0^{(1)}$	P	constant mag energy term, product
83	gb2	F	$\Phi_1^{(1)}$	$\mathrm{P{\cdot}T}^{-1}$	linear mag energy term, product
84	gc2	F	$\Phi_2^{(1)}$	$\mathrm{P{\cdot}T}^{-2}$	quadratic mag energy term, product
85	dga1	F	$\Phi_0'^{(0)}$	$\mathrm{P{\cdot}K}^{-1}$	constant mag energy θ-derivative, parent
86	dgb1	F	$\Phi_1'^{(0)}$	$\mathrm{P{\cdot}K}^{-1}{\cdot}\mathrm{T}^{-1}$	linear mag energy θ-derivative, parent
87	dgc1	F	$\Phi_2'^{(0)}$	$\mathrm{P{\cdot}K}^{-1}{\cdot}\mathrm{T}^{-2}$	quadratic mag energy θ-derivative, parent
88	dga2	F	$\Phi_0'^{(1)}$	$\mathrm{P{\cdot}K}^{-1}$	constant mag energy θ-derivative, product
89	dgb2	F	$\Phi_1'^{(1)}$	$\mathrm{P{\cdot}K}^{-1}{\cdot}\mathrm{T}^{-1}$	linear mag energy θ-derivative, product
90	dgc2	F	$\Phi_2'^{(1)}$	$\mathrm{P{\cdot}K}^{-1}{\cdot}\mathrm{T}^{-2}$	quadratic mag energy θ-derivative, product
91	cva1	F	$c_{V0}^{(0)}$	$\mathrm{P{\cdot}K}^{-1}$	constant total specific heat, parent
92	cvb1	F	$c_{V1}^{(0)}$	$\mathrm{P{\cdot}K}^{-1}{\cdot}\mathrm{T}^{-1}$	linear total specific heat, parent
93	cvc1	F	$c_{V2}^{(0)}$	$\mathrm{P{\cdot}K}^{-1}{\cdot}\mathrm{T}^{-2}$	quadratic total specific heat, parent
94	cva2	F	$c_{V0}^{(1)}$	$\mathrm{P{\cdot}K}^{-1}$	constant total specific heat, product
95	cvb2	F	$c_{V1}^{(1)}$	$\mathrm{P{\cdot}K}^{-1}{\cdot}\mathrm{T}^{-1}$	linear total specific heat, product
96	cvc2	F	$c_{V2}^{(1)}$	$\mathrm{P{\cdot}K}^{-1}{\cdot}\mathrm{T}^{-2}$	quadratic total specific heat, product

* see text

Those constants that require further explanation, as marked by an asterisk in the sixth column, are discussed next:

- `dmgtype`: set to 6 for damage model of CMF1 and Section 2.1.4 with $c^\phi = 0$ (i.e., no dilatation, brittle damage); set to 7 for damage model of CMF1 and Section 2.1.4 with $c^\phi = 1$ (i.e., dilatation, ductile damage); set to 0 to disable damage model (i.e., $\phi = 0$ and $\dot{\phi} = 0$)
- `d8`$= \phi_c$: sets $\phi(t^+) \rightarrow 1$ when $\phi(t) \geq \phi_c$ to rapidly fail material
- `d9`$= \phi_m$: constrains maximum dilatation from voids to $c^\phi \phi_m$ to avoid singularity in J^ϕ as $\phi \rightarrow 1$
- `objrate`: set to 5 to invoke full version of Eq. 55; set to 4 to invoke Eq. 55 with $\boldsymbol{\omega} = \mathbf{0}$; set to 1 to invoke classic Jaumann rate (i.e., Eq. 55 with right term $[\cdot]\bar{\boldsymbol{t}}^e$ omitted); set to 0 to deactivate objective rate (i.e., $\overset{\nabla}{\bar{\boldsymbol{t}}}{}^e \rightarrow \dot{\bar{\boldsymbol{t}}}^e$ in Eq. 55)
- `mskip`: enables magnetization models of CMF1 if set equal to 1; disables magnetization, magnetic energy, and explicit magnetic contribution to specific heat if set to 0 (these MAGUMAT features become redundant)
- `stressfac`: converts base SI units to arbitrary pressure units used in mechanics calculations (e.g., if P is in MPa, `stressfac` $= 10^{-6}$); multiply quantity in base SI units of Pa by `stressfac` to obtain quantity in P units
- `em1`: implements magnetostriction strain of this magnitude in parent phase for $|H|$ equal to or exceeding `hsat1`
- `em2`: implements magnetostriction strain of this magnitude in product phase for $|H|$ equal to or exceeding `hsat2`
- `hsat1`: local field strength required to induce magnetic response in parent phase; if $|H|$ is less than `hsat1`, then magnetic field and magnetostriction are zero in parent phase
- `hsat2`: local field strength required to induce magnetic response in product phase; if $|H|$ is less than `hsat2`, then magnetic field and magnetostriction are zero in product phase

- `magmod`: set to 1 to enable quadratic splines for specific heat, magnetization, magnetostatic energy, and temperature derivative of magnetostatic energy in Eq. 52 of CMF1; set to 0 to invoke classic ferromagnetic saturation model[11,12] wherein $|H| \geq H^{S(\alpha)} \Rightarrow M_0^{(\alpha)} \rightarrow M^{S(\alpha)}$ with $M_0^{(\alpha)} = 0$ for $|H| < H^{S(\alpha)}$; setting `magmod` to 0 uses properties 63–66 but does not use properties 73–96 that are accessed when `magmod` is set to 1; the same model, but with possibly different parameters, must be used for all phases (i.e., only one value of 0 or 1 is input as `magmod` for both phases)

3.2 CMF2

Input constants 1 through 32 for CMF2 are identified in Table 4. Constants 33 through 64 are identified in Table 5. Constants 65 through 96 are identified in Table 6. The first (i.e., leftmost) column numbers each constant. The second column labels each constant by its name in the FORTRAN source code of the UMAT subroutine. The third column gives the data type: floating point number (F) or integer (I). The fourth column gives the mathematical symbol used in the theoretical formulation of Section 2.2 and/or the original reference.[10] The fifth column gives the units. The sixth (i.e., rightmost) column gives the definition of the constant.

Since CMF2 does not explicitly consider magnetization, many of the 96 input constants are not used. Notably, as is evident from Table 6, constants 68–96 are irrelevant for the standard CMF2 prescription of `mskip` of 0 that disables electromagnetism consistently with Section 2.2. However, the standard input file syntax of Section 3.3 requires that all 96 constants be supplied in this file; hence, floating point values of 0.0 should be entered for constants 68–96 when the input syntax expects all 96. Thermal and electrical conduction are still permissible and thermodynamically admissible in CMF2.

Regarding `eostype`, a positive value enables temperature change, while a negative value invokes the same theory but forces the temperature rate to zero, such that isothermal conditions are maintained. In the isothermal case, the temperature remains fixed at `temp0` (θ_i, input constant 67) for all calculations, including the EOS, plastic yield and flow rules, and phase transition kinetics. The same scheme (i.e., sign convention on `eostype`) is used for CMF1.

Table 4 Input constants 1–32 for CMF2

Number	Name	Type	Symbol	Units	Definition
1	`lambda`	F	λ_0	P	Lamé modulus $\lambda_0 = \mathsf{B}_0 - \frac{2}{3}\mathsf{G}_0$
2	`mu0`	F	G_0	P	undamaged reference shear modulus
3	`theta0`	F	θ_0	K	reference temperature (not necessarily initial θ)
4	`tempmelt`	F	θ_M	K	melting temperature
5	`hsltype`	I	-	-	set to 3 for plastic potential of CMF1 and CMF2
6	`hsl1`	F	$\sigma_0^{(0)}$	P	initial yield strength of parent phase
7	`hsl2`	F	$\Theta_0^{(0)}$	P	affects strain hardening of parent phase
8	`hsl3`	F	$\delta^{(0)}$	-	affects strain hardening of parent phase
9	`hsl4`	F	$\epsilon^{(0)}$	-	affects strain hardening of parent phase
10	`hsl5`	F	$\dot{e}_0$	t^{-1}	reference normalization strain rate
11	`hsl6`	F	$m^{(\alpha)}$	-	strain rate sensitivity; same for all phases
12	`hsl7`	F	$p^{(\alpha)}$	-	thermal softening; same for all phases
13	`hsl8`	F	$\dot{e}_{\min}$	t^{-1}	minimum allowed value of $\dot{e}^P$ in Eq. 28
14	`dmgtype`	I	-	-	damage model type*
15	`d1`	F	$\hat{m}$	-	damage growth parameter
16	`d2`	F	c_0	-	damage nucleation parameter
17	`d3`	F	-	-	not used; set to 0.0
18	`d4`	F	-	-	not used; set to 0.0
19	`d5`	F	-	-	not used; set to 0.0
20	`d6`	F	-	-	not used; set to 0.0
21	`d7`	F	-	-	not used; set to 0.0
22	`d8`	F	ϕ_c	-	max local damage at failure*
23	`d9`	F	ϕ_m		enforces* $c^\phi = \phi_m/\phi$ for $\phi \geq \phi_m$
24	`d10`	F	$p_{\min}$	P	min tensile pressure supported if $\phi \geq \phi_c$
25	`eostype`	I	-	-	set to 2 for EOS of Eq. 64; -2 for $\dot{\theta} = 0$
26	`cv0`	F	c_V	$\mathrm{P{\cdot}K}^{-1}$	constant specific heat per unit volume
27	`b0`	F	B_0	P	undamaged reference bulk modulus
28	`dbdp`	F	B_0'	-	pressure derivative of bulk modulus
29	`cte`	F	A	K^{-1}	vol. thermal expansion; $\gamma_0 = A\mathsf{B}_0/c_V$
30	`gruna`	F	-	-	not used; set to 0.0
31	`ecolds`	F	-	-	not used; set to 0.0
32	`ecold0`	F	-	-	not used; set to 0.0

* see text

Table 5 Input constants 33–64 for CMF2

Number	Name	Type	Symbol	Units	Definition
33	ecold1	F	-	-	not used; set to 0.0
34	ecold2	F	-	-	not used; set to 0.0
35	beta	F	$\hat{\beta}$	-	constant Taylor-Quinney factor
36	objrate	I	-	-	objective rate formulation type*
37	vf0	F	ξ_0	-	initial fraction of second (product) phase
38	pttype	I	-	-	set to 2 for $\dot{\xi}$ of Eq. 78; set to 0 for $\dot{\xi} = 0$
39	pt1	F	δ^{ξ}	-	transformation volume change
40	pt2	F	γ_0^{ξ}	-	tranformation shear strain
41	pt3	F	$\gamma^{\xi}/\gamma_0^{\xi}$	-	1.0 for CMF2; redundant scaling for γ^{ξ}
42	pt4	F	-	-	not used; set to 0.0
43	pt5	F	ψ_0	P	free energy offset of product phase
44	pt6	F	R	-	transition rate scaling exponent
45	pt7	F	$\dot{\xi}_0$	t^{-1}	$\dot{\xi}_m = \dot{\xi}_0$ if $\gamma^{\xi} = 0$ and/or $\dot{e}^P = 0$
46	pt8	F	λ_T	P	latent heat parameter
47	pt9	F	θ_T	K	transition temperature
48	pt10	F	ι	-	transition strength accommodation
49	pt11	F	$\sigma_0^{(1)}$	P	initial yield strength of product
50	pt12	F	$\Theta_0^{(1)}$	P	affects strain hardening of product
51	pt13	F	$\delta^{(1)}$	-	affects strain hardening of product
52	pt14	F	$\epsilon^{(1)}$	-	affects strain hardening of product
53	pt15	F	$\dot{e}_1$	t^{-1}	normalizes transition rate scaling
54	pt16	F	μ_{ξ}	-	dimensionless transition viscosity
55	hsl9	F	k_s	-	strength amplification, viscous drag
56	hsl10	F	$\dot{e}_s$	t^{-1}	normalizing strain rate, viscous drag
57	mskip	I	-	-	set to 0 for CMF2
58	stressfac	F	-	-	not used in CMF2; Pa/P to SI unit*
59	econd	F	Σ	$10^6 A(Vm)^{-1}$	electrical conductivity
60	tcond	F	κ	$W(Km)^{-1}$	thermal conductivity
61	em1	F	$e_{\|\|}^{M(0)}$	10^{-6}	not used; set to 0.0 for CMF2
62	em2	F	$e_{\|\|}^{M(1)}$	10^{-6}	not used; set to 0.0 for CMF2
63	hsat1	F	$\mu_0 H^{S(0)}$	T	not used; set to 0.0 for CMF2
64	hsat2	F	$\mu_0 H^{S(1)}$	T	not used; set to 0.0 for CMF2

* see text

Table 6 Input constants 65–96 for CMF2

Number	Name	Type	Symbol	Units	Definition
65	msat1	F	$\mu_0 M^{S(0)}$	T	not used; set to 0.0 for CMF2
66	msat2	F	$\mu_0 M^{S(1)}$	T	not used; set to 0.0 for CMF2
67	temp0	F	θ_i	K	initial temperature $\theta(t=0)$
68	pt17	F	$\Delta^* A$	K^{-1}	not used; set to 0.0 for CMF2
69	pt18	F	α_1	$P{\cdot}T^{-1}$	not used; set to 0.0 for CMF2
70	pt19	F	β_1	$P{\cdot}T^{-1}$	not used; set to 0.0 for CMF2
71	pt20	F	ι_1	T^{-1}	not used; set to 0.0 for CMF2
72	magmod	I	-	-	not used; set to 0 for CMF2
73	ma1	F	$m_0^{(0)}$	T	not used; set to 0.0 for CMF2
74	mb1	F	$m_1^{(0)}$	-	not used; set to 0.0 for CMF2
75	mc1	F	$m_2^{(0)}$	T^{-1}	not used; set to 0.0 for CMF2
76	ma2	F	$m_0^{(1)}$	T	not used; set to 0.0 for CMF2
77	mb2	F	$m_1^{(1)}$	-	lnot used; set to 0.0 for CMF2
78	mc2	F	$m_2^{(1)}$	T^{-1}	not used; set to 0.0 for CMF2
79	ga1	F	$\Phi_0^{(0)}$	P	not used; set to 0.0 for CMF2
80	gb1	F	$\Phi_1^{(0)}$	$P{\cdot}T^{-1}$	not used; set to 0.0 for CMF2
81	gc1	F	$\Phi_2^{(0)}$	$P{\cdot}T^{-2}$	not used; set to 0.0 for CMF2
82	ga2	F	$\Phi_0^{(1)}$	P	not used; set to 0.0 for CMF2
83	gb2	F	$\Phi_1^{(1)}$	$P{\cdot}T^{-1}$	not used; set to 0.0 for CMF2
84	gc2	F	$\Phi_2^{(1)}$	$P{\cdot}T^{-2}$	not used; set to 0.0 for CMF2
85	dga1	F	$\Phi_0'^{(0)}$	$P{\cdot}K^{-1}$	not used; set to 0.0 for CMF2
86	dgb1	F	$\Phi_1'^{(0)}$	$P{\cdot}K^{-1}{\cdot}T^{-1}$	not used; set to 0.0 for CMF2
87	dgc1	F	$\Phi_2'^{(0)}$	$P{\cdot}K^{-1}{\cdot}T^{-2}$	not used; set to 0.0 for CMF2
88	dga2	F	$\Phi_0'^{(1)}$	$P{\cdot}K^{-1}$	not used; set to 0.0 for CMF2
89	dgb2	F	$\Phi_1'^{(1)}$	$P{\cdot}K^{-1}{\cdot}T^{-1}$	not used; set to 0.0 for CMF2
90	dgc2	F	$\Phi_2'^{(1)}$	$P{\cdot}K^{-1}{\cdot}T^{-2}$	not used; set to 0.0 for CMF2
91	cva1	F	$c_{V0}^{(0)}$	$P{\cdot}K^{-1}$	not used; set to 0.0 for CMF2
92	cvb1	F	$c_{V1}^{(0)}$	$P{\cdot}K^{-1}{\cdot}T^{-1}$	lnot used; set to 0.0 for CMF2
93	cvc1	F	$c_{V2}^{(0)}$	$P{\cdot}K^{-1}{\cdot}T^{-2}$	not used; set to 0.0 for CMF2
94	cva2	F	$c_{V0}^{(1)}$	$P{\cdot}K^{-1}$	not used; set to 0.0 for CMF2
95	cvb2	F	$c_{V1}^{(1)}$	$P{\cdot}K^{-1}{\cdot}T^{-1}$	not used; set to 0.0 for CMF2
96	cvc2	F	$c_{V2}^{(1)}$	$P{\cdot}K^{-1}{\cdot}T^{-2}$	not used; set to 0.0 for CMF2

* see text

Though generally untested in this regard, the present software implementation does permit mixing of features among those of CMF1 and CMF2. For example, the EOS of CMF1 (`eostype` ± 3) could be used with the phase transition theory of CMF2 (`pttype` 2). Mixing of choices of `dmgtype` and `objrate` among those recommended in the two frameworks is also possible, but such non-standard mixing may not be theoretically consistent.

Furthermore, `mskip` can be set to 1, with `magmod` set to 0 or 1, to enable magnetism irrespective of the other model options. When magnetism is enabled, constants 68–96 should be chosen according to their physical definitions in Table 3. The user is cautioned that all possible combinations of features have not undergone verification testing, and some combinations might not be thermodynamically consistent even if apparently reasonable numerical results are obtained. Thus, combinations of features that are deviations from CMF1 and CMF2 should be undertaken at the user's own risk. In these cases, the user should consult the source code for a full understanding of the underlying calculations.

Those constants that require further explanation, as marked by an asterisk in the sixth column, are discussed next:

- `dmgtype`: set to 4 for damage model of CMF2 and Section 2.2.4 with $c^\phi = 0$ (i.e., no dilatation, brittle damage); set to 5 for damage model of CMF2 and Section 2.2.4 with $c^\phi = 1$ (i.e., dilatation, ductile damage); set to 0 to disable damage model (i.e., $\phi = 0$ and $\dot{\phi} = 0$)
- `d8`$= \phi_c$: sets $\phi(t^+) \rightarrow 1$ when $\phi(t) \geq \phi_c$ to rapidly fail material
- `d9`$= \phi_m$: constrains maximum dilatation from voids to $c^\phi \phi_m$ to avoid singularity in J^ϕ as $\phi \rightarrow 1$
- `objrate`: set to 3 to invoke full version of Eq. 80; set to 2 to invoke Eq. 80 with $\boldsymbol{\omega} = \mathbf{0}$; set to 1 to invoke classic Jaumann rate (i.e., Eq. 80 with right term $[\cdot]\bar{\boldsymbol{\sigma}}$ omitted); set to 0 to deactivate objective rate (i.e., $\overset{\nabla}{\bar{\boldsymbol{\sigma}}} \rightarrow \dot{\bar{\boldsymbol{\sigma}}}$ in Eq. 80)
- `mskip`: set to 0 to disable magnetization, magnetic energy, and magnetic contribution to specific heat consistently with CMF2; enables magnetization models of CMF1 if equal to 1

- `stressfac`: converts base SI units to arbitrary pressure units used in mechanics calculations (e.g., if P is in MPa, `stressfac` $= 10^{-6}$); multiply quantity given in base SI units of Pa by `stressfac` to obtain quantity in P units; not used in the code if `mskip` is set to 0

3.3 Syntax for Input Files

Syntax for an `ABAQUS`-style UMAT input file is given below. The syntax may differ for other host codes. Here, `cmname` is an arbitrary 8-character identifier. Denoted by `nstatv` is an integer specifying the number of state variables, to be elaborated in Section 4. For the present models, `nstatv` should be set to 27. Denoted by `nprops` is the integer number of input constants whose value should be set to 96 for the present models. These input constants, defined in Section 3.1 and Section 3.2 and Tables1–6 are entered in a comma-separated fashion, 8 per line. The eight parameter of each line is not followed by a comma. Free spaces are optional between commas and input values when the MPS of Section 6 is used in lieu of a host code.

*MATERIAL, NAME = `cmname`
*DEPVAR
`nstatv` (27)
*USER MATERIAL, CONSTANTS = `nprops` (96)
`lambda, mu0, theta0, tempmelt, hsltype, hsl1, hsl2, hsl3`
`hsl4, hsl5, hsl6, hsl7, hsl8, dmgtype, d1, d2`
`d3, d4, d5, d6, d7, d8, d9, d10`
`eostype, cv0, b0, dbdp, cte, gruna, ecolds, ecold0`
`ecold1, ecold2, beta, objrate, vf0, pttype, pt1, pt2`
`pt3, pt4, pt5, pt6, pt7, pt8, pt9, pt10`
`pt11, pt12, pt13, pt14, pt15, pt16, hsl9, hsl10`
`mskip, stressfac, econd, tcond, em1, em2, hsat1, hsat2`
`msat1, msat2, temp0, pt17, pt18, pt19, pt20, magmod`
`ma1, mb1, mc1, ma2, mb2, mc2, ga1, gb1`
`gc1, ga2, gb2, gc2, dga1, dgb1, dgc1, dga2`
`dgb2, dgc2, cva1, cvb1, cvc1, cva2, cvb2, cvc2`

4. State Variables

User-defined state variables CMF1 are listed in Table 7. The same state variables are defined for both frameworks, that is, CMF1 and CMF2. However, some are dummy variables for the standard implementation CMF2, since magnetic effects are omitted by the default input file selections for CMF2, as discussed in Section 3.2.

Other standard state variables (i.e., those that are not purely user-defined) are also passed in and out of the UMAT subroutine. These standard state variables include the deformation gradient $\boldsymbol{F}$ at the beginning (`dfgrd0`) and end (`dfgrd1`) of the time step and a vector of six components (`stress`) of the symmetric Cauchy stress tensor $\boldsymbol{\sigma}$ that are updated within the UMAT. The solution time at the beginning of the step (t, `time`) and the time step (dt, `dtime`) are also standard input. The user is referred to the documentation of the host code (e.g., `ABAQUS`) for more information on standard input/output variables.

The layout of Table 7 is similar to that of Tables 1–6. State variables are numbered from 1 to `nstatv`, the latter with a fixed value of 27 in the current software implementation. The first (i.e., leftmost) column numbers each user-defined state variable according to its sequence in the `statev` array that is passed in/out of the UMAT subroutine. The second column labels each variable by its name used in the FORTRAN source code of the UMAT subroutine. The third column gives the data type: floating point number (F) or integer (I). The fourth column gives the mathematical symbol used in the theoretical formulation of Section 2.1, Section 2.2, and/or the original references.[10–12] The time value t at the beginning of the time step of size dt corresponds to the input value of any state variable. If the state variable is updated within the UMAT, its output value passed to the host code corresponds to that at the end time of the step, that is, at $t + \mathrm{d}t$. The fifth column gives the units. The sixth column gives the initial condition (IC) if prescribed by the UMAT itself. Those variables without an IC listed in this column are initialized by the host code or the MPS. The seventh (i.e., rightmost) column gives the definition of the state variable.

Table 7 User-defined state variables for CMF1 and CMF2

Number	Name	Type	Symbol	Units	IC	Definition
1	epsl0	F	$e^P(t)$	-	0.0	cumulative plastic strain
2	epdsl0	F	$\dot{e}^P(t)$	t^{-1}	0.0	deviatoric plastic strain rate
3	energy	F	$U(t)$	P	$\hat{c}_{V0}\theta_i$	internal energy relative to 0 K
4	tempsv	F	$\theta(t)$	K	θ_i	temperature
5	dmg	F	$\phi(t)$	-	0.0	damage, continuous $\in [0,1]$
6	dmgflag	F	-	-	0.0	0.0 (not failed) or 1.0 (failed)*
7	dmgdot	F	$\dot{\phi}(t)$	t^{-1}	0.0	damage rate
8	mu	F	$\hat{\mathsf{G}}(t)$	P	G_0	tangent shear modulus, obj. rate
9	vf	F	$\xi(t)-\xi_0$	-	0.0	phase fraction minus initial frac.
10	vfdot	F	$\dot{\xi}(t)$	t^{-1}	0.0	phase transition rate
11	vfq	F	-	-	0.0	identifies strain-assist transition*
12	tebar(1)	F	$\bar{t}^e_{11}(t)$	P	0.0	deviatoric elastic stress 11-comp.
13	tebar(2)	F	$\bar{t}^e_{22}(t)$	P	0.0	deviatoric elastic stress 22-comp.
14	tebar(3)	F	$\bar{t}^e_{33}(t)$	P	0.0	deviatoric elastic stress 33-comp.
15	tebar(4)	F	$\bar{t}^e_{12}(t)$	P	0.0	deviatoric elastic stress 12-comp.
16	tebar(5)	F	$\bar{t}^e_{13}(t)$	P	0.0	deviatoric elastic stress 13-comp.
17	tebar(6)	F	$\bar{t}^e_{23}(t)$	P	0.0	deviatoric elastic stress 23-comp.
18	bmag	F	$B(t)$	T	**	signed magnitude magnetic flux*
19	xmag	F	$M(t)$	$\mathrm{P{\cdot}T}^{-1}$	**	signed magnitude magnetic field*
20	bdir(1)	F	$i_1(t)$	-	**	magnetic field direction 1-comp.
21	bdir(2)	F	$i_2(t)$	-	**	magnetic field direction 2-comp.
22	bdir(3)	F	$i_3(t)$	-	**	magnetic field direction 3-comp.
23	cv	F	$c_V(t)$	P	**	total specific heat $c_V(\xi,\theta,\lvert H\rvert)$
24	delcv	F	$\Delta^* c_V(t)$	P	**	$c_V^{(1)}(\theta,\lvert H\rvert)-c_V^{(0)}(\theta,\lvert H\rvert)$
25	delphi	F	$\Delta^*\Phi(t)$	P	**	$\Phi^{(1)}(\theta,\lvert H\rvert)-\Phi^{(0)}(\theta,\lvert H\rvert)$
26	ddelphi	F	$\Delta^*\Phi'(t)$	P	**	$\Phi'^{(1)}(\theta,\lvert H\rvert)-\Phi'^{(0)}(\theta,\lvert H\rvert)$
27	dtempcond	F	$\Delta\theta_{\mathrm{c}}(t)$	K	**	$\Delta\theta_{\mathrm{c}}=J[\Sigma\lvert\mathbf{E}\rvert^2+\kappa\nabla^2\theta]\mathrm{d}t/c_V$*

* see text

** host code

Those variables that require further explanation, as marked by a single asterisk in the seventh column, are discussed next:

- `dmgflag`: takes value of 0.0 or 1.0; becomes 1.0 when $\phi \geq \phi_c$; when `dmgflag` equals 1.0, material has no strength ($\bar{\sigma}^V \rightarrow 0$), becomes non-magnetic ($M \rightarrow 0$), and supports little tensile pressure ($p \geq -p_{\text{min}}$); failed material can support unlimited compressive pressure $p > 0$ and can conduct heat and electricity

- `vfq`: initial value of 0.0 for no transition, then returns 1.0 if transition occurs with $e^P(t) = 0$ (e.g., a thermally driven or pressure assisted transition), or returns 2.0 if transition occurs with $e^P(t) > 0$ (i.e., plastic strain before transition initiates, a strain-assisted transformation)

- `bmag`: spatial magnetic flux density vector is $\boldsymbol{B} = B\boldsymbol{i}$; recall B can be zero, positive, or negative; input variable to the UMAT supplied by host code or MPS

- `xmag`: spatial magnetization vector is $\boldsymbol{M} = M\boldsymbol{i}$; recall M can be zero, positive, or negative; SI units are scaled such that $B \cdot M$ is stress unit (P); input variable to the UMAT supplied by host code or MPS

- `dtempcond`: temperature increment from electrical and thermal conduction; input variable to the UMAT supplied by host code or MPS

The custom subroutine referred to as the MAGUMAT has been implemented to update electromagnetic and conductive variables outside the magnetohydrodynamic routines of the host code. The MAGUMAT accesses the same `props` and `statev` arrays that are implemented in the mechanical UMAT subroutine. Additional input/output variables for the UMAT routine are associated with the electromagnetic and conductive loading conditions prescribed by the MPS. These variables are discussed in Section 6 in the context of documentation for the MPS.

5. Code Maturity

In addition to verification examples presented in Section 7, the two model frameworks have been implemented in predictive numerical simulations in prior published works.[10–12] Table 8 lists, for each framework, the ferrous materials, loading conditions, and source references of these prior studies. Further remarks follow for CMF1 in Section 5.1 and for CMF2 in Section 5.2. In fact, the verification examples of Section 7 are a subset of cases listed in Table 8.

As will be clear from subsequent discussion, CMF2 is considered more mature since it has been used to solve more problems and has been implemented in a large-scale host code. However, CMF1 is more theoretically robust since it improves on the theory of CMF2 to include electromagnetic effects and reversible phase transitions, physics that are omitted by CMF2.

Table 8 Successful prior research implementations of CMF1 and CMF2

Model framework	Material	Mechanical -thermal loading	Magnetic loading	Reference
CMF1	pure Fe	hydrostatic	constant $\boldsymbol{H}$	Clayton et al.[12]
		adiabatic uniaxial strain	constant $\boldsymbol{B}$	
		adiabatic uniaxial strain	constant $\boldsymbol{H}$	
	austentic steels (2)	isothermal extension	constant $\boldsymbol{H}$	
CMF2	Mn steels (3)	isothermal extension	none	Clayton and Lloyd[10]
		isothermal compression	none	
		adiabatic extension	none	
		adiabatic compression	none	
		adiabatic uniaxial strain	none	
		dynamic simple shear	none	
		shear + tension/compress	none	
		Taylor cylinder impact	none	Limmer et al.[3]

5.1 CMF1

The present numerical framework for CMF1 originated with the publication of Clayton et al.[12] That work contained simulations of the high-pressure $\alpha \leftrightarrow \epsilon$ transition in pure Fe for verification of numerical methods with analytical solutions[11] and for validation with experimental data.[4,57–61] Cases considered for pure Fe were hydrostatic compression with and without constant applied magnetic fields, and

adiabatic uniaxial strain compression with and without applied constant magnetic fields or fluxes. The latter sought to replicate planar shock impact experiments.[4,59]

The work of Clayton et al.[12] also contained simulations of quasi-static, uniaxial stress tensile loading of two austenitic steels of the same chemical composition but different heat treatments. Magnetic loading conditions, if nonzero, considered applied constant $\boldsymbol{H}$ orthogonal to the direction of mechanical extension. Those isothermal simulations sought to replicate experiments performed at ARL first documented in the same publication.[12] The phase transitions in these alloys primarily involved $\gamma/\epsilon \rightarrow \alpha$ for tensile loading. The γ and ϵ phases were homogenized into a single close-packed phase in simulations since CMF1 can explicitly address only two phases simultaneously.

All prior and current numerical simulations that invoke CMF1 rely on the MPS (see Section 6) to call both the UMAT and MAGUMAT subroutines. All of these simulations are thus restricted to a single material point (i.e., a single integration point or a single 3-D hexahedral finite element with reduced integration). Mechanical and electromagnetic fields are necessarily constant over the homogenized domain, though magnetization and local deviatoric stress may differ among coexisting phases. The MAGUMAT subroutine has not, as of the date of this report, been implemented in a host code capable of modeling spatially heterogeneous response (i.e., capable of representing multiple finite elements).

5.2 CMF2

The present numerical framework for CMF2 originated with the publication of Clayton and Lloyd.[10] That work contained numerous simulations of the coupled mechanical and thermal response of three different medium-high manganese (Mn) steels with different SFEs. The alloy with highest SFE was a SLIP steel, that with moderate SFE a TWIP steel, and that with lowest SFE a TRIP steel. Phase transitions were only significant in the latter (TRIP) material, dominated by, and modeled as, $\gamma \rightarrow \alpha$ transformations.

Complementary experiments for many, but not all, of the simulated loading protocols are reported in that work and four others.[2,62–64] Simulations that were either calibrated or validated versus experimental data addressed static and dynamic tension and static and dynamic compression, under room-temperature and high-

temperature conditions. Adiabatic uniaxial strain compression was also considered, primarily for validation of the EOS versus experimental shock compression data on other steels.[65] Predictive simulations also considered dynamic adiabatic simple shear, with and without superposed tensile or compressive pressure.

Simulations and results of Clayton and Lloyd[10] were all obtained with the MPS, which called a previous generation of the UMAT and did not require the MAGUMAT since explicit magnetic effects were not modeled. More recent simulations reported by Limmer et al.[3] modeled Taylor impact experiments performed at ARL on the same three steels (SLIP, TWIP, and TRIP). These 3-D dynamic simulations invoked the same generation of UMAT subroutine installed in the `ALE3D` host code.[19] These simulations predicted the deformation and failure of a cylindrical projectile striking a flat surface, which necessitated numerous finite elements to resolve the geometry and heterogenous response. Qualitative agreement, with experiments, in trends of deformed shapes of the cylinders was predicted by this implementation of CMF2. The parameters for Taylor impact simulations of all three steels were obtained from the prior study[10] and were not tuned to match the observations.

6. Standalone Simulator (MPS)

The MPS is used in lieu of a large-scale FE code. Simulations using the MPS are restricted to a single continuum point, with spatially homogeneous boundary conditions. The current MPS imposes mechanical and/or magnetic loading on the continuum point, both input by the user. The magnitude of electric field and the Laplacian of temperature are also imposed on the continuum point as specified by the user. Analogous loading conditions would be imposed by the larger-scale host code on an element-wise basis. For each time increment within the user-specified load history, the MPS calls the UMAT to obtain the mechanical response and the MAGUMAT to obtain the electromagnetic response. Results are output to two different files throughout the loading history.

An overview of procedures entering the MPS, including input/output of the UMAT and MAGUMAT, is given in Section 6.1. Input file syntax is described in Section 6.2. Output files are discussed in Section 6.3. The source code for the MPS is in Appendix B, a FORTRAN file called `mps.f`. This file, in turn, accesses the UMAT source, called `umat.f`, and the MAGUMAT source, called `magumat.f`. Source files `umat.f` and `magumat.f` are not included in the current report.

6.1 Procedures

The workflow of the MPS follows the general sequence below:

1. Initialize time, deformation, temperature; set null electromagnetic variables
2. Open input file `sim.inp` and output files `sim.txt` and `out.txt`
3. Read material properties and loading conditions from input file
4. Call MAGUMAT to update electromagnetic state at null deformation
5. Call UMAT to obtain initial tangent stiffness
6. Write out initial state with imposed electromagnetic field but null deformation
7. Increment one time step; update applied loads
8. Call MAGUMAT to update magnetism and conduction
9. Call UMAT to update stress and state variables (energy, temperature, etc.)
10. Write incremental results to `sim.txt`, and `out.txt`
11. Return to step 7 if load history incomplete; otherwise close files and end

Depending on the loading protocols, the MPS performs time step cutbacks (i.e., reduction in dt) to achieve convergence consistent with the boundary conditions (e.g., uniaxial stress). Time step increases can be enabled when convergence is achieved with a single iteration using the current step size dt. These details are included in the software but were omitted from the documentation of the workflow.

The primary variables that enter and exit the UMAT and MAGUMAT subroutines are listed in Table 9. A one-to-one correspondence between input and output quantities on the same row of this table does not generally apply; the exception is correspondence of magnetic field and flux density for the MAGUMAT. If $\boldsymbol{B}$ is imposed as the input loading condition, then $\boldsymbol{H}$ is calculated and output by the MAGUMAT according to assumptions in Section 2.1.5, and vice versa (i.e., if $\boldsymbol{H}$ is imposed, then $\boldsymbol{B}$ is calculated and output).

Table 9 Primary input and output variables for UMAT and MAGUMAT subroutines

Routine	Input (name; symbol)	Output (name; symbol)
UMAT	material properties (`props`)	Cauchy stress (`stress`; $\boldsymbol{\sigma}$)
	deformation gradient (`dfgrd0,1`; $\boldsymbol{F}$)	temperature* (θ)
	magnetization and magnetic flux* (M, B)	plastic strain* (e^P)
	time and time step (`time,dtime`; t, dt)	phase fraction* (ξ)
		damage* (ϕ)
MAGUMAT	material properties (`props`)	magnetization ($\boldsymbol{M}$)
	magnetic field OR flux ($\boldsymbol{H}$ OR $\boldsymbol{B}$)	magnetic flux OR field ($\boldsymbol{B}$ OR $\boldsymbol{H}$)
	phase fraction* (ξ)	total specific heat* (c_V)
	temperature* (θ)	magnetic energy difference* ($\Delta^*\Phi$)
	temperature Laplacian ($\nabla^2\theta$)	mag-thermal energy difference* ($\Delta^*\Phi'$)
	electric field magnitude (E)	specific heat difference* ($\Delta^* c_V$)
	time and time step (`time,dtime`; t, dt)	conduction temperature rise* ($\Delta\theta_c$)

*`statev` array

6.2 Syntax for Input File with Loading Conditions

The input file read by the MPS is named `sim.inp`. The first 16 lines of this file are identical in format to the input file syntax for state variable designation and user material constants described in Section 3.3. To these lines are appended the electromagnetic, thermal, and mechanical boundary conditions and the protocols for time step incrementation. The present implementation only admits one loading step for electromagnetic and thermal conditions but multiple steps for mechanical conditions, the latter consistent with capabilities of the `ABAQUS` software (though this, or any other, host code is not used simultaneously with the MPS). A "loading step" of duration $t_1 - t_0$, where t_0 and t_1 are start and end times of the step, should not be confused with a local "time step" of typically much smaller duration dt. The syntax is given below for an input file prescribing a single loading step:

*MATERIAL, NAME = `cmname`
*DEPVAR
`nstatv` (27)
*USER MATERIAL, CONSTANTS = `nprops` (96)
`lambda, mu0, theta0, tempmelt, hsltype, hsl1, hsl2, hsl3`
... (properties 9 through 88) ...
`dgb2, dgc2, cva1, cvb1, cvc1, cva2, cvb2, cvc2`

```
mloading_mode, fmag, frate, bdir(1), bdir(2), bdir(3), efield, dtempdx, d2tempdx2
```
*STEP
```
dtime, time_max, dtime_max
loading_mode[i][j], velocity, velocity2
```

Line 17 (small font size immediately above) defines the electromagnetic and thermal loading:

- `mloading_mode`: a single character; to be input as H for applied magnetic field $\boldsymbol{H}$ or input as B for applied magnetic flux density $\boldsymbol{B}$
- `fmag`: constant applied field magnitude $\mu_0 H_0$ or B_0 (units of T)
- `frate`: linear applied field rate $\mu_0 \dot{H}_0$ or $\dot{B}_0$ (units of T·t^{-1})
- `bdir(1),bdir(2),bdir(3)`: i_1, i_2, i_3 components of fixed direction $\boldsymbol{i}$ for applied $\boldsymbol{H}$ or $\boldsymbol{B}$
- `efield`: constant applied electric field magnitude $\mathsf{E} = |\mathbf{E}|$ (units V·m^{-1})
- `dtempdx`: constant temperature gradient magnitude $|\nabla\theta|$ (units K·m^{-1})
- `d2tempdx2`: constant temperature Laplacian $\nabla^2\theta$ (units K·m^{-2})

For `mloading_mode` H, the local applied magnetic field is, in units of Tesla (T),

$$\mu_0 \boldsymbol{H}(t) = [\mu_0 H_0 + \mu_0 \dot{H}_0 t]\boldsymbol{i}. \tag{83}$$

For `mloading_mode` B, the local magnetic flux density is, in units of T,

$$\boldsymbol{B}(t) = [B_0 + \dot{B}_0 t]\boldsymbol{i}. \tag{84}$$

Line 19 defines the time integration settings for the current loading step:

- `dtime`: initial time step dt (units of t)
- `time_max`: end time of the loading step (e.g., t_1) (units of t)
- `dtime_max`: maximum size of dt permitted by convergence relaxation (units of t); minimum dt for convergence tightening hardcoded at 10^{-15} (units of t)

Time t is initialized to $t_0 = 0$ for the first loading step. If multiple loading steps are prescribed, lines 18 through 20 are repeated for each additional step, and t_0 for the next step is set to t_1 of the previous step. An example with two steps is provided in Section 7.1. Line 20 defines the mechanical load history for the load step:

- `loading_mode`: single character; input as C for uniaxial-stress compression or tension, input as T for torsion, input as P for plane strain, input as S for simultaneous compression or tension with torsion, input as B for biaxial tension or compression, input as U for uniaxial strain compression or tension, input as V for volumetric (i.e., spherical) compression or expansion, input as Z for simple shear with possible spherical compression or expansion
- `[i][j]`: each a single integer, written as i or j in what follows, specifying the loading direction
- `velocity,velocity2`: each a floating point number, written as v_1, v_2 in what follows, specifying the loading rate; `velocity2` is not used for all loading modes (units of t^{-1}, no length scale)

For modes C, P, S, B, U, V, and Z, the corresponding v_1 or v_2 should be positive in sign for tensile loading and negative in sign for compressive loading. Table 10 lists the form of the *applied* deformation gradient components $\boldsymbol{F}$ and three quantities that are written to the `out.txt` file described in Section 6.3: output strain measure ϵ, output normal stress measure P (units of P, possible scaled), and output shear stress measure τ (units of P, possibly scaled). Output strain measures are hardcoded for particular loading directions, and certain output stresses are scaled by 10^{-3} for convenience; see the source code for `mps.f` in Appendix B. For loading mode C, the axial component of the total nominal stress $\hat{\boldsymbol{T}} = J\boldsymbol{T}\boldsymbol{F}^{-\mathrm{T}}$ is used for output. The total stress tensor[11,12,23] is $\boldsymbol{T}$ includes Cauchy and Maxwell stress contributions. Effective total strain $e = (\frac{2}{3}\int |\boldsymbol{d} : \boldsymbol{d}|\mathrm{d}t)^{1/2}$ is used for ϵ in several protocols that await further verification (i.e., P, S, and B) , with p for P and zero for τ. The source code can be easily edited to supply definitions of the user's convenience in these cases. Loading conditions follow conventional mechanics definitions. For example, orthogonal components of displacement to the loading direction are null for uniaxial strain but are generally nonzero for uniaxial stress.

Table 10 Mechanical loading options and hardcoded strain ϵ and stresses P, τ for the MPS

Mode	Direction	$\boldsymbol{F}$	ϵ	P	τ	Remarks
C	$i = j$	$F_{ij} = 1 + v_1 t$	$\lvert F_{33} - 1 \rvert$	$\lvert \hat{T}_{33} \rvert$	$\frac{1}{2} \lvert \hat{T}_{33} \rvert$	compression/tension along $i = j$
T	$i \neq j$	$F_{ij} = v_1 t$	F_{12}	p	σ_{12}	shearing in i direction on i-j plane
P	$i \neq j$	$F^*_{jj} = 1 + v_1 t$	e	p	0	plane strain along j in i-j plane
S	$i \neq j$	$F^*_{jj} = 1 + v_1 t$ $F_{ij} = v_2 t$	e	p	0	compression and torsion in j-k plane
B	$i \neq j$	$F^*_{ii} = 1 + v_1 t$ $F^*_{jj} = 1 + v_2 t$	e	p	0	biaxial loading in i-j plane
U	$i = j$	$F_{ij} = 1 + v_1 t$	$1 - F_{33}$	$-\sigma_{33}$	$\lvert \frac{1}{2}(\sigma_{11} - \sigma_{33}) \rvert$	uniaxial strain along $i = j$
V	$i = j$	$F_{kk} = 3 + v_1 t$	$1 - \det \boldsymbol{F}$	p	$\lvert \frac{1}{2}(\sigma_{11} - \sigma_{33}) \rvert$	spherical loading along $i = j = k$
Z	$i \neq j$	$F_{kk} = 3 + v_1 t$ $F_{ij} = v_2 t$	F_{12}	p	σ_{12}	shear in i-j plane + spherical loading

* no sum

6.3 Output Files

The MPS writes output to two different text files: `sim.txt` and `out.txt`. The `sim.txt` file duplicates the output that is simultaneously printed to the terminal. Quantities output to the `sim.txt` file for the initial conditions and for every time increment spanned by dt thereafter are the following:

- Load step number (integer) and total time t (units of t)
- Effective total strain $e = (\frac{2}{3}\int |\boldsymbol{d} : \boldsymbol{d}|\mathrm{d}t)^{1/2}$
- All six independent components of symmetric Cauchy stress $\boldsymbol{\sigma}$ (units of P)
- Six components of deformation gradient: $F_{11}, F_{22}, F_{33}, F_{12}, F_{23}, F_{13}$
- Effective deviatoric plastic strain e^P and its rate $\dot{e}^P$ (units of t^{-1})

Quantities written to the `out.txt` file for the initial conditions and for every time increment spanned by dt thereafter are the following:

- Load step number (integer) and total time t (units of t)
- Strain measure ϵ as defined per mechanical loading mode in Section 6.2
- Determinant of deformation gradient $J = \det \boldsymbol{F}$
- Normal stress measure P as defined per loading mode in Section 6.2
- Shear stress measure τ as defined per loading mode in Section 6.2
- Temperature θ (units of K)
- Damage ϕ
- Damage flag `dmgflag`; see Section 4
- Damage rate $\dot{\phi}$ (units of t^{-1})
- Tangent shear modulus $\hat{\mathsf{G}}$ (units of P); see Section 4
- Change in phase volume fraction $\Delta\xi = \xi - \xi_0$
- Phase transition rate $\dot{\xi}$

- Strain-assisted transition indicator `vfq`; see Section 4
- Magnetic flux density, signed magnitude B (units of T)
- Magnetization, signed magnitude $\mu_0 M$ (units of T)
- Magnetic energy difference $\Delta^* \Phi$ (units of P)

7. Verification Examples

Two classes of verification problem for each CMF are included. The physical problems and results are discussed in Sections 7.1 and 7.2. Corresponding input files for the MPS are contained in Appendix A in respective Sections A.1 and A.2.

7.1 CMF1

The first problem for CMF1 considers a sample of pure Fe subjected to hydrostatic compression followed by hydrostatic decompression. This problem exercises the logarithmic EOS, the phase transformation model, and the magnetization model with saturation option (`magmod` of 0). Plasticity and damage are absent for hydrostatic compression. Isothermal conditions are assumed ($\theta = 300$ K), with no thermal or electrical conduction. The imposed magnetic field $\mu_0 H$ is time-independent, of strength 0, 25, or 50 T. The loading is of spherical compression to a volume reduction of around 15%, leading to a maximum Cauchy pressure of 25.5 GPa. The input file to the MPS is the first given in Section A.1. That particular version sets a field strength of 50 T (i.e., 50.0 in bold font on line 17). This number can be changed to 0.0 or 25.0 to obtain results for the corresponding imposed field strength. Simulation results described next are obtained from the file `out.txt`.

Selected results are shown in Fig. 1. In the top image, the experimental data of Taylor et al.[57] are obtained from diamond anvil cell (DAC) experiments. In the bottom two images, the evolution of the phase volume fraction (left) and density of the mixture (right) are compared for different field strengths. For compressive pressures exceeding around 13 GPa, forward transformation from α to ϵ phases begins. Reverse transformation occurs on decompression. The magnetic field increases the pressure required for transformation. Results are discussed in more detail by Clayton et al.[12]

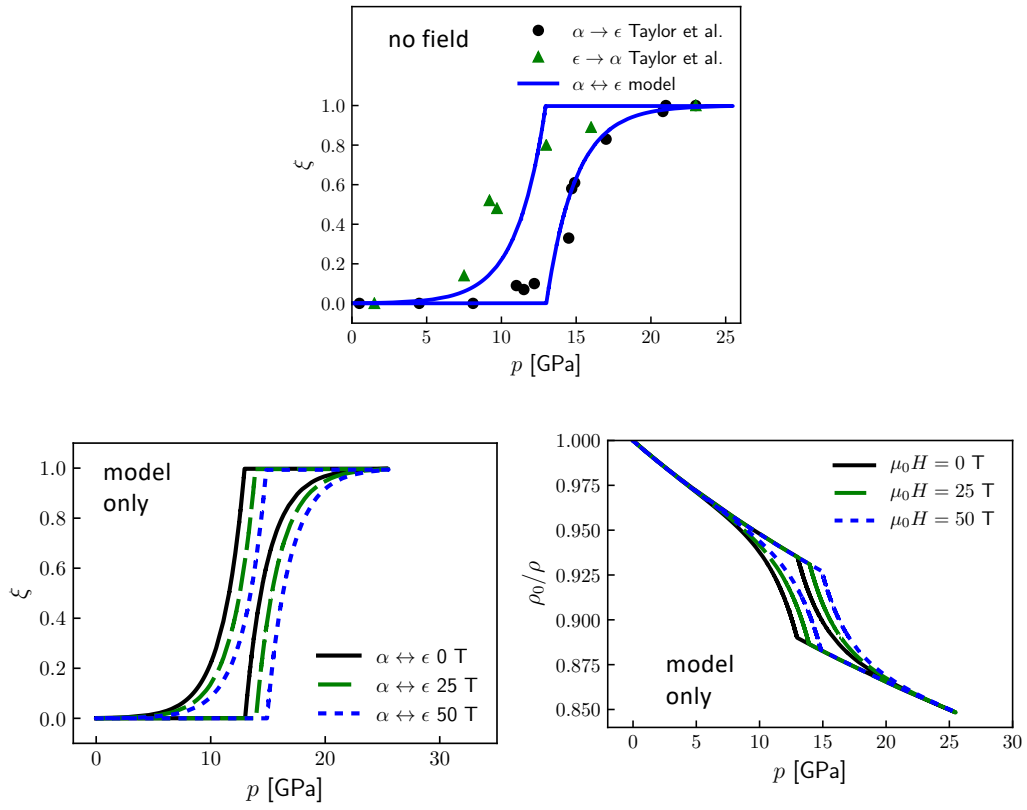

Fig. 1 Hydrostatic compression and decompression of pure Fe, with model results obtained from CMF1. Top image shows results of ϵ phase ξ vs. pressure p for null applied magnetic field compared to experimental DAC data.[57] Bottom left image shows predicted ξ for field strengths of 0, 25, and 50 T for loading-unloading cycles to 25.5 GPa; bottom right image shows density ratio $J = \rho_0/\rho$ vs. p for the same cycles, with hysteresis enclosed by the curves.

The second problem for CMF1 considers a ferrous alloy (steel) designed and fabricated at ARL. Experiments and simulations[12] subject a sample of this material to quasi-static, uniaxial stress extension, with or without a magnetic field applied orthogonal to the direction of mechanical loading. If applied, the static magnetic field is of strength $\mu_0 H = 0.9$ T. Isothermal conditions at 300 K, without electrical or thermal currents, are again assumed. Experiments at null applied field, and both simulations, continue extension of the sample until mechanical failure.

This problem exercises the logarithmic EOS, the plasticity model, the phase transition model, the quadratic spline magnetization model (`magmod` of 1), and the damage model. The input file to the MPS is the second given in Section A.1. The particular version there sets a field strength of 0.9 T (i.e., 0.9 in bold font on line

17). This number can be changed to 0.0 to obtain the results for null applied field. Simulation results described next are obtained from the file `out.txt`.

Selected results are shown in Fig. 2. In the top left image, no field is imposed (i.e., $H = 0$), and the predicted nominal stress $\hat{T}_{33}$ matches the experimental data including the failure point signified by the abrupt load drop. In the top right image, the magnetic field is of strength 0.9 T, and the simulation matches the experiment to the applied strain at which the latter was ceased. In the bottom image, the evolutions of the phase volume fractions are compared versus experimental data obtained from electron backscatter diffraction (EBSD). The transformation is of forward type only, from the close-packed γ/ϵ phase to the less dense α phase. The initial fraction of α is 0.058. The magnetic field reduces the applied tensile stress and strain required for transformation. Results are discussed in more detail by Clayton et al.[12]; the alloy is labeled AR for as-rolled in that reference.

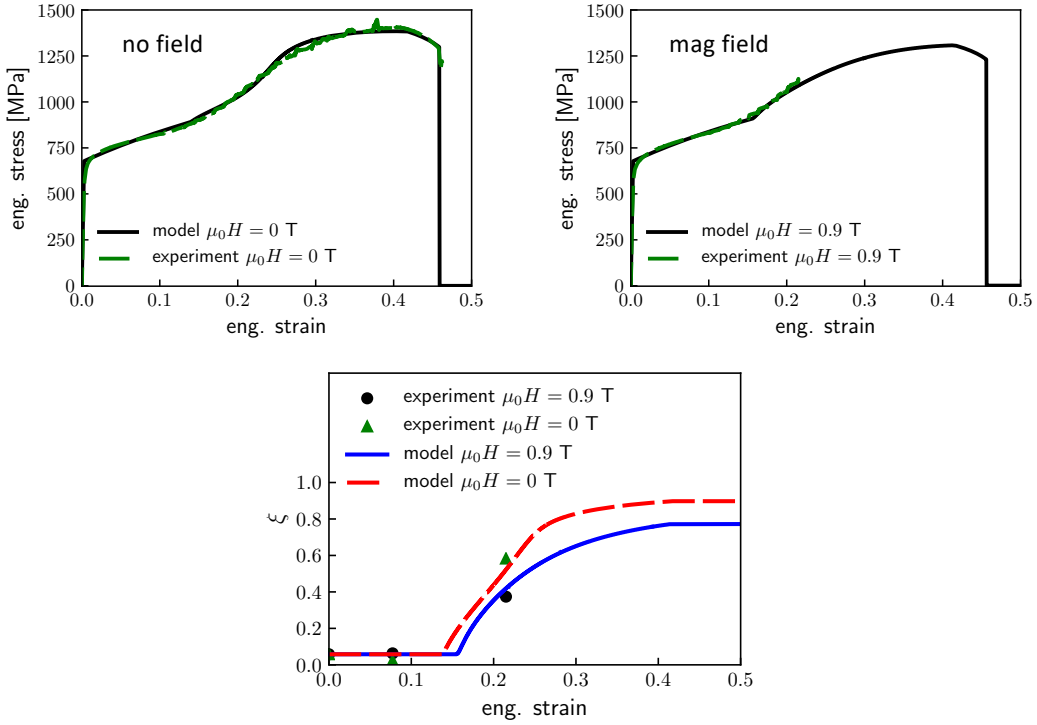

Fig. 2 Tension of a ferrous alloy, with model results obtained from CMF1. Respective top left and right images show engineering stress without and with applied field of strength 0.9 T. Bottom image shows volume fraction of α phase including experimental data from EBSD.[12]

7.2 CMF2

The first problem class for CMF2 considers adiabatic uniaxial-strain compression to a volume reduction of 35% (i.e., to $J = 0.65$). Also addressed is isothermal hydrostatic compression to the same volume reduction. In each problem, the initial temperature $\theta_i = \theta(0)$ is 300 K, and the material is a TRIP steel designed and processed at ARL.[3,10] The material consists primarily of γ phase initially, and the possible phase transition modeled by CMF2 is $\gamma \to \alpha$. The high-pressure state modeled in this example precludes transformation to the less dense α phase, and the model predicts that no transition to α occurs for hydrostatic or uniaxial strain compression. Similarly, no damage in the form of voids arises for compressive loading. This problem exercises the Eulerian EOS and the plasticity model, where the latter becomes inactive for hydrostatic compression.

The input files to the MPS are the first and second given in Section A.2. The first corresponds to adiabatic uniaxial strain, the second to isothermal spherical compression. Deformation rates are quasi-static. Simulation results described next are obtained from the file `out.txt`.

No corresponding experiments for these loading protocols have been performed on this material, so results of axial stress P for uniaxial-strain compression are compared with plate impact shock-compression data on several stainless steels.[65] Comparisons are shown in the top image of Fig. 3. The hydrostat is slightly lower than the adiabat because the hydrostat omits material strength and thermal expansion. At large compression $\epsilon = 1 - J \gtrsim 0.25$, the model underpredicts the shock response since the homogeneous strain approximation omits shock dissipation that is order three in compressive strain.[25,26,52] The bottom left image in Fig. 3 shows the temperature θ, the bottom right the shear stress τ. As physically expected, isothermal hydrostatic compression produces no shear stress.

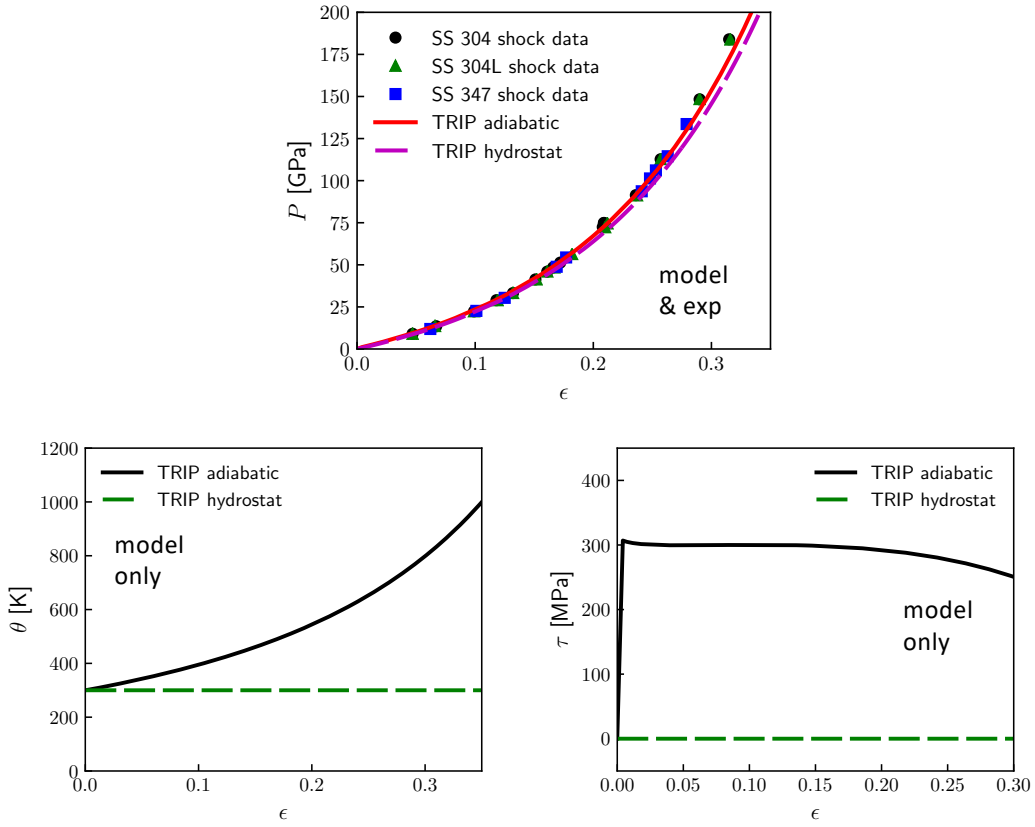

Fig. 3 Adiabatic uniaxial-strain and isothermal hydrostatic compression of a TRIP steel, with model results obtained from CMF2. Top image compares predictions with shock compression data on three stainless steels.[65] Respective bottom left and right images show temperature and shear stress.[10]

The second problem class for CMF2 considers uniaxial-stress compression of the same TRIP steel[2,10,64] at low and high loading rates and low and high initial temperatures. Quasi-static loading is imposed with $\dot{\epsilon} = 10^{-3}$/s under isothermal conditions at either 293 K or 473 K. Dynamic loading is imposed with $\dot{\epsilon} = 2500$/s under adiabatic conditions with an initial temperature θ_i of either 293 K or 473 K. No failure by void coalescence arises for compressive loading; the model correctly predicts this. Even though the pressure is compressive, the shear driving force is sufficient to initiate the $\gamma \rightarrow \alpha$ transition for uniaxial stress loading at initially room temperature conditions (i.e., $\theta_i = 293$ K). At elevated temperature (i.e., 473 K), the phase transition is suppressed. This class of problems exercises the Eulerian EOS, the plasticity model, and the phase transformation model.

The input files to the MPS are the third, fourth, fifth, and sixth given in Section A.2. Notice that θ_0 is set to the initial temperature θ_i in every case to avoid pressure from thermal expansion, manifesting from $\theta \neq \theta_0$ the EOS, that would otherwise arise before deformation is applied, at $t = 0$. Simulation results are again obtained from the file `out.txt`. The source code of `wrapper.txt` is edited so the true Cauchy stress component $\sigma = |\sigma_{33}|$ is printed to the file `out.txt` rather than the nominal (i.e., engineering) stress printed by default in Section 7.1. This source code is likewise edited to output the logarithmic strain $|\ln F_{33}|$ used in Fig. 4.

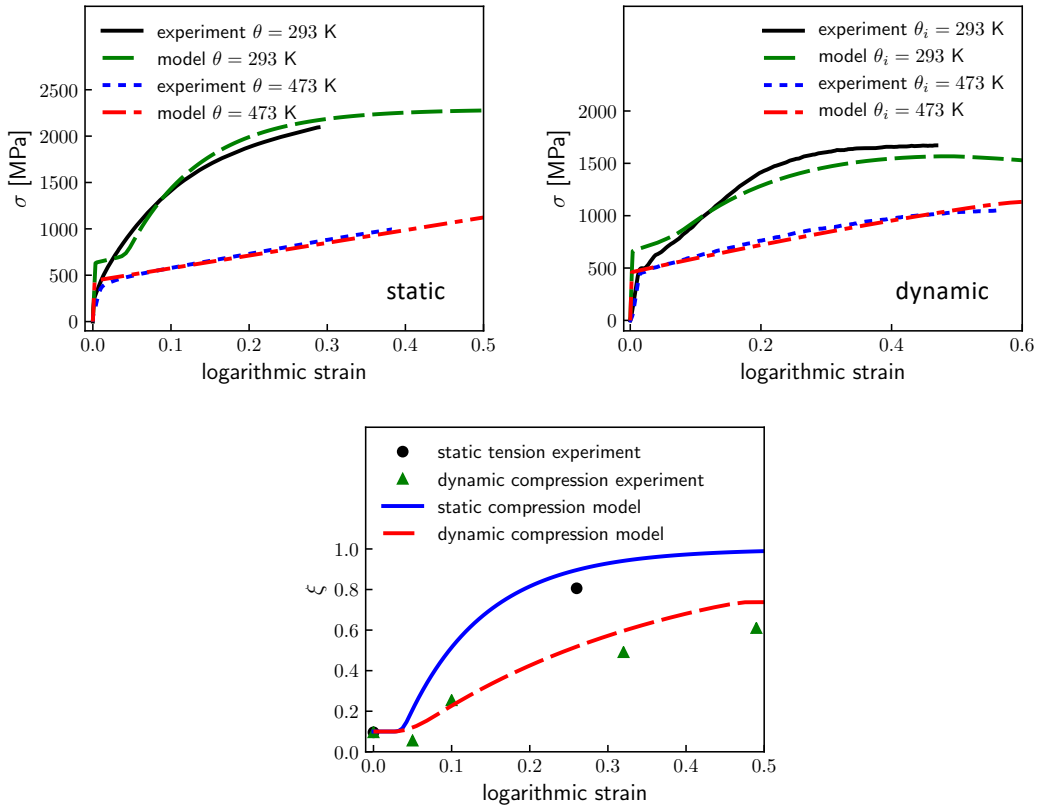

Fig. 4 Uniaxial-stress compression of a TRIP steel, with model results obtained from CMF2. Respective top left and right images show quasi-static and dynamic compression. Results of simulations and experiments obtained at room temperature and elevated temperature. Bottom image shows volume fraction of α phase including experimental data from EBSD.[10]

True stress-strain predictions from CMF2 are compared with experimental data in the top two images of Fig. 4. The phase volume fraction ξ is compared with EBSD data in the bottom image of Fig. 4. Experimental data on ξ do not exist for static compression, but static tensile data do exist and are shown for reference. Stress-

strain predictions agree closely with experimental data at high temperatures when transitions are absent. They agree reasonably well with experimental data at lower temperatures when transitions occur. Predicted phase volume fractions are less accurate, but are still regarded as reasonable given the uncertainty and sample-to-sample variations affecting experimental measurements of the α phase. Detailed discussion of these results, as well as origins of material parameters used to obtain them, can be found in Clayton and Lloyd.[10]

8. Conclusion

Theory and software for numerical simulations of the constitutive response of ferrous metals have been documented for two model frameworks Both frameworks account for nonlinear elasticity, plasticity, phase transitions, and damage. One framework also includes magnetization and electromechanical forces. Verification problems have been documented for both frameworks. A standalone simulator that can be used, for homogeneous loading conditions, as a substitute for a larger scale host code has also been documented. Corresponding input and source files for the simulator are included in Appendices A and B.

9. References

1. Field D, Baker D, Van Aken D. On the prediction of α-martensite temperatures in medium manganese steels. Metallurgical and Materials Transactions A. 2017;48:2150–2163.

2. Lloyd J, Field D, Limmer K. A four parameter hardening model for TWIP and TRIP steels. Materials and Design. 2020;194:108878.

3. Limmer K, Lloyd J, Field D, Magagnosc D, Hornbuckle B, Walter T, Meredith C, Clayton J, Jannotti P. Designing steels to mitigate failure during ballistic deformation (summary technical report, Oct 2018–Sept 2021). DEVCOM Army Research Laboratory; 2022. Report No.: ARL-TR-9424.

4. Curran D. Dynamic mechanical behavior of iron. In: Shock Waves and the Mechanical Properties of Solids. Burke J, Weiss V, editors. Syracuse University Press; 1971. p. 121–138.

5. Gilder S, Glen J. Magnetic properties of hexagonal closed-packed iron deduced from direct observations in a diamond anvil cell. Science. 1998;279:72–74.

6. Murdoch H, Hernández-Rivera E, Giri A. Modeling magnetically influenced phase transformations in alloys. Metallurgical and Materials Transactions A. 2021;52:2896–2908.

7. Murdoch H, Hernandez E, Giri A, Field D. Development of a Python module for calculating magnetization and magnetically influenced phase transformations. DEVCOM Army Research Laboratory; 2022. Report No.: ARL-TR-9527.

8. Andrews D. Equation of state of the alpha and epsilon phases of iron. Journal of the Physics and Chemistry of Solids. 1973;34:825–840.

9. Boettger J, Wallace D. Metastability and dynamics of the shock-induced phase transition in iron. Physical Review B. 1997;55:2840–2849.

10. Clayton J, Lloyd J. A dynamic finite-deformation constitutive model for steels undergoing slip, twinning, and phase changes. Journal of Dynamic Behavior of Materials. 2021;7:217–247.

11. Clayton J, Lloyd J. Finite strain continuum theory for phase transformations in ferromagnetic elastic-plastic solids. Continuum Mechanics and Thermodynamics. 2022;34:1579–1620.

12. Clayton J, Murdoch H, Lloyd J, Magagnosc D, Field D. Modeling magnetic field and strain driven phase transitions and plasticity in ferrous metals. 2023; forthcoming.

13. Lloyd J, Field D, Magagnosc D, Limmer K, Turnage S, Williams C, Clayton J. Manipulating shock waves with metallurgy. Acta Materialia. 2022;234:118042.

14. Lee S, Cho H, Bronkhorst C, Pokharel R, Brown D, Clausen B, Vogel S, Anghel V, Gray G, Mayeur J. Deformation, dislocation evolution and the non-Schmid effect in body-centered-cubic single-and polycrystal tantalum. International Journal of Plasticity. 2023;163:103529.

15. Zubelewicz A, Clayton J. Yield surfaces and plastic potentials for metals, with analysis of plastic dilatation and strength asymmetry in BCC crystals. Metals. 2023;13:523.

16. Clayton J. An alternative three-term decomposition for single crystal deformation motivated by non-linear elastic dislocation solutions. Quarterly Journal of Mechanics and Applied Mathematics. 2014;67:127–158.

17. Clayton J. Defects in nonlinear elastic crystals: differential geometry, finite kinematics, and second-order analytical solutions. Zeitschrift fur Angewandte Mathematik und Mechanik (ZAMM). 2015;95:476–510.

18. Hibbitt D, Karlsson B, Sorensen P. ABAQUS theory manual. HKS Corporation; 2006.

19. Noble C, Anderson A, Barton et al. N. ALE3D: An arbitrary Lagrangian-Eulerian multi-physics code. Lawrence Livermore National Laboratory (LLNL); 2017. Report No.: LLNL-TR-732040.

20. Haill K T.A. Cochrane, Garasi C, Mehlhorn T, Robinson A, Summers R. ALEGRA-MHD: Version 4.6. Sandia National Laboratories; 2005. Report No.: SAND2004-5997.

21. Johnson G, Stryk R, Holmquist T, Beissel S. Numerical algorithms in a Lagrangian hydrocode. Alliant Techsystems Inc.; 1997. Report No.: WL-TR-1997-7039.

22. Gerlach C, Johnson G, Becker R, Lloyd J, Tonge A. Documentation of distributed user material (UMAT) and vectorized user material (VUMAT) models. DEVCOM Army Research Laboratory; 2020. Report No.: ARL-TN-1023.

23. Maugin G. Continuum mechanics of electromagnetic solids. North-Holland; 1988.

24. Clayton J. Nonlinear mechanics of crystals. Springer; 2011.

25. Clayton J. Nonlinear elastic and inelastic models for shock compression of crystalline solids. Springer; 2019.

26. Davison L. Fundamentals of shock wave propagation in solids. Springer; 2008.

27. Maugin G, Eringen A. Deformable magnetically saturated media. I. Field equations. Journal of Mathematical Physics. 1972;13:143–155.

28. Clayton J. Finsler differential geometry in continuum mechanics: fundamental concepts, history, and renewed application to ferromagnetic solids. Mathematics and Mechanics of Solids. 2022;27:910–949.

29. Landau L, Lifshitz E, Pitaevskii L. Electrodynamics of continuous media. 2nd ed. Pergamon; 1982.

30. Clayton J. Analysis of shock compression of strong single crystals with logarithmic thermoelastic-plastic theory. International Journal of Engineering Science. 2014;79:1–20.

31. Clayton J. Dynamic plasticity and fracture in high density polycrystals: constitutive modeling and numerical simulation. Journal of the Mechanics and Physics of Solids. 2005;53:261–301.

32. Clayton J. Modeling dynamic plasticity and spall fracture in high density polycrystalline alloys. International Journal of Solids and Structures. 2005;42:4613–4640.

33. Clayton J. A continuum description of nonlinear elasticity, slip and twinning, with application to sapphire. Proceedings of the Royal Society of London A. 2009;465:307–334.

34. Luscher D, Bronkhorst C, Alleman C, Addessio F. A model for finite-deformation nonlinear thermomechanical response of single crystal copper under shock conditions. Journal of the Mechanics and Physics of Solids. 2013;61:1877–1894.

35. Lloyd J, Clayton J, Austin R, McDowell D. Plane wave simulation of elastic-viscoplastic single crystals. Journal of the Mechanics and Physics of Solids. 2014;69:14–32.

36. Bronkhorst C, Cerreta E, Xue Q, Maudlin P, Mason T, Gray G. An experimental and numerical study of the localization behavior of tantalum and stainless steel. International Journal of Plasticity. 2006;22:1304–1335.

37. Grujicic M, Sankaran N. Dispersed-phase martensitic transformation controlled deformation behavior of two-phase metallic materials. International Journal of Solids and Structures. 1997;34:4421–4446.

38. Bammann D, Chiesa M, Horstemeyer M, Weingarten L. Failure in ductile materials using finite element methods. In: Structural crashworthiness and failure. Jones N, Wierzbicki T, editors. Elsevier Applied Science, The Universities Press; 1993. p. 1–54.

39. Marin E, McDowell D. Associative versus non-associative porous viscoplasticity based on internal state variable concepts. International Journal of Plasticity. 1996;12:629–669.

40. Stringfellow R, Parks D, Olson G. A constitutive model for transformation plasticity accompanying strain-induced martensitic transformations in metastable austenitic steels. Acta Metallurgica et Materialia. 1992;40:1703–1716.

41. Tomita Y, Iwamoto T. Constitutive modeling of TRIP steel and its application to the improvement of mechanical properties. International Journal of Mechanical Sciences. 1995;37:1295–1305.

42. Preston D, Tonks D, Wallace D. Model of plastic deformation for extreme loading conditions. Journal of Applied Physics. 2003;93:211–220.

43. Lloyd J, Clayton J, Becker R, McDowell D. Simulation of shock wave propagation in single crystal and polycrystalline aluminum. International Journal of Plasticity. 2014;60:118–144.

44. Lloyd J, Clayton J, Austin R, McDowell D. Shock compression modeling of metallic single crystals: comparison of finite difference, steady wave, and analytical solutions. Advanced Modeling and Simulation in Engineering Sciences. 2015;2:14.

45. de Rességuier T, Hallouin M. Effects of the α- ε phase transition on wave propagation and spallation in laser shock-loaded iron. Physical Review B. 2008;77:174107.

46. Tuler F, Butcher B. A criterion for the time dependence of dynamic fracture. International Journal of Fracture Mechanics. 1968;4:431–437.

47. Davison L, Stevens A. Continuum measures of spall damage. Journal of Applied Physics. 1972;43:988–994.

48. Hanim S, Ahzi S. A unified approach for pressure and temperature effects in dynamic failure criteria. International Journal of Plasticity. 2001;17:1215–1244.

49. Cocks A, Ashby M. On creep fracture by void growth. Progress in Materials Science. 1982;27:189–244.

50. Daniel L, Hubert O, Buiron N, Billardon R. Reversible magneto-elastic behavior: a multiscale approach. Journal of the Mechanics and Physics of Solids. 2008;56:1018–1042.

51. James R, Kinderlehrer D. Frustration in ferromagnetic materials. Continuum Mechanics and Thermodynamics. 1990;2:215–239.

52. Clayton J. Nonlinear Eulerian thermoelasticity for anisotropic crystals. Journal of the Mechanics and Physics of Solids. 2013;61:1983–2014.

53. Mackenzie J. The elastic constants of a solid containing spherical holes. Proceedings of the Physical Society B. 1950;63:2–11.

54. Clayton J, Freed A. A constitutive framework for finite viscoelasticity and damage based on the Gram-Schmidt decomposition. Acta Mechanica. 2020;231:3319–3362.

55. Turteltaub S, Suiker A. Transformation-induced plasticity in ferrous alloys. Journal of the Mechanics and Physics of Solids. 2005;53:1747–1788.

56. Turteltaub S, Suiker A. A multiscale thermomechanical model for cubic to tetragonal martensitic phase transformations. International Journal of Solids and Structures. 2006;43:4509–4545.

57. Taylor R, Pasternak M, Jeanloz R. Hysteresis in the high pressure transformation of bcc-to hcp-iron. Journal of Applied Physics. 1991;69:6126–6128.

58. Bancroft D, Peterson E, Minshall S. Polymorphism of iron at high pressure. Journal of Applied Physics. 1956;27:291–298.

59. Barker L, Hollenbach R. Shock wave study of the $\alpha \leftrightarrow \varepsilon$ phase transition in iron. Journal of Applied Physics. 1974;45:4872–4887.

60. Mao HK, Bassett W, Takahashi T. Effect of pressure on crystal structure and lattice parameters of iron up to 300 kbar. Journal of Applied Physics. 1967;38:272–276.

61. Giles P, Longenbach M, Marder A. High-pressure $\alpha \leftrightarrow \epsilon$ martensitic transformation in iron. Journal of Applied Physics. 1971;42:4290–4295.

62. Magagnosc D, Field D, Meredith C, Walter T, Limmer K, Lloyd J. Superior strength and ductility in a low density duplex steel studied by in situ neutron diffraction. Materials Science and Engineering A. 2021;799:140252.

63. Magagnosc D, Field D, Meredith C, An K, Walter T, Limmer K, Lloyd J. Temperature and stress dependent twinning behavior in a fully austenitic medium-Mn steel. Acta Materialia. 2022;231:117864.

64. Lloyd J, Magagnosc D, Meredith C, Limmer K, Field D. Improved dynamic strength of trip steel via pre-straining. Scripta Materialia. 2022;220:114941.

65. S. Marsh (ed.). LASL shock hugoniot data. University of California Press; 1980.

Appendix A. Input Files for Verification Examples

Inputs necessary to solve the verification problems of Section 7 follow subsequently. Each input should be saved in a file called `sim.inp` for use with the material point simulator (MPS) of Appendix B. If the MPS is not used, the syntax will differ according to that required by the host code. Recall that only CMF1 formally accommodates magnetic fields and explicit magnetization effects.

A.1 CMF1

The input syntax below is used to generate results for hydrostatic compression and decompression of pure Fe under an applied magnetic field. These results are shown in Fig. 1 and discussed in Section 7.1.

```
*MATERIAL,NAME=pureiron
*DEPVAR
27
*USER MATERIAL,CONSTANTS=96
103666.7, 89000.0, 300.0, 1811.0, 3,600.0, 0.0,5.0
0.0, 0.001, 0.0, 10.0, 0.000001, 0, 10.0, 0.01
0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 5.0
-3, 3.54, 163000.0, 5.29, 0.000038, 0.0, 0.0, 0.0
0.0, 0.0, 0.60, 5, 0.0, 4, -0.05122, 0.0
0.0, 0.0, 90.5, 90.5, 519.0, 171.0, 763.0, 0.0
600.0, 0.0, 5.0, 0.0, 25.0, 25.0, 0.0,10000.0
1,0.000001, 10.0,78.5, -11.9,0.0, 0.05,0.05
2.15,0.0, 300.0, 0.0,0.0,0.0,0.0, 0
2.15,0.0,0.0, 0.0,0.0,0.0, 0.0,0.0
0.0, 0.0,0.0,0.0, 0.0,0.0,0.0, 0.0
0.0,0.0, 3.54,0.0,0.0, 3.54,0.0,0.0
H,50.0,0.0, 0.0,0.0,1.0, 0.0,0.0,0.0
*STEP
0.001, 160.0, 0.01
V33, -0.001
*STEP
0.001, 160.0, 0.01
V33, 0.001
```

The input syntax below is used to generate results for tension of a ferrous alloy

deformed under an applied magnetic field. Results are shown in Fig. 2 and discussed in Section 7.1.

```
*MATERIAL,NAME=steelone
*DEPVAR
27
*USER MATERIAL,CONSTANTS=96
120312.5, 94531.3, 300.0, 1811.0, 3,610.0, 5000.0,0.1
1.0, 0.001, 0.0, 10.0, 0.000001, 7, 15.0, 0.01
0.0, 0.0, 0.0, 0.0, 0.0, 0.1, 0.05, 5.0
-3, 3.665, 183333.3, 5.29, 0.0000607, 0.0, 0.0, 0.0
0.0, 0.0, 0.90, 5, 0.058, 4, 0.03325, 0.1
469.0,9999.0,22.0,1.0, -0.0655,-8433.41,315.64,0.195
1800.0, 2000.0, 0.1, 3.0, 25.0,25.0, 0.0,10000.0
1,0.000001, 10.0,78.5, 0.0,-11.9, 0.05,0.05
0.0,1.64, 300.0, -0.0000228,7.77778,7.77778,-0.062222, 1
0.0,0.0001238,0.0, 1.64,0.000139,0.0, 0.0,0.0
-0.00005, 0.0,-6.86,0.981, 0.0,0.0,0.0, 0.0000369
0.000813,-0.000165, 3.6746,0.0,0.0, 3.5113,0.000366,-0.000079
H,0.9,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0
*STEP
0.0001, 1000, 0.5
C33, 0.001
```

A.2 CMF2

The input syntax below is used to generate results for confined compression of a TRIP steel. Results are shown in Fig. 3 and discussed in Section 7.2. The first input data correspond to adiabatic uniaxial-strain compression:

```
*MATERIAL,NAME=tripstee
*DEPVAR
27
*USER MATERIAL,CONSTANTS=96
109615.0,73077.0,300.0, 1800.0,3,450.0,2200.0,20.0
0.6, 0.001, 0.003, 0.55, 0.000001, 5, 10.0, 0.001
0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 5.0
```

2, 3.59, 158333.3, 5.3, 0.0000519, 0.0, 0.0, 0.0
0.0, 0.0, 0.80, 3, 0.10, 2, 0.04, 0.26
1.0, 0.0, 1.0, 0.08, 0.003, -100.0, 180.0, 0.02
2300.0, 0.0, 5.0, 0.6, 0.001, 5.0, 0.25, 10000.0
0,0.000001, 10.0,38.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 300.0, 0.0,0.0,0.0,0.0, 0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
H,0.0,0.0, 0.0,0.0,0.0, 0.0,0.0,0.0
*STEP
0.001, 350, 0.01
U33, -0.001

The second input data correspond to isothermal hydrostatic compression:

*MATERIAL,NAME=tripstee
*DEPVAR
27
*USER MATERIAL,CONSTANTS=96
109615.0,73077.0,300.0, 1800.0,3,450.0,2200.0,20.0
0.6, 0.001, 0.003, 0.55, 0.000001, 5, 10.0, 0.001
0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 5.0
-2, 3.59, 158333.3, 5.3, 0.0000519, 0.0, 0.0, 0.0
0.0, 0.0, 0.80, 3, 0.10, 2, 0.04, 0.26
1.0, 0.0, 1.0, 0.08, 0.003, -100.0, 180.0, 0.02
2300.0, 0.0, 5.0, 0.6, 0.001, 5.0, 0.25, 10000.0
0,0.000001, 10.0,38.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 300.0, 0.0,0.0,0.0,0.0, 0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
H,0.0,0.0, 0.0,0.0,0.0, 0.0,0.0,0.0
*STEP
0.001, 410.0, 0.01
V33, -0.001

The input syntax below (four sets) is used to generate results for unconfined compression of a TRIP steel. Results are shown in Fig. 4 and discussed in Section 7.2. The next input data correspond to quasi-static, room-temperature compression:

```
*MATERIAL,NAME=tripstee
*DEPVAR
27
*USER MATERIAL,CONSTANTS=96
109615.0,73077.0,293.0, 1800.0,3,450.0,2200.0,20.0
0.6, 0.001, 0.003, 0.55, 0.000001, 5, 10.0, 0.001
0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 5.0
-2, 3.59, 158333.3, 5.3, 0.0000519, 0.0, 0.0, 0.0
0.0, 0.0, 0.80, 3, 0.10, 2, 0.04, 0.26
1.0, 0.0, 1.0, 0.08, 0.003, -100.0, 180.0, 0.02
2300.0, 0.0, 5.0, 0.6, 0.001, 5.0, 0.25, 10000.0
0,0.000001, 10.0,38.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 293.0, 0.0,0.0,0.0,0.0, 0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
H,0.0,0.0, 0.0,0.0,0.0, 0.0,0.0,0.0
*STEP
0.0001, 500.0, 5.0
C33, -0.001
```

The next input data correspond to quasi-static, high-temperature compression:

```
*MATERIAL,NAME=tripstee
*DEPVAR
27
*USER MATERIAL,CONSTANTS=96
109615.0,73077.0,473.0, 1800.0,3,450.0,2200.0,20.0
0.6, 0.001, 0.003, 0.55, 0.000001, 5, 10.0, 0.001
0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 5.0
-2, 3.59, 158333.3, 5.3, 0.0000519, 0.0, 0.0, 0.0
0.0, 0.0, 0.80, 3, 0.10, 2, 0.04, 0.26
```

1.0, 0.0, 1.0, 0.08, 0.003, -100.0, 180.0, 0.02
2300.0, 0.0, 5.0, 0.6, 0.001, 5.0, 0.25, 10000.0
0,0.000001, 10.0,38.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 473.0, 0.0,0.0,0.0,0.0, 0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
H,0.0,0.0, 0.0,0.0,0.0, 0.0,0.0,0.0
*STEP
0.0001, 500.0, 5.0
C33, -0.001

The next input data are for dynamic, initially room-temperature compression:

*MATERIAL,NAME=tripstee
*DEPVAR
27
*USER MATERIAL,CONSTANTS=96
109615.0,73077.0,293.0, 1800.0,3,450.0,2200.0,20.0
0.6, 0.001, 0.003, 0.55, 0.000001, 5, 10.0, 0.001
0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 5.0
2, 3.59, 158333.3, 5.3, 0.0000519, 0.0, 0.0, 0.0
0.0, 0.0, 0.80, 3, 0.10, 2, 0.04, 0.26
1.0, 0.0, 1.0, 0.08, 0.003, -100.0, 180.0, 0.02
2300.0, 0.0, 5.0, 0.6, 0.001, 5.0, 0.25, 10000.0
0,0.000001, 10.0,38.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 293.0, 0.0,0.0,0.0,0.0, 0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
H,0.0,0.0, 0.0,0.0,0.0, 0.0,0.0,0.0
*STEP
0.0000000001, 0.0002, 0.000005
C33, -2500.0

The final input data are for dynamic, initially elevated-temperature compression:

*MATERIAL,NAME=tripstee
*DEPVAR
27
*USER MATERIAL,CONSTANTS=96
109615.0,73077.0,473.0, 1800.0,3,450.0,2200.0,20.0
0.6, 0.001, 0.003, 0.55, 0.000001, 5, 10.0, 0.001
0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 5.0
2, 3.59, 158333.3, 5.3, 0.0000519, 0.0, 0.0, 0.0
0.0, 0.0, 0.80, 3, 0.10, 2, 0.04, 0.26
1.0, 0.0, 1.0, 0.08, 0.003, -100.0, 180.0, 0.02
2300.0, 0.0, 5.0, 0.6, 0.001, 5.0, 0.25, 10000.0
0,0.000001, 10.0,38.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 473.0, 0.0,0.0,0.0,0.0, 0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
0.0,0.0, 0.0,0.0, 0.0,0.0, 0.0,0.0
H,0.0,0.0, 0.0,0.0,0.0, 0.0,0.0,0.0
*STEP
0.0000000001, 0.0002, 0.000005
C33, -2500.0

Appendix B. MPS Source Code (`mps.f`)

```
c-----------------------------------------------------------------------
c
c  Material Point Simulator (MPS) for mechanics and magnetism
c  Uses: umat.f, magumat.f
c  Input: sim.inp
c  Output: sim.txt, out.txt, terminal
c  FORTRAN source
c  Version 22 April 2023
c
c-----------------------------------------------------------------------

      include 'umat.f'
      include 'magumat.f'

      implicit double precision (a-h,o-z)

c-----------------------------------------------------------------------

      parameter( Max_isv = 20000,
     & Max_props = 100,
     & Tolerance = .00001,
     & Max_itr   = 10,
     & ntens     = 6,
     & ndi       = 3,
     & nshr      = 3,
     & noel      = 1,
     & Num_modes = 8)

      character(8)   cmname
      character(4)   strtoint
      character(132) text
      character(80)  loading_mode
      character(80)  mloading_mode
      character(80)  loading_key(2,Num_modes)
      character(80)  mloading_key(2,2)

      logical Out_of_time

c-----------------------------------------------------------------------
c  Dimension arrays used only in this main program.
c-----------------------------------------------------------------------

      dimension
     &  array1  (3,3),       ! Dummy Array
     &  array2  (3,3),       ! Dummy Array
     &  array3  (3,3),       ! Dummy Array
     &  D_dt    (3,3),       ! Rate of Def tensor * dtime
     &  W_dt    (3,3),       ! Spin tensor * dtime
     &  sig (3,3),           ! Stress Tensor
     &  statev_ref(Max_isv), ! Reference ISVs
     &  stress0(6),
     &  bdir(3)              ! B-H field direction

c-----------------------------------------------------------------------
c  Dimension arrays passed into the UMAT sub
c-----------------------------------------------------------------------

      dimension
     & coords(3),            ! Coordinates of Gauss pt. being evaluated
```

```
     & ddsdde(ntens,ntens),  ! Tangent Stiffness Matrix
     & ddsddt(ntens),    ! Change in stress per change in temperature
     & dfgrd0(3,3),      ! Deformation gradient at beginning of step
     & dfgrd1(3,3),      ! Deformation gradient at end of step
     & dpred(1),         ! Change in predefined state variables
     & drplde(ntens),    ! Change in heat generation per change in strain
     & drot(3,3),        ! Rotation matrix
     & dstrain(ntens),   ! Strain increment tensor stored in vector form
     & predef(1),        ! Predefined state vars dependent on field variables
     & props(Max_props), ! Material properties passed in
     & statev(Max_isv),  ! State Variables
     & strain(ntens),    ! Strain tensor stored in vector form
     & stress(ntens),    ! Cauchy stress tensor stored in vector form
     & time(2)           ! Step Time and Total Time

c-----------------------------------------------------------------------
c  Initialize loading modes
c-----------------------------------------------------------------------

      data
     &  loading_key(1,1),loading_key(2,1) / 'C', 'TensionComp' /
     &  loading_key(1,2),loading_key(2,2) / 'T', 'Torsion'     /
     &  loading_key(1,3),loading_key(2,3) / 'P', 'PlaneStrain' /
     &  loading_key(1,4),loading_key(2,4) / 'S', 'SimuComTors' /
     &  loading_key(1,5),loading_key(2,5) / 'B', 'BiaxialTens' /
     &  loading_key(1,6),loading_key(2,6) / 'U', 'UniaxStrain' /
     &  loading_key(1,7),loading_key(2,7) / 'V', 'Volumetric' /
     &  loading_key(1,8),loading_key(2,8) / 'Z', 'PressShear' /
     &  mloading_key(1,1),mloading_key(2,1) / 'H', 'HField' /
     &  mloading_key(1,2),mloading_key(2,2) / 'B', 'BField' /


c-----------------------------------------------------------------------
c  Initialize Pi
c-----------------------------------------------------------------------

      Pi = 4. * atan(1.d0)

      xmu0 = Pi*4.0*1.0d-07

c-----------------------------------------------------------------------
c  Initialize time & deformation gradients & temperature increment
c-----------------------------------------------------------------------

      time(1) = 0.0
      time(2) = 0.0
      E_eff   = 0.0
      istep   = 0

      do i = 1,6
        strain(i)  = 0.0
        dstrain(i) = 0.0
      end do

      do i = 1,3
       do j = 1,3
         dfgrd0(i,j) = 0.0
         dfgrd1(i,j) = 0.0
         drot  (i,j) = 0.0
```

```
       end do
       dfgrd0(i,i) = 1.0
       dfgrd1(i,i) = 1.0
       drot  (i,i) = 1.0
      end do

      dtemp = 0.0

c-----------------------------------------------------------------------
c    Initialize mag field variables
c-----------------------------------------------------------------------

      bmag = 0.0
      hmag = 0.0
      efield = 0.0
      dtempdx = 0.0
      d2tempdx2 = 0.0

c-----------------------------------------------------------------------
c  Open ABAQUS files
c-----------------------------------------------------------------------

      open(1,file='inp/sim.inp',status='old')
      open(8,file='out/sim.txt',status='unknown')
      open(9,file='out/out.txt',status='unknown')

c-----------------------------------------------------------------------
c  Print header in sim.txt and out.txt files
c-----------------------------------------------------------------------

      write(8,'(42a,38a)')' STEP     TIME     E_eff     SIG(1,1) ',
     & ' SIG(2,2)  SIG(3,3)  SIG(1,2)  SIG(1,3)  SIG(2,3) ',
     & ' F(1,1)    F(2,2)    F(3,3)    F(1,2)    F(2,3)    F(1,3)  ',
     & '  EPEFF    EPEFFD '

      write(9,'(42a,38a)')' STEP     TIME    STRAIN      DETF    ',
     & 'STRESS     SHEAR    TEMPER    DAMAGE     DFLAG     ',
     & 'DMDOT     MUEFF     VFRAC      VFDOT       VFQ      ',
     & '   B         M      DPHI'

c-----------------------------------------------------------------------
c  Read in number of ISVs
c-----------------------------------------------------------------------

  100 read(1,'(a132)') text
      n = index(text,'DEPVAR')
      if (n .eq. 0) go to 100

      read(1,*) nstatv
      if (nstatv .gt. Max_isv) then
        print*,'STOP!!!  The number of state variables in the input'
        print*,'         file exceeds the max value in the Max_isv'
        print*,'         variable in the program.'
        Print*,' '
        Print*,'Set Max_isv in program to at least: ',nstatv+1,'.'
        Print*,' '
        STOP
      end if
```

```
c-----------------------------------------------------------------------
c  Read in number of material properties
c-----------------------------------------------------------------------

  200 read(1,'(a132)') text
      n = index(text,'CONSTANTS')
      if (n .eq. 0) go to 200

      read(text(n+10:n+20),*) nprops

      if (nprops .gt. Max_props) then
        print*,' '
        print*,'STOP!!!  The number of mat properties in the input'
        print*,'         file exceeds the max value in the Max_props'
        print*,'         variable in the program.'
        Print*,' '
        Print*,'Set Max_props in program to at least: ',nprops+1,'.'
        Print*,' '
        STOP
      end if

c-----------------------------------------------------------------------
c  Read in material properties
c-----------------------------------------------------------------------

      do i = 1,nprops/8
        read(1,'(a132)') text
        read(text,*)(props(8*(i-1)+j),j=1,8)
      end do

      itest = nprops - nprops / 8 * 8

      print*,'itest = ',itest

      if (itest .gt. 0) then
        read(1,'(a132)') text
        read(text,*)(props(j),j=(nprops/8)*8+1,nprops)
      end if

c-----Convert electric conductivity and magnetostriction to true SI-

      props(59) = props(59)*1.0d06
      props(61) = props(61)*1.0d-06
      props(62) = props(62)*1.0d-06

c=======================================================================
c  Start new time step
c  Read in data
c=======================================================================

c     read(1,'(a132)') text
c     n = index(text,'MBC')

      read(1,'(a132)') text
      do i = 1,2
        if (text(1:1) .eq. mloading_key(1,i)) then
          mloading_mode = mloading_key(2,i)
        end if
      end do
```

```
      read(text(3:132),*)fmag,frate,bdir(1),bdir(2),bdir(3),
     & efield,dtempdx,d2tempdx2

      if (mloading_mode .eq. 'HField') then
       hmag = fmag/xmu0
       frate = frate/xmu0
       bmag = 0.0
       mload = 1
       print*,'HField load',hmag,frate,bdir(1),bdir(2),bdir(3)
      else if (mloading_mode .eq. 'BField') then
       hmag = 0.0
       bmag = fmag
       mload = 2
       print*,'BField load',bmag,frate,bdir(1),bdir(2),bdir(3)
      end if

  300 time(1) = 0.0
      Out_of_time = .false.

      read(1,'(a132)',END=999) text
      n = index(text,'STEP')
      if (n .eq. 0) go to 300

      read(1,'(a132)') text
      n = index(text,'**')
      if (n.eq.1) go to 300

      istep = istep + 1

      read(text,*)dtime,time_max,dtime_max


      print*,' '
      print*,' dtime    time_max    dtime_max'
      write(6,'(f7.4,f12.2,f11.2/)')dtime,time_max,dtime_max


c-----------------------------------------------------------------------
c  Read in loading mode
c-----------------------------------------------------------------------

  350 loading_mode = 'none'
      read(1,'(a132)') text
      do i = 1,Num_modes
        if (text(1:1) .eq. loading_key(1,i)) then
          loading_mode = loading_key(2,i)
        end if
      end do
      if (loading_mode .eq. 'none') go to 350

c-----------------------------------------------------------------------
c  Read in loading direction
c-----------------------------------------------------------------------

      read(text(2:2),*)i
      read(text(3:3),*)j

c-----------------------------------------------------------------------
```

```
c  Read in loading rate
c-----------------------------------------------------------------------

      if (loading_mode .eq. 'SimuComTors') then
        read(text(5:132),*)velocity2,velocity
      else if (loading_mode .eq. 'BiaxialTens') then
        read(text(5:132),*)velocity,velocity2
      else if (loading_mode .eq. 'PressShear') then
        read(text(5:132),*)velocity,velocity2
      else
        read(text(5:132),*)velocity
      end if


c=======================================================================
c  Set up loading directions
c=======================================================================
c-----------------------------------------------------------------------
c  Tension & Compression
c-----------------------------------------------------------------------

      if (loading_mode .eq. 'TensionComp') then

        print*,'Tension/Compression in the ',j,' direction.'
        Print*,' '

        if (i .ne. j) then
          Print*,'WARNING!!! Loading direction not clear in'
          Print*,'           input file.'
          Print*,' '
        end if
        k = j
        i = mod(k  ,3)+1
        j = mod(k+1,3)+1

c-----------------------------------------------------------------------
c  Torsion
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'Torsion') then

        print*,'Torsion in the ',i,' -',j,' plane.'
        Print*,' '

        if (i .eq. j) then
          Print*,'STOP!!! Loading plane must be two DIFFERENT'
          Print*,'        integers in input file.'
          Print*,' '
          STOP
        end if
        k = 6 - i - j

c-----------------------------------------------------------------------
c  Plane Strain Compression
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'PlaneStrain') then

        print*,'Plane-strain in the ',i,' -',j,' plane.'
```

```
        Print*,' '

        if (i .eq. j) then
          Print*,'STOP!!! Loading plane must be two DIFFERENT'
          Print*,'        integers in input file.'
          Print*,' '
          STOP
        end if
        k = 6 - i - j

c-----------------------------------------------------------------------
c  Simultaneous Compression & Torsion
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'SimuComTors') then

        print*,'Both Comp & Torsion in the ',i,' -',j,' plane.'
        Print*,' '

        if (i .eq. j) then
          Print*,'STOP!!! Loading plane must be two DIFFERENT'
          Print*,'        integers in input file.'
          Print*,' '
          STOP
        end if
        k = j
        j = 6 - i - k

c-----------------------------------------------------------------------
c  Biaxial Tension (and Compression)
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'BiaxialTens') then

        print*,'Biaxial tension in the ',i,' -',j,' plane.'
        Print*,' '

        if (i .eq. j) then
          Print*,'STOP!!! Loading plane must be two DIFFERENT'
          Print*,'        integers in input file.'
          Print*,' '
          STOP
        end if
        k = 6 - i - j

c-----------------------------------------------------------------------
c  Uniaxial strain
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'UniaxStrain') then

        print*,'Uniaxial strain loading in the ',j,' direction.'
        Print*,' '

        if (i .ne. j) then
          Print*,'WARNING!!! Loading direction not clear in'
          Print*,'           input file.'
          Print*,' '
        end if
```

```
        k = j
        i = mod(k  ,3)+1
        j = mod(k+1,3)+1

c-----------------------------------------------------------------------
c  Volumetric strain
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'Volumetric') then

        print*,'Spherical loading in all 3 directions.'
        Print*,' '

        if (i .ne. j) then
          Print*,'WARNING!!! Loading direction not clear in'
          Print*,'           input file.'
          Print*,' '
        end if
        k = j
        i = mod(k  ,3)+1
        j = mod(k+1,3)+1

c-----------------------------------------------------------------------
c  Pressure-shear
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'PressShear') then

        print*,'Pressure plus shear in the ',i,' -',j,' plane.'
        Print*,' '

        if (i .eq. j) then
          Print*,'STOP!!! Loading plane must be two DIFFERENT'
          Print*,'        integers in input file.'
          Print*,' '
          STOP
        end if
        k = 6 - i - j

      end if

c-----------------------------------------------------------------------
c   Write out initial state with null magnetic effects
c-----------------------------------------------------------------------

      if (istep.eq.1) then

      write(6,'(42a,38a)')' STEP     TIME     E_eff     SIG(1,1) ',
     & ' SIG(2,2)  SIG(3,3)  SIG(1,2)  SIG(1,3)  SIG(2,3) ',
     & ' F(1,1)    F(2,2)    F(3,3)    F(1,2)    F(2,3)    F(1,3)   ',
     & ' EPEFF    EPEFFD '

      write(6,'(i4,f10.4,f10.5,6(f10.3),3(f10.3),3(f10.3),2(f10.5))')
     & istep, 0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,1.0,
     & 0.0,0.0,0.0,0.0,0.0

      write(8,'(i4,f10.4,f10.5,6(f10.3),3(f10.3),3(f10.3),2(f10.5))')
     & istep, 0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,1.0,1.0,
     & 0.0,0.0,0.0,0.0,0.0
```

```
      e0 = props(67)*props(26)

      write(9,'(i4,f10.4,f10.5,f10.6,3(f10.3),f10.5,f10.1,e10.3,
     & f10.1,f10.5,e11.3,f10.3,f10.5,f10.5,f10.5)')
     & istep,0.0,0.0,1.0,0.0,0.0,props(67),0.0,0.0,0.0,props(2),
     & 0.0,0.0,0.0,0.0,0.0,0.0,0.0

       end if

c-----------------------------------------------------------------------
c  Call magumat to update stress with mag terms
c-----------------------------------------------------------------------

      call magumat (time,dtime,temp,dfgrd1,props,nprops,
     &              statev,nstatv,bmag,hmag,mload,bdir,
     &              efield,dtempdx,d2tempdx2)

c-----------------------------------------------------------------------
c  Call umat in order to get tangent stiffness matrix
c-----------------------------------------------------------------------

      call cumat(stress,  statev,  ddsdde,  sse,     spd,
     & scd,     rpl,     ddsddt,  drplde,  drpldt,
     & strain,  dstrain, time,    dtime,   temp,
     & dtemp,   predef,  dpred,   cmname,  ndi,
     & nshr,    ntens,   nstatv,  props,   nprops,
     & coords,  drot,    pnewdt,  celent,  dfgrd0,
     & dfgrd1,  noel,    npt,     layer,   kspt,
     & kstep,   kinc)

      do m = 1,nstatv
        statev_ref(m) = statev(m)
      end do

      do m = 1, 6
        stress0(m) = stress(m)
      end do

c-----------------------------------------------------------------------
c  Write initial state again with mag field but no deformation
c-----------------------------------------------------------------------

      if (istep.eq.1) then

      stressfac = props(58)
      xmfld = statev(19)*xmu0/stressfac
      xdelphi = statev(25)
      ttilde = 1.0

      pressure = -(1.0/3.0)*(stress(1)+stress(2)+stress(3))
      detF = determinant(dfgrd0)

      if (loading_mode .eq. 'TensionComp') then
        strn = dfgrd0(3,3)-1.0
c       strn = dlog(dfgrd0(3,3))
        strn = dabs(strn)
        strs = stress(3)
     &         + stressfac*bmag*bmag*bdir(3)*bdir(3)/xmu0
```

```
     &         - bmag*statev(19)*bdir(3)*bdir(3)
     &         - 0.5*stressfac*bmag*bmag/xmu0 + bmag*statev(19)
        strs = dabs(strs)*detF/dfgrd0(3,3)
        shr = 0.5*strs
      else if (loading_mode .eq. 'UniaxStrain') then
        strn = 1.0-detF
        strs = stress(3)
        strs = -strs/1000.0
        shr  = 0.5*dabs(stress(3)-stress(1))
      else if (loading_mode .eq. 'Volumetric') then
        strn = 1.0-detF
        strs = pressure
        strs = strs/1000.0
        shr  = -0.5*(stress(3)-stress(1))
      else if (loading_mode .eq. 'Torsion') then
        strn = dfgrd0(1,2)
        strs = stress(4)
        shr  = strs
        strs = pressure
      else if (loading_mode .eq. 'PressShear') then
        strn = dfgrd0(1,2)
        strs = stress(4)
        shr  = strs
        strs = pressure
      else
        strn = E_eff
        strs = pressure
        shr = 0.0
      end if

       if (abs(fmag).gt.1e-15) then

       write(9,'(i4,f10.4,f10.5,f10.6,3(f10.3),f10.5,f10.1,e10.3,
     & f10.1,f10.5,e11.3,f10.3,f10.5,f10.5,f10.5)')
     & istep,0.0,strn,detF,strs,shr,props(67),0.0,0.0,0.0,
     & props(2),0.0,0.0,0.0,bmag,xmfld,xdelphi

        end if

      end if

c=======================================================================
c  Increments one time step
c=======================================================================

  400 time(1) = time(1) + dtime
      time(2) = time(2) + dtime

c=======================================================================
c     Update the H/B field
c=======================================================================

      if (mload.eq.1) then
         hmag = hmag + frate * dtime
      else if (mload.eq.2) then
         bmag = bmag + frate * dtime
      end if

c-----------------------------------------------------------------------
```

```
c  Tension & Compression in the 'k' direction
c-----------------------------------------------------------------------

      if (loading_mode .eq. 'TensionComp') then
      dfgrd1(k,k) = dfgrd0(k,k) + velocity * dtime
      delta_Dkk = (dfgrd1(k,k) - dfgrd0(k,k)) / dfgrd1(k,k)
      delta_Dii = -(stress(i)+stress(j) + (ddsdde(i,k)+ddsdde(j,k))
     & * delta_Dkk) / (ddsdde(i,i) + ddsdde(i,j)
     & + ddsdde(j,i) + ddsdde(j,j))
      dfgrd1(i,i) = dfgrd0(i,i) / (1 - delta_Dii)
      dfgrd1(j,j) = dfgrd0(j,j) / (1 - delta_Dii)

c-----------------------------------------------------------------------
c  Torsion in the i-j plane
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'Torsion') then

      m = i + j + 1
      dfgrd1(i,j) = dfgrd0(i,j) + velocity * dtime
      delta_Dij = velocity * dtime / 2 / dfgrd1(i,i)
      delta_Djj = -(stress(j)+ddsdde(j,m)*delta_Dij) / ddsdde(j,j)
      dfgrd1(j,j) = dfgrd0(j,j) / (1 - delta_Djj)

c-----------------------------------------------------------------------
c  Plane Strain Tension/Compression in the i-j plane
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'PlaneStrain') then

      dfgrd1(j,j) = dfgrd0(j,j) + velocity * dtime
      delta_Djj = (dfgrd1(j,j) - dfgrd0(j,j)) / dfgrd1(j,j)
      delta_Dii = -(stress(i)+ddsdde(i,j)*delta_Djj) / ddsdde(i,i)
      dfgrd1(i,i) = dfgrd0(i,i) / (1 - delta_Dii)

c-----------------------------------------------------------------------
c  Simultaneous Compress & Torsion
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'SimuComTors') then

      m = i + k + 1
      dfgrd1(k,k) = dfgrd0(k,k) + velocity * dtime
      dfgrd1(i,k) = dfgrd0(i,k) + velocity2* dtime
      delta_Dkk = (dfgrd1(k,k) - dfgrd0(k,k)) / dfgrd1(k,k)
      delta_Dik = velocity2 * dtime / 2 / dfgrd1(i,i)

      det = ddsdde(i,i) * ddsdde(j,j) - ddsdde(i,j) * ddsdde(j,i)
      b1 = -stress(i) - ddsdde(i,k) * delta_Dkk
     & - ddsdde(i,m) * delta_Dik
      b2 = -stress(j) - ddsdde(j,k) * delta_Dkk
     & - ddsdde(j,m) * delta_Dik
      delta_Dii = (b1 * ddsdde(j,j) - b2 * ddsdde(i,j)) / det
      delta_Djj = (ddsdde(i,i) * b2 - ddsdde(j,i) * b1) / det
      dfgrd1(i,i) = dfgrd0(i,i) / (1 - delta_Dii)
      dfgrd1(j,j) = dfgrd0(j,j) / (1 - delta_Djj)

c-----------------------------------------------------------------------
c  Biaxial Tension in the i - j plane !updated, 6/9/14
```

```
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'BiaxialTens') then

      dfgrd1(i,i) = dfgrd0(i,i) + velocity * dtime
      dfgrd1(j,j) = dfgrd0(j,j) + velocity2* dtime
      delta_Dii = (dfgrd1(i,i) - dfgrd0(i,i)) / dfgrd1(i,i)
      delta_Djj = (dfgrd1(j,j) - dfgrd0(j,j)) / dfgrd1(j,j)
      delta_Dkk = (stress(k) - ddsdde(k,i) * delta_Dii
     & - ddsdde(k,j) * delta_Djj)
     & / ddsdde(k,k)

      dfgrd1(k,k) = dfgrd0(k,k) / (1 - delta_Dkk)

c-----------------------------------------------------------------------
c  Uniaxial straining in the 'k' direction
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'UniaxStrain') then
      dfgrd1(k,k) = dfgrd0(k,k) + velocity * dtime
      dfgrd1(i,i) = dfgrd0(i,i)
      dfgrd1(j,j) = dfgrd0(j,j)

c-----------------------------------------------------------------------
c  Spherical expansion or compression
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'Volumetric') then
      dfgrd1(k,k) = dfgrd0(k,k) + velocity * dtime/3.0
      dfgrd1(i,i) = dfgrd0(i,i) + velocity * dtime/3.0
      dfgrd1(j,j) = dfgrd0(j,j) + velocity * dtime/3.0

c-----------------------------------------------------------------------
c  Pressure plus shear in the i-j plane
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'PressShear') then

      dfgrd1(i,j) = dfgrd0(i,j) + velocity * dtime
      dfgrd1(k,k) = dfgrd0(k,k) + velocity2 * dtime/3.0
      dfgrd1(i,i) = dfgrd0(i,i) + velocity2 * dtime/3.0
      dfgrd1(j,j) = dfgrd0(j,j) + velocity2 * dtime/3.0

      end if

c=======================================================================
c  Start new convergence iteration.  Restart with time-step
c  cut in half if necessary. Note that for the re-compute only the
c  strain in the non-specified direction is changed.
c=======================================================================
c     print*, 'dtime: ', dtime
      Kinc = 0
  500 if (Kinc .ge. Max_itr) then
        time(1) = time(1) - dtime
        time(2) = time(2) - dtime
        dtime = dtime / 2

        if (abs(dtime).le.1d-15) then !updated, 6/9/14
          print*, 'ERROR: DTIME IS LESS THAN 1D-15, EXITING'
```

```
        go to 999
       end if

       go to 400
      end if
      Kinc = Kinc + 1

c-----------------------------------------------------------------------
c  Calculate F_dot * dtime
c-----------------------------------------------------------------------

      do m = 1,3
       do n = 1,3
          array1(m,n) = dfgrd1(m,n) - dfgrd0(m,n)
          array3(m,n) =(dfgrd1(m,n) + dfgrd0(m,n))/2
       end do
      end do

c-----------------------------------------------------------------------
c  multiply F_dot * F_inverse * dtime
c-----------------------------------------------------------------------

      call Xinverse_3x3(array3,array2)
      call Xaa_dot_bb(3,array1,array2,array3)

c-----------------------------------------------------------------------
c  Get D_dt and W_dt
c-----------------------------------------------------------------------

      do m = 1,3
       do n = 1,3
         D_dt(m,n) = (array3(m,n) + array3(n,m)) / 2
         W_dt(m,n) = (array3(m,n) - array3(n,m)) / 2
       end do
      end do

c-----------------------------------------------------------------------
c  Store D_dt in dstrain
c-----------------------------------------------------------------------

      dstrain(1) = D_dt(1,1)
      dstrain(2) = D_dt(2,2)
      dstrain(3) = D_dt(3,3)
      dstrain(4) = D_dt(1,2) * 2
      dstrain(5) = D_dt(1,3) * 2
      dstrain(6) = D_dt(2,3) * 2

c-----------------------------------------------------------------------
c  Convert spin to drot(i,j) array for the UMAT
c-----------------------------------------------------------------------

c      call Xspin_to_matrix(W_dt,drot)

       do m = 1,3
        do n = 1,3
          drot(m,n) = W_dt(m,n)     ! JDC change
        end do
       end do
```

```
c-----------------------------------------------------------------------
c  Call magumat to update stress with mag terms
c-----------------------------------------------------------------------

      call magumat (time,dtime,temp,dfgrd1,props,nprops,
     &              statev,nstatv,bmag,hmag,mload,bdir,
     &              efield,dtempdx,d2tempdx2)

c-----------------------------------------------------------------------
c  Call umat in order to get stress
c-----------------------------------------------------------------------

      call cumat(stress,  statev,  ddsdde,  sse,     spd,
     & scd,     rpl,     ddsddt,  drplde,  drpldt,
     & strain,  dstrain, time,    dtime,   temp,
     & dtemp,   predef,  dpred,   cmname,  ndi,
     & nshr,    ntens,   nstatv,  props,   nprops,
     & coords,  drot,    pnewdt,  celent,  dfgrd0,
     & dfgrd1,  noel,    npt,     layer,   kspt,
     & kstep,   kinc)

c-----------------------------------------------------------------------
c  Check to see if need to iterate
c  Tension & Compression are immediately below
c-----------------------------------------------------------------------

      if (loading_mode .eq. 'TensionComp') then

        test = abs(stress(i)+stress(j))/abs(stress(k))

      if (test .gt. tolerance) then
        delta_Dii = -(stress(i)+stress(j))/(ddsdde(i,i)+ddsdde(i,j)
     &    + ddsdde(j,i) + ddsdde(j,j))
        dfgrd1(i,i) = dfgrd1(i,i) / (1 - delta_Dii)
        dfgrd1(j,j) = dfgrd1(j,j) / (1 - delta_Dii)
        do m = 1,nstatv
          statev(m) = statev_ref(m)
        end do
        do m = 1,6
          stress(m) = stress0(m) !updated, 6/9/14
        end do
        go to 500
      end if

c-----------------------------------------------------------------------
c  Torsion
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'Torsion') then

      m = i + j + 1
      test = abs(stress(j)/stress(m))

      if (test .gt. tolerance) then
        delta_Djj = -stress(j) / ddsdde(j,j)
        dfgrd1(j,j) = dfgrd1(j,j) / (1 - delta_Djj)
        do m = 1,nstatv
          statev(m) = statev_ref(m)
        end do
```

```
        do m = 1,6
          stress(m) = stress0(m) !updated, 6/9/14
        end do
        go to 500
      end if

c-----------------------------------------------------------------------
c  Plane Strain Compresssion
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'PlaneStrain') then

      test = abs(stress(i)/stress(j))

      if (test .gt. tolerance) then
        delta_Dii = -stress(i) / ddsdde(i,i)
        dfgrd1(i,i) = dfgrd1(i,i) / (1 - delta_Dii)
        do m = 1,nstatv
          statev(m) = statev_ref(m)
        end do
        do m = 1,6
          stress(m) = stress0(m) !updated, 6/9/14
        end do
        go to 500
      end if

c-----------------------------------------------------------------------
c  Simultaneous Compress & Torsion
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'SimuComTors') then

      m = i + k + 1
      test = (abs(stress(i)) + abs(stress(j))) /
     & (abs(stress(k)) + abs(stress(m)))

      if (test .gt. tolerance) then
        det = ddsdde(i,i) * ddsdde(j,j) - ddsdde(i,j) * ddsdde(j,i)
        b1 = -stress(i)
        b2 = -stress(j)
        delta_Dii = (b1 * ddsdde(j,j) - b2 * ddsdde(i,j)) / det
        delta_Djj = (ddsdde(i,i) * b2 - ddsdde(j,i) * b1) / det
        dfgrd1(i,i) = dfgrd1(i,i) / (1 - delta_Dii)
        dfgrd1(j,j) = dfgrd1(j,j) / (1 - delta_Djj)
        do m = 1,nstatv
          statev(m) = statev_ref(m)
        end do
        do m = 1,6
          stress(m) = stress0(m) !updated, 6/9/14
        end do
        go to 500
      end if

c-----------------------------------------------------------------------
c  Biaxial Tension
c-----------------------------------------------------------------------

      else if (loading_mode .eq. 'BiaxialTens') then
```

```
      test = 2.d0 * abs(stress(k)) / (abs(stress(i)) + abs(stress(j)))

      if (test .gt. tolerance) then
        delta_Dkk = -stress(k) / ddsdde(k,k)
        dfgrd1(k,k) = dfgrd1(k,k) / (1.d0 - delta_Dkk)
        do m = 1,nstatv
          statev(m) = statev_ref(m)
        end do
        do m = 1,6
          stress(m) = stress0(m) !updated, 6/9/14
        end do
        go to 500
      end if

      end if

c=====================================================================
c  Finished increment!
c  Calc effective delta strain and add it to E_eff
c=====================================================================

      sum = 0
      do m = 1,3
        do n = 1,3
          sum = sum + D_dt(m,n) * D_dt(m,n)
        end do
      end do
      dE_eff = sqrt(2. * sum / 3.)
      E_eff = E_eff + dE_eff

c---------------------------------------------------------------------
c  Update strain gradient
c---------------------------------------------------------------------

      do m = 1,6
        strain(m) = strain(m) + dstrain(m)
      end do

c---------------------------------------------------------------------
c  Update stress
c---------------------------------------------------------------------

      do m = 1,6
        stress0(m) = stress(m)
      end do

c---------------------------------------------------------------------
c  Update deformation gradient
c---------------------------------------------------------------------

      do m = 1,3
        do n = 1,3
          dfgrd0(m,n) = dfgrd1(m,n)
        end do
      end do

c---------------------------------------------------------------------
c  Update statev_ref
c---------------------------------------------------------------------
```

```
      do m = 1,nstatv
        statev_ref(m) = statev(m)
      end do

c-----------------------------------------------------------------------
c  Write out results
c-----------------------------------------------------------------------

      write(6,'(i4,f10.4,f10.5,6(f10.2),3(f10.5),3(f10.5),2(e10.3))')
     & istep, time(2),E_eff,(stress(m),m=1,6), (dfgrd1(n,n),n=1,3),
     & dfgrd1(1,2), dfgrd1(2,3), dfgrd1(1,3), statev(1), statev(2)


      write(8,'(i4,f10.4,f10.5,6(f10.2),3(f10.5),3(f10.5),2(e10.3))')
     & istep, time(2),E_eff,(stress(m),m=1,6), (dfgrd1(n,n),n=1,3),
     & dfgrd1(1,2), dfgrd1(2,3), dfgrd1(1,3), statev(1), statev(2)

      pressure = -(1.0/3.0)*(stress(1)+stress(2)+stress(3))
      detF = determinant(dfgrd0)

      if (loading_mode .eq. 'TensionComp') then
        strn = dfgrd1(3,3)-1.0
c       strn = dlog(dfgrd1(3,3))
        strn = dabs(strn)
        strs = stress(3)
     &         + stressfac*bmag*bmag*bdir(3)*bdir(3)/xmu0
     &         - bmag*statev(19)*bdir(3)*bdir(3)
     &         - 0.5*stressfac*bmag*bmag/xmu0 + bmag*statev(19)
c       strs = dabs(strs)
        strs = dabs(strs)*detF/dfgrd1(3,3)
        shr = 0.5*strs
      else if (loading_mode .eq. 'UniaxStrain') then
        strn = 1.0-detF
        strs = stress(3)
        strs = -strs/1000.0
        shr  = 0.5*dabs(stress(3)-stress(1))
      else if (loading_mode .eq. 'Volumetric') then
        strn = 1.0-detF
        strs = pressure
        strs = strs/1000.0
        shr  = -0.5*(stress(3)-stress(1))
      else if (loading_mode .eq. 'Torsion') then
        strn = dfgrd1(1,2)
        strs = stress(4)
        shr  = strs
        strs = pressure
      else if (loading_mode .eq. 'PressShear') then
        strn = dfgrd1(1,2)
        strs = stress(4)
        shr  = strs
        strs = pressure
      else
        strn = E_eff
        strs = pressure
        shr = 0.0
      end if

      stressfac = props(58)
```

```
      xmfld = statev(19)*xmu0/stressfac
      xdelphi = statev(25)

      write(9,'(i4,f10.4,f10.5,f10.6,f10.3,2(f10.3),f10.5,f10.1,
     & e10.3,f10.1,f10.5,e11.3,f10.3,f10.5,f10.5,f10.5)')
     & istep,time(2),strn,detF,strs,shr,(statev(m),m=4,11),
     & bmag,xmfld,xdelphi

c-----------------------------------------------------------------------
c  If not out of time, then loop back up
c-----------------------------------------------------------------------

      if (.not.Out_of_time) then
        if (time(1).ne.dtime) dtime = dtime * 1.5
        if (dtime .gt. dtime_max) dtime = dtime_max
        if (time_max-time(1).lt.dtime) then
          dtime = time_max - time(1)
          Out_of_time = .true.
        end if
        go to 400
      end if

c-----------------------------------------------------------------------
c  Loop back up and check for another STEP
c-----------------------------------------------------------------------

      go to 300

c-----------------------------------------------------------------------
c  THE END!!!
c-----------------------------------------------------------------------

  999 continue

      close(1)
      close(7)
      close(8)
      close(9)
      stop
      end

c=======================================================================
c=======================================================================
c
c  Calculate the dot product of two 2nd rank tensors.
c  Result is stored in cc(i,j)
c
c-----------------------------------------------------------------------

      subroutine Xaa_dot_bb(n,a,b,c)

      implicit double precision (a-h,o-z)

      dimension a(n,n), b(n,n), c(n,n)

      do i = 1,n
        do j = 1,n
          c(i,j) = 0
          do k = 1,n
```

```
            c(i,j) = c(i,j) + a(i,k) * b(k,j)
          end do
        end do
      end do

      return
      end

c=======================================================================
c=======================================================================
c
c  Calculate the inverse of a 3 x 3 matrix.
c
c-----------------------------------------------------------------------

      subroutine Xinverse_3x3(a,b)

      implicit double precision (a-h,o-z)

      dimension a(3,3), b(3,3)

      b(1,1) = a(2,2) * a(3,3) - a(3,2) * a(2,3)
      b(1,2) = a(3,2) * a(1,3) - a(1,2) * a(3,3)
      b(1,3) = a(1,2) * a(2,3) - a(2,2) * a(1,3)
      b(2,1) = a(3,1) * a(2,3) - a(2,1) * a(3,3)
      b(2,2) = a(1,1) * a(3,3) - a(3,1) * a(1,3)
      b(2,3) = a(2,1) * a(1,3) - a(1,1) * a(2,3)
      b(3,1) = a(2,1) * a(3,2) - a(3,1) * a(2,2)
      b(3,2) = a(3,1) * a(1,2) - a(1,1) * a(3,2)
      b(3,3) = a(1,1) * a(2,2) - a(2,1) * a(1,2)

      det = a(1,1) * b(1,1) + a(1,2) * b(2,1) + a(1,3) * b(3,1)

      do i = 1,3
        do j = 1,3
          b(i,j) = b(i,j) / det
        end do
      end do

      return
      end

c=======================================================================
c=======================================================================
c
c  Convert a spin tensor to a rotation matrix
c
c-----------------------------------------------------------------------

      subroutine Xspin_to_matrix(a,b)

      implicit double precision (a-h,o-z)

      dimension a(3,3), b(3,3)

      Pi = 4*atan(1D0)

c-----------------------------------------------------------------------
c  Store spin tensor, a(i,j), as a rotation vector
```

```
c-----------------------------------------------------------------------

      p1 = a(3,2)
      p2 = a(1,3)
      p3 = a(2,1)
      ang = sqrt(p1*p1+p2*p2+p3*p3)

      s = sin(ang)
      c = cos(ang)

c-----------------------------------------------------------------------
c  Normalize vector
c-----------------------------------------------------------------------

      if (ang .le. 1.D-300) then
        p1 = 0
        p2 = 0
        p3 = 1
      else
        p1 = p1 / ang
        p2 = p2 / ang
        p3 = p3 / ang
      end if

c-----------------------------------------------------------------------
c  Calculate rotation matrix
c-----------------------------------------------------------------------

      b(1,1) = c + (1 - c) * p1 * p1
      b(1,2) = (1 - c) * p1 * p2 - s * p3
      b(1,3) = (1 - c) * p1 * p3 + s * p2
      b(2,1) = (1 - c) * p2 * p1 + s * p3
      b(2,2) = c + (1 - c) * p2 * p2
      b(2,3) = (1 - c) * p2 * p3 - s * p1
      b(3,1) = (1 - c) * p3 * p1 - s * p2
      b(3,2) = (1 - c) * p3 * p2 + s * p1
      b(3,3) = c + (1 - c) * p3 * p3

      return
      end
```

List of Symbols, Abbreviations, and Acronyms

TERMS:

ARL	Army Research Laboratory
BCC	body centered cubic
BCT	body centered tetragonal
CMF	constitutive model framework
DAC	diamond anvil cell
DEVCOM	US Army Combat Capabilities Development Command
EBSD	electron backscatter diffraction
EOS	equation of state
FCC	face centered cubic
Fe	iron
FE	finite element
HCP	hexagonal close packed
CMF	constitutive model framework
Mn	manganese
MAGUMAT	magnetic user material subroutine
MPS	material point simulator
SFE	stacking fault energy
SLIP	slip dominated plasticity
TRIP	transformation induced plasticity
TWIP	twinning induced plasticity
UMAT	user material subroutine

MATHEMATICAL SYMBOLS:

$\boldsymbol{B}$	magnetic flux density

$\boldsymbol{F}$	deformation gradient
$\mathbb{G}$	Gibbs free energy
$\boldsymbol{H}$	magnetic field
$\boldsymbol{M}$	magnetization per unit spatial volume
p	Cauchy pressure
t	time
U	internal energy
$\boldsymbol{x}$	spatial coordinates
$\boldsymbol{X}$	reference coordinates
ρ	mass density
$\boldsymbol{\sigma}$	symmetrized Cauchy stress tensor
$\boldsymbol{v}$	particle velocity
ξ	transformed phase volume fraction
θ	temperature
ψ	Helmholtz free energy

1 (PDF) DEFENSE TECHNICAL INFORMATION CTR
DTIC OCA

1 (PDF) DEVCOM ARL
FCDD RLB CI
TECH LIB

1 (PDF) LAWRENCE LIVERMORE NAT LAB
R AUSTIN

1 (PDF) SANDIA NAT LAB
J NIEDERHAUS

1 (PDF) UNIV WISCONSIN
C BRONKHORST

32 (PDF) DEVCOM ARL
FCDD RLA
J CIEZAK
S SCHOENFELD
FCDD RLA B
R BECKER
A TONGE
FCDD RLA HC
J CRONE
J KNAP
FCDD RLA MB
D MAGAGNOSC
G GAZONAS
Z WILSON
FCDD RLA MD
S CLUFF
FCDD RLA MF
D FIELD
C HAINES
K LIMMER
H MURDOCH
FCDD RLA T
M FERMEN-COKER
FCDD RLA TA
M COPPINGER
W UHLIG
C WILLIAMS
FCDD RLA TB
D CASEM
J CLAYTON
B FAGAN
D KRAYTERMAN
J MCDONALD
P MCKEE
C MEREDITH
R REGUEIRO
S SATAPATHY
FCDD RLA TE
J LLOYD
T SCHARF
FCDD RLA TF
J CAZAMIAS
R LEAVY
D MALLICK