Computational Materials Science 169 (2019) 109078

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

PRISMS-Plasticity: An open-source crystal plasticity finite element software

Mohammadreza Yaghoobi[a,*], Sriram Ganesan[b], Srihari Sundar[b], Aaditya Lakshmanan[b], Shiva Rudraraju[c,1], John E. Allison[a], Veera Sundararaghavan[b]

[a] Materials Science and Engineering, University of Michigan, Ann Arbor, MI 48109, USA
[b] Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA
[c] Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA

ARTICLE INFO

Keywords:
Crystal plasticity finite element
Open source software
Parallel performance
Twinning

ABSTRACT

An open-source parallel 3-D crystal plasticity finite element (CPFE) software package *PRISMS-Plasticity* is presented here as a part of PRISMS integrated framework. A highly efficient rate-independent crystal plasticity algorithm is implemented along with developing its algorithmic tangent modulus. Additionally, a twin activation mechanism is incorporated into the framework based on an integration point sensitive scheme. The integration of the software as a part of the PRISMS framework is demonstrated. To do so, the integration of the *PRISMS-Plasticity* software with experimental characterization techniques such as electron backscatter diffraction (EBSD) and synchrotron X-ray diffraction using available open source software packages of DREAM.3D and Neper is elaborated. The integration of the *PRISMS-Plasticity* software with the information repository of Materials Commons is also presented. The parallel performance of the software is characterized which demonstrates that it scales well for large problems running on hundreds of processors. Various examples of polycrystalline metals with face-centered cubic (FCC), body-centered cubic (BCC), and hexagonal close-packed (HCP) crystals structures are presented to show the capability of the software to efficiently solve crystal plasticity boundary value problems, in addition to integration with preprocessing and postprocessing tools. *PRISMS-Plasticity* is an important activity within the broader PRISMS Center and future enhancements to *PRISMS-Plasticity* are planned and will be described.

1. Introduction

The integrated computational materials engineering (ICME) approach can effectively link manufacturing and design using advanced materials models which can greatly accelerate materials development and reduce development costs [1–3]. One of the tools that has been effectively incorporated into the ICME approach is the crystal plasticity finite element (CPFE) [4]. CPFE has been used to solve very important problems in polycrystalline metallic samples including grain fragmentation, texture evolution, evolution of dislocation density, recrystallization, damage initiation and propagation, twinning, and grain size effects [5]. The CPFE method is the integration of crystal plasticity models with finite element (FE) implementation.

Crystal plasticity models can capture the behavior of single crystalline grain. In the case of polycrystalline materials, various methods have been developed to extract the homogenized behavior of the sample made up from grains with different orientations. These methods include the original Taylor model [6], modified Taylor model including relaxation [7–9], cluster models such as LAMEL [10–12], grain interaction (GIA) [13–15], and relaxed grain cluster (RGC) [16], spectral finite elements schemes using fast Fourier transform [17,18], and full-field homogenization using crystal plasticity finite element (CPFE) (see, e.g., [5]). Similar to any FE implementation, the system response is obtained by discretizing the weak form of the governing equations, which are the equilibrium equations for CPFE. The constitutive model of the system at each point is governed using crystal plasticity by considering the texture, i.e., grain orientation, of that point. Unlike other crystal plasticity homogenization methods, CPFE can accurately handle general boundary conditions in a straight forward manner. Furthermore, CPFE can capture the local fields of stress and strain in addition to the homogenized behavior of the sample.

Although the CPFE method can accurately capture many microstructural features and homogenized properties of polycrystalline samples, it can become computationally expensive. It is thus essential to develop parallel CPFE codes which can scale well with increasing problem size and processing power (CPU cores). In this work, an open-

[*] Corresponding author.
E-mail address: yaghoobi@umich.edu (M. Yaghoobi).
[1] Present Address: Mechanical Engineering, University of Wisconsin-Madison, WI 53706, USA.

https://doi.org/10.1016/j.commatsci.2019.109078
Received 5 May 2019; Received in revised form 5 June 2019; Accepted 11 June 2019
Available online 27 June 2019

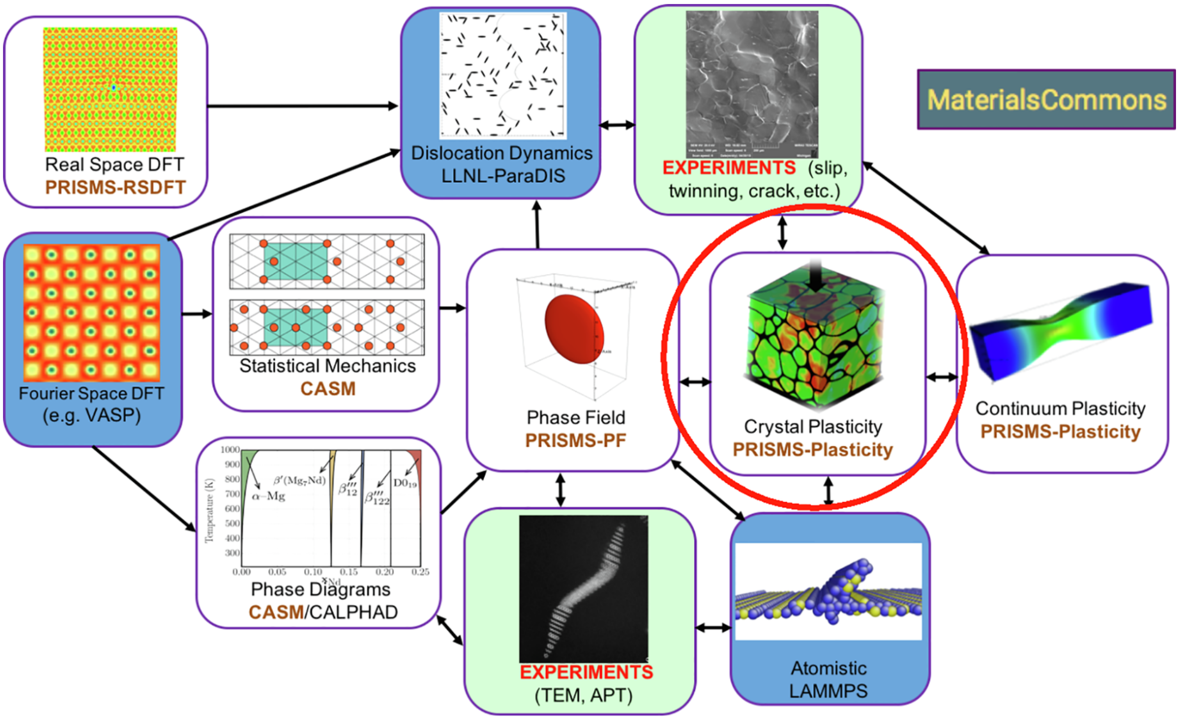


Fig. 1. PRISMS-Plasticity is one of the primary open-source codes that are part of the PRISMS Integrated Framework. White: PRISMS Software, Green: PRISMS Experiments, Blue: Existing External Software (After Aagesen et al. [4]).

source, parallel, scalable, 3-D crystal plasticity finite element (CPFE) software package called *PRISMS-Plasticity* is described. The software is implemented on top of the deal.II [19] finite element framework. The formulation of the rate-independent crystal plasticity model which incorporates deformation twinning is described. A derivation of the algorithmic tangent modulus is presented to numerically integrate the crystal plasticity model in the FE framework using the Newton-Raphson iterative procedure. Problem sizes of up to 2 million elements and 6 million degrees of freedom are solved to demonstrate the speed and efficiency of the software.

One of the primary targets of computational material science tools is to design a material with targeted behavior. To do so, one should be able to vary the microstructural features of the material to obtain the target mechanical behavior. Accordingly, an integrated PRISMS design framework shown in Fig. 1 [4] has been developed within the PRISMS Center that links microstructural features and properties of materials to their mechanical behavior. *PRISMS-Plasticity* plays an important role in this integrated framework. The initial step is the generation of a representative volume element (RVE) which can mimic the main characteristics of experimental characterization methods. This can be accomplished using microstructural evolution codes such as the *PRISMS-PF* phase field code that is included in this framework, or from experimental characterization of microstructures. Next, the information passes to the *PRISMS-Plasticity* CPFE software to simulate the mechanical response of this RVE. The efficient integration between the sample generation tools and CPFE software is required to provide a material design framework. In the current work, we describe the integration of the *PRISMS-Plasticity* software into the integrated PRISMS framework (Fig. 1) coupled with experimental characterization methods such as electron backscatter diffraction (EBSD) and synchrotron X-ray diffraction. In the current case, we describe the use of readily available open source software packages, DREAM.3D [20] and Neper [21]. Accordingly, a specific set of filters are provided to make an automated linkage between DREAM.3D [20] and Neper [21] to *PRISMS-Plasticity* software.

The current pace of materials discovery and development requires new approaches and means of collaborating and sharing information. With the PRISMS integrated framework, the Materials Commons [22] serves the function of being a collaboration platform and permanent information repository for structural materials community. The Materials Commons, which has been described in Puchala et al. [22], is being developed to be a continuous and seamless part of the scientific workflow process. The integration of the *PRISMS-Plasticity* software with Materials Commons is presented here. The Materials Commons Python Application Programming Interfaces (MCAPI) wrapper (prismscpfe_mcapi) is used to automate the integration of *PRISMS-Plasticity* CPFE simulations into the Materials Commons.

A set of examples for polycrystalline samples with face-centered cubic (FCC), body-centered cubic (BCC), and hexagonal close packed (HCP) crystal structures are presented to demonstrate different aspects of the *PRISMS-Plasticity* CPFE software. The initial samples are generated using the software packages of DREAM.3D [20] and Neper [21]. The results obtained are compared against those from the literature to validate the code. Different post-processing results are also presented to demonstrate the capability of the software in providing various types of outputs.

One of the important aspects of a parallel code is the capability to scale to a large number of degrees of freedom using multiple processors while solving problems in practicable times. In order to demonstrate the parallel scalability of *PRISMS-Plasticity* CPFE software, strong scaling and weak scaling studies were conducted. The strong scaling study focuses on the variation of the simulation time (wall time) for a fixed problem size as the number of processors increases. The weak scaling study focuses on the variation of simulation time (wall time) for a fixed number of elements per processor as the problem size increases. Finally, planned enhancements and ongoing research using the *PRISMS-Plasticity* CPFE code are outlined.

2. Crystal plasticity model

2.1. Constitutive model

A rate-independent single crystal plasticity theory developed by Anand and Kothari [23] is incorporated in the *PRISMS-Plasticity* CPFE software to define the constitutive model of each grain. The constitutive

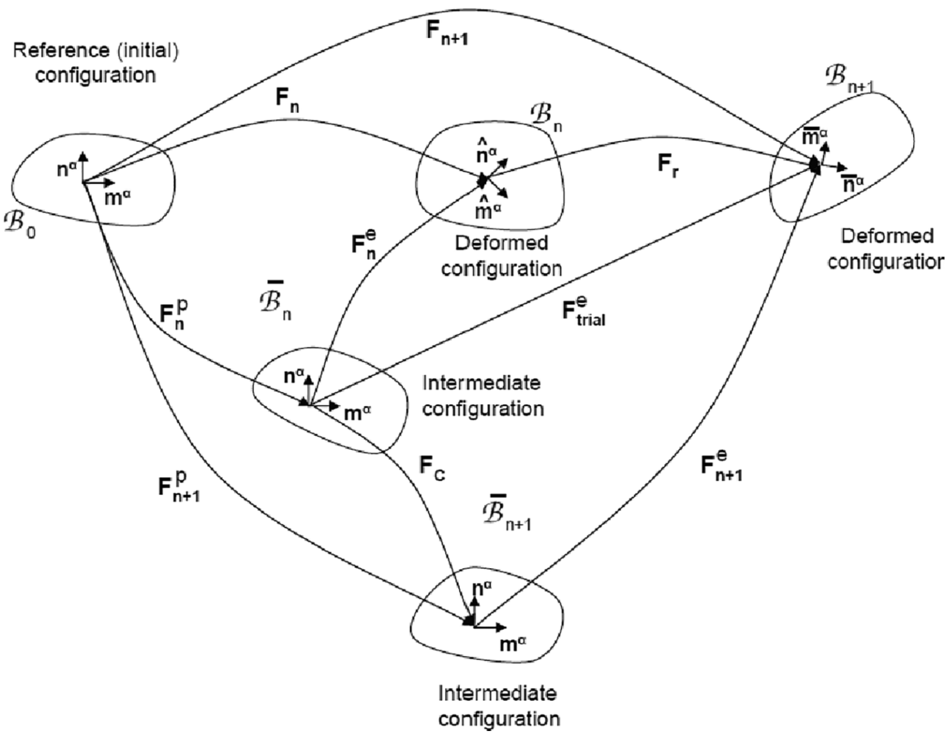


Fig. 2. Schematic of the various material configurations, for a single crystal, used in the integration of the constitutive model. The slip systems ($\boldsymbol{m}^{\alpha}$, $\boldsymbol{n}^{\alpha}$ are known on the reference (initial) configuration. Also, $\hat{\boldsymbol{m}}^{\alpha}$, $\bar{\boldsymbol{m}}^{\alpha}$ are the slip directions (different from $\boldsymbol{m}^{\alpha}$ because of crystal re-orientation) in the deformed configurations ß$_{\text{n}}$ and ß$_{\text{n}+1}$, respectively.

model has been successfully incorprotaed to capture the response of polycrystalline samples with FCC (Sundararaghavan and Zabaras [24]), BCC (Kumar and Sundararaghavan [25]), and HCP (Acar et al. [26]) crystal structures. The theory is based on the notion that plastic flow is accommodated through slip on prescribed slip systems. The finite deformation continuum mechanics framework is adopted. Accordingly, a multiplicative decomposition of the deformation gradient tensor $\mathbf{F}$ into its elastic and plastic components, i.e., $\mathbf{F}^{\text{e}}$ and $\mathbf{F}^{\text{p}}$, is assumed here as follows:

$$\mathbf{F} = \mathbf{F}^{\text{e}}\mathbf{F}^{\text{p}} \tag{1}$$

The kinematics of single crystal slip is illustrated in Fig. 2. In the current crystal plasticity framework, two independent deformation mechanisms of elastic distortion of the crystal lattice and pure shear as a result of plastic slip – accommodate the applied deformation. The macroscopic velocity gradient tensor $\mathbf{L}$ can be additively decomposed into the elastic and plastic components, i.e., $\mathbf{L}^{e}$ and $\mathbf{L}^{\text{p}}$, respectively, as follows:

$$\mathbf{L} = \mathbf{L}^{e} + \mathbf{L}^{p} \tag{2}$$

The key idea of crystal plasticity is to link the plastic part of the velocity gradient, as a representative of macroscopic response, to the superposition of shear deformation induced by crystallographic slip on multiple slip, as a microscopic mechanisms of deformation, which can be described as follows:

$$\mathbf{L}^{\text{p}} = \sum_{\alpha} \dot{\gamma}^{\alpha}\mathbf{S}^{\alpha}\ \text{sign}(\tau^{\alpha}) \tag{3}$$

where $\mathbf{L}^{\text{p}}$ is the plastic velocity gradient which denotes the macroscopic rate of plastic distortion, $\dot{\gamma}^{\alpha}$ is the shearing rate on slip system α, τ^{α} is the resolved shear stress acting on slip system α, and $\mathbf{S}^{\alpha}$ is the Schmid tensor for the slip system α, which can be defined as follows:

$$\mathbf{S}^{\alpha} = \mathbf{m}^{\alpha} \otimes\ \mathbf{n}^{\alpha} \tag{4}$$

where unit vectors $\mathbf{m}^{\alpha}$ and $\mathbf{n}^{\alpha}$ denote the slip direction and slip plane normal, respectively, in the deformed configuration.

The resolved shear stress on the slip system α can be obtained as follows:

$$\tau^{\alpha} = \sigma{:}\ \mathbf{S}^{\alpha} \tag{5}$$

where σ is the Cauchy stress tensor and $\cdot$ operator denotes the standard inner product of tensors.

In the framework of rate-independent algorithm, the yield surface of slip system α can be defined as follows:

$$f^{\alpha} = |\tau^{\alpha}| - s^{\alpha} \tag{6}$$

where s^{α} is the slip resistance on slip system α. For slip to take place on the slip system, $|\tau^{\alpha}|$ must reach a critical value of s^{α} (Fig. 3). The inactive slip systems, i.e., $\dot{\gamma}^{\alpha} = 0$, are those with $|\tau^{\alpha}| < s^{\alpha}$, or $|\tau^{\alpha}| = s^{\alpha}$ and the trial stress rate points to the inside of the yield surface. If $|\tau^{\alpha}| = s^{\alpha}$ and the trial stress rate points to the outside of the yield surface, the slip system is potentially active with $\dot{\gamma}^{\alpha} \geq 0$. These conditions are the specialization of the Kuhn-Tucker consistency conditions in computational plasticity and more generally in mathematical optimization.

The evolution of slip resistance for slip system α can be obtained as follows:

$$\dot{s}^{\alpha} = \sum_{\beta} h^{\alpha\beta}\dot{\gamma}^{\beta} \tag{7}$$

where $h^{\alpha\beta}$, which is commonly termed as the hardening moduli, defines the variation of slip resistance for slip system α due to the slip rate on slip system β. The hardening moduli $h^{\alpha\beta}$ can be described as a power-law relationship considering the combined effect of work hardening and recovery as follows:

$$h^{\alpha\beta} = \begin{cases} h_0^{\beta}\left[1 - \frac{s^{\beta}}{s_s^{\beta}}\right]^{a^{\beta}} & \text{if}\,\alpha = \beta\ \text{(coplanar systems)} \\ h_0^{\beta}q\left[1 - \frac{s^{\beta}}{s_s^{\beta}}\right]^{a^{\beta}} & \text{if}\,\alpha \neq \beta \end{cases} \tag{8}$$

where q is the latent hardening ratio, h_0^{β} denotes the hardening

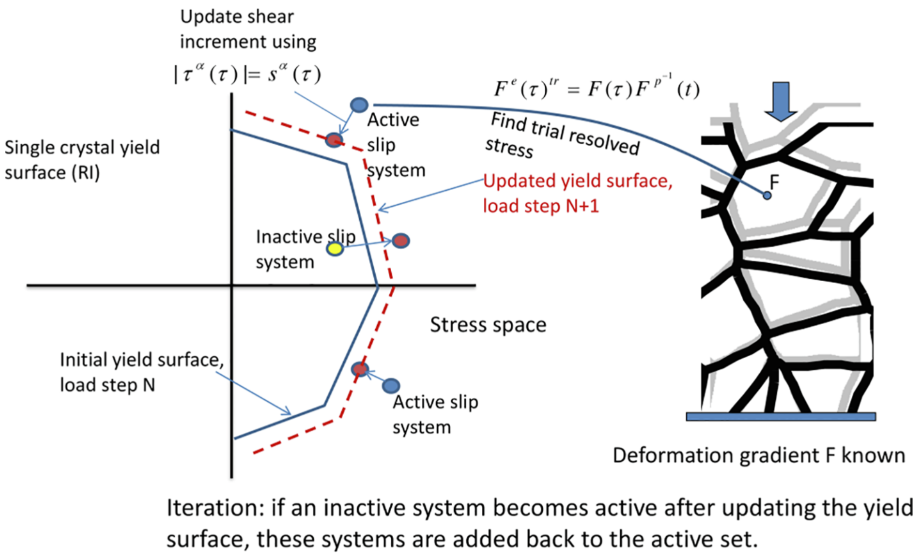

Update shear increment using
$|\tau^{\alpha}(\tau)| = s^{\alpha}(\tau)$
$F^{e}(\tau)^{tr} = F(\tau)F^{p^{-1}}(t)$
Find trial resolved stress
Single crystal yield surface (RI)
Active slip system
Updated yield surface, load step N+1
F
Inactive slip system
Stress space
Initial yield surface, load step N
Active slip system
Deformation gradient F known
Iteration: if an inactive system becomes active after updating the yield surface, these systems are added back to the active set.

Fig. 3. Schematic of the active slip systems and computation of the updated yield surface. The iterative algorithm ensures that all the slip systems lie on or inside the yield surface.

parameter for slip system β, s_s^{β} is the slip resistance at hardening saturation for slip system β, and a^{β} is a material constant for slip system β which governs the sensitivity of the hardening moduli to the slip resistance.

2.2. Incremental formulation

During deformation, a deformed body at time$t$ with the observable thermodynamics variables of $\{\mathbf{F}(t), \mathbf{F}^{\mathrm{e}}(t), \mathbf{F}^{\mathrm{p}}(t), \sigma(t), s^{\alpha}(t)\}$ undergoes the incremental deformation and the new deformed body at time $\tau = t + \Delta t$ has the updated observable thermodynamics variables of $\{\mathbf{F}(\tau), \mathbf{F}^{\mathrm{e}}(\tau), \mathbf{F}^{\mathrm{p}}(\tau), \sigma(\tau), s^{\alpha}(\tau)\}$. The constitutive law can be described as below:

$$\mathbf{S}(\tau) = \mathscr{C}[\mathbf{E}^{\mathrm{e}}(\tau)] \tag{9}$$

where $\mathbf{S}(\tau)$ is the second Piola-Kirchhoff (PKII) stress tensor relative to the relaxed configuration at time τ, $\mathscr{C}$ is the fourth order elastic stiffness tensor. The relaxed configuration, which is also known as intermediate configuration, defines a fictitious state of the material in which the material point is unloaded. In this configuration, the material point is deformed solely plastically (see Fig. 2). Finally, $\mathbf{E}^{\mathrm{e}}(\tau)$ is the elastic Green strain tensor at time τ defined as follows:

$$\mathbf{E}^{\mathrm{e}}(\tau) = \frac{1}{2}[\mathbf{C}^{\mathrm{e}}(\tau) - \mathbf{I}] \tag{10}$$

where $\mathbf{I}$ is the second order identity tensor and $\mathbf{C}^{\mathrm{e}}(\tau)$ is the elastic right Cauchy-Green deformation tensor at time τ, which can be defined as follows:

$$\mathbf{C}^{\mathrm{e}}(\tau) = \mathbf{F}^{\mathrm{e}}(\tau)^{\mathrm{T}}\mathbf{F}^{\mathrm{e}}(\tau) \tag{11}$$

The relation between the PKII stress tensor $\mathbf{S}(\tau)$ and the Cauchy stress tensor $\sigma(\tau)$ can be defined as follows:

$$\mathbf{S}(\tau) = J^{e}\mathbf{F}^{\mathrm{e}}(\tau)^{-1}\sigma(\tau)\mathbf{F}^{\mathrm{e}}(\tau)^{-\mathrm{T}} \tag{12}$$

where $J^{e} = \det[\mathbf{F}^{\mathrm{e}}(\tau)]$. The resolved shear stress for slip system α can be obtained using the PKII stress tensor $\mathbf{S}(\tau)$ as follows:

$$\tau^{\alpha}(\tau) = \{\mathbf{C}^{\mathrm{e}}(\tau)\mathbf{S}(\tau)\}: \mathbf{S}_0^{\alpha} \tag{13}$$

where $\mathbf{S}_0^{\alpha}$ is the Schmid tensor for slip system α corresponding to the relaxed configuration, which is the same as that in the undeformed configuration because plastic slip does not directly affect the crystallography of the underlying lattice. In the case of infinitesimal elastic stretches, Eq. (13) can be simplified by neglecting elastic stretches as below (see Anand and Kothari [23]):

$$\tau^{\alpha}(\tau) = \mathbf{S}(\tau): \mathbf{S}_0^{\alpha} \tag{14}$$

The trial state is defined to develop a framework of constitutive model in which the incremental deformation from time $= t$ to time $= \tau$ is assumed to be purely elastic. The trial elastic deformation gradient tensor is obtained as follows:

$$\mathbf{F}^{\mathrm{e}}(\tau)^{\mathrm{tr}} = \mathbf{F}(\tau)\mathbf{F}^{\mathrm{p}}(t)^{-1} \tag{15}$$

The trial elastic right Cauchy-Green deformation tensor $\mathbf{C}^{\mathbf{e}}(\tau)^{\mathrm{tr}}$ and trial elastic Green strain tensor $\mathbf{E}^{\mathrm{e}}(\tau)^{\mathrm{tr}}$ can be obtained as follows:

$$\mathbf{C}^{\mathrm{e}}(\tau)^{\mathrm{tr}} = \mathbf{F}^{\mathrm{e}}(\tau)^{\mathrm{tr\,T}}\mathbf{F}^{\mathrm{e}}(\tau)^{\mathrm{tr}} \tag{16}$$

$$\mathbf{E}^{\mathrm{e}}(\tau)^{\mathrm{tr}} = \frac{1}{2}[\mathbf{C}^{\mathrm{e}}(\tau)^{\mathrm{tr}} - \mathbf{I}] \tag{17}$$

The trial PKII stress tensor $\mathbf{S}(\tau)^{\mathrm{tr}}$ is then obtained as below:

$$\mathbf{S}(\tau)^{\mathrm{tr}} = \mathscr{C}[\mathbf{E}^{\mathrm{e}}(\tau)^{\mathrm{tr}}] \tag{18}$$

Finally, the trial resolved shear stress can be described as follows:

$$\tau^{\alpha}(\tau)^{\mathrm{tr}} = \mathbf{S}(\tau)^{\mathrm{tr}}: \mathbf{S}_0^{\alpha} \tag{19}$$

Furthermore, based on the following assumption (Anand and Kothari [23]):

$$\mathrm{sign}[\tau^{\alpha}(\tau)^{\mathrm{tr}}] = \mathrm{sign}[\tau^{\alpha}(\tau)] \tag{20}$$

The incremental flow rule is obtained by integration of Eq. (3) followed by a linearization as follows:

$$\mathbf{F}^{\mathrm{p}}(\tau) = \left[\mathbf{I} + \sum_{\alpha} \Delta\gamma^{\alpha}\mathrm{sign}[\tau^{\alpha}(\tau)^{\mathrm{tr}}]\mathbf{S}_0^{\alpha}\right]\mathbf{F}^{\mathrm{p}}(t) \tag{21}$$

Anand and Kothari [23] introduced the concept of inactive and active slip systems (Fig. 3). In the case of trial resolved shear stress point inside the yield surface, i.e., $|\tau^{\alpha}(\tau)^{\mathrm{tr}}| - s^{\alpha} < 0$, slip system α is inactive and $\Delta\gamma^{\alpha} = 0$. However, the slip system α can be potentially active if the trial resolve shear stress point located outside of the yield surface, i.e., $\tau^{\alpha}(\tau)^{\mathrm{tr}} - s^{\alpha} > 0$. The final active systems are the slip systems with $\Delta\gamma^{\alpha} > 0$, which should be among the potential active ones. Accordingly, first, the potentially active slip systems should be determined. Next, the slip resistances are updated as follows:

$$s^{\alpha}(\tau) = s^{\alpha}(t) + \sum_{\beta \in \mathscr{A}} h^{\alpha\beta}(t)\Delta\gamma^{\beta} \tag{22}$$

where $\mathscr{A}$ represents the set of active slip systems. The elastic deformation gradient tensor $\mathbf{F}^{e}(\tau)$ can be obtained using Eq. (21) as follows:

$$\mathbf{F}^{e}(\tau) = \mathbf{F}(\tau)\mathbf{F}^{p}(\tau)^{-1} = \mathbf{F}(\tau)\left\{\left[\mathbf{I} + \sum_{\alpha} \Delta\gamma^{\alpha}\mathrm{sign}[\tau^{\alpha}(\tau)^{\mathrm{tr}}]\mathbf{S}_{0}^{\alpha}\right]\mathbf{F}^{p}(t)\right\}^{-1} \tag{23}$$

Neglecting the higher order terms of $\Delta\gamma^{\beta}$, Eq. (23) can be simplified as follows:

$$\mathbf{F}^{e}(\tau) = \mathbf{F}^{e}(\tau)^{\mathrm{tr}}\left[\mathbf{I} - \sum_{\alpha} \Delta\gamma^{\alpha}\mathrm{sign}[\tau^{\alpha}(\tau)^{\mathrm{tr}}]\mathbf{S}_{0}^{\alpha}\right] \tag{24}$$

Accordingly, the PKII stress tensor and resolved shear stress can be obtained as follows:

$$\mathbf{S}(\tau) = \mathbf{S}(\tau)^{\mathrm{tr}} - \sum_{\alpha\in\mathscr{A}} \Delta\gamma^{\alpha}\mathrm{sign}[\tau^{\alpha}(\tau)^{\mathrm{tr}}]\mathscr{L}[\mathrm{sym}(\mathbf{C}^{\mathbf{e}}(\tau)^{\mathrm{tr}}\mathbf{S}_{0}^{\alpha})] \tag{25}$$

$$|\tau^{\alpha}(\tau)| = |\tau^{\alpha}(\tau)^{\mathrm{tr}}| - \sum_{\beta\in\mathscr{A}} \mathrm{sign}[\tau^{\alpha}(\tau)^{\mathrm{tr}}]\mathrm{sign}[\tau^{\beta}(\tau)^{\mathrm{tr}}]\mathbf{S}_{0}^{\alpha} : \mathscr{L}[\mathrm{sym}(\mathbf{C}^{\mathbf{e}}(\tau)^{\mathrm{tr}}\mathbf{S}_{0}^{\beta})]\Delta\gamma^{\beta} \tag{26}$$

The consistency condition can be rewritten by incorporating Eqs. (22) and (26) as follows:

$$\sum_{\beta\in\mathscr{A}} A^{\alpha\beta}x^{\beta} = b^{\alpha} \tag{27}$$

where:

$$A^{\alpha\beta} = h^{\alpha\beta}(t) + \mathrm{sign}[\tau^{\alpha}(\tau)^{\mathrm{tr}}]\mathrm{sign}[\tau^{\beta}(\tau)^{\mathrm{tr}}]\mathbf{S}_{0}^{\alpha}\colon \mathscr{L}[\mathrm{sym}(\mathbf{C}^{\mathbf{e}}(\tau)^{\mathrm{tr}}\mathbf{S}_{0}^{\beta})] \tag{28}$$

$$b^{\alpha} = |\tau^{\alpha}(\tau)^{\mathrm{tr}}| - s^{\alpha}(t) > 0 \tag{29}$$

$$x^{\beta} = \Delta\gamma^{\beta} \tag{30}$$

After solving Eqs. (27)–(30), the slip systems with $x^{\beta} = \Delta\gamma^{\beta} \leq 0$ should be removed from the active slip system set. Accordingly, Eqs. (27)–(30) should be updated and solved again until all active slip systems have $\Delta\gamma > 0$. After obtaining the consistent set of values for $\Delta\gamma$ for each active slip system, the plastic deformation gradient is updated using Eq. (21). It is noted that Eq. (21) does not ensure that $\det[\mathbf{F}^{\mathbf{p}}(\tau)] = 1$. Accordingly, one should normalize $\mathbf{F}^{\mathbf{p}}(\tau)$ as follows:

$$\mathbf{F}^{\mathbf{p}}(\tau) = \{\det[\mathbf{F}^{\mathbf{p}}(\tau)]\}^{-\frac{1}{3}}\mathbf{F}^{\mathbf{p}}(\tau) \tag{31}$$

where $\mathbf{F}^{\mathbf{p}}(\tau)$ in the left-hand side of Eq. (31) is normalized while the one in the right-hand side of Eq. (31) is not normalized. Next, the elastic deformation gradient is updated using the multiplicative decomposition as follows:

$$\mathbf{F}^{\mathbf{e}}(\tau) = \mathbf{F}(\tau)\mathbf{F}^{\mathbf{p}}(\tau)^{-1} \tag{32}$$

The PKII stress tensor $\mathbf{S}(\tau)$ is obtained using Eq. (25). The slip resistances can then be updated using Eq. (22) and the updated Cauchy stress tensor $\boldsymbol{\sigma}(\tau)$ is obtained below:

$$\boldsymbol{\sigma}(\tau) = J^{e-1}\mathbf{F}^{\boldsymbol{e}}(\tau)\mathbf{S}(\tau)\mathbf{F}^{\boldsymbol{e}}(\tau)^{\mathrm{T}} \tag{33}$$

It should be noted that $J = \det(\mathbf{F}) = J^{e} = \det[\mathbf{F}^{\mathbf{e}}(\tau)]$ according to Eqs. (31) and (32).

One can check the consistency equation for all active slip systems as follows:

$$f^{\alpha} = |\tau^{\alpha}(\tau)| - s^{\alpha}(\tau) = 0;\ \ \alpha \in \mathscr{A} \tag{34}$$

If $|\tau^{\alpha}(\tau)| > s^{\alpha}(\tau)$ for some slip systems, the plastic shear strain should be updated. Accordingly, the procedure should be repeated to satisfy the consistency equation.

The *PRISMS-Plasticity* CPFE model is implemented on top of the open source deal.II [19] C++ finite element library. The *PRISMS-Plasticity* open source software package, including its manual and several examples, is available through the GitHub repository (see Appendix A).

2.3. Deformation solver

The equilibrium equation of the body can be expressed in Lagrangian framework as follows:

$$\nabla_{0}\cdot\mathbf{P} + \mathbf{f} = 0 \tag{35}$$

where ∇_{0} is the divergence in the reference configuration, $\mathbf{P}$ is the first Piola-Kirchhoff (PKI) stress tensor and $\mathbf{f}$ is the reference body force vector. Principle of virtual work states that the body ß_{0} is in equilibrium if and only if the PKI stress field, i.e., $\mathbf{P}$, satisfies the virtual work functional for any kinematically admissible test function $\mathbf{u}$:

$$G(\mathbf{u},\mathbf{u}) \equiv \int_{\text{ß}_{0}} \mathbf{P}\colon \nabla_{0}\mathbf{u}dV - \int_{\partial\text{ß}_{0}} \lambda\cdot\mathbf{u}dA - \int_{\text{ß}_{0}} \mathbf{f}\cdot\mathbf{u}dV = 0\ \forall\ \mathbf{u} \in \mathscr{V} \tag{36}$$

where $\mathbf{u}$ is the displacement field, $\mathscr{V}$ is a finite dimensional vector space of all admissible shape functions in the material domain, $\mathbf{f}$ is the reference body force, and $\boldsymbol{\lambda}$ is the surface traction field.

The dependency of the function $G(\mathbf{u},\mathbf{u})$ on the displacement vector $\mathbf{u}$ is originated from the constitutive dependency of the PKI stress tensor $\mathbf{P}$ on deformation gradient $\mathbf{F}$, which can be related to the displacement vector $\mathbf{u}$ as follows:

$$\mathbf{F} = \mathbf{I} + \nabla_{0}\mathbf{u} \tag{37}$$

The Newton-Raphson iterative scheme with a line search procedure is employed to solve the nonlinear system of Eq. (36). The Gâteaux derivative of G at $\mathbf{u}_{n}$ in the direction of $\delta\mathbf{u}$ can be obtained as follows:

$$\mathrm{D}G(\mathbf{u}_{n},\mathbf{u})[\delta\mathbf{u}] = \int_{\text{ß}_{0}} \mathbf{A}\colon \nabla_{0}\delta\mathbf{u}\colon \nabla_{0}\mathbf{u}dV \tag{38}$$

where $\boldsymbol{A}$ is commonly termed as the material tangent modulus, which can be described as follows:

$$\boldsymbol{A} \equiv \left.\frac{\partial\mathbf{P}}{\partial\mathbf{F}}\right|_{\mathbf{F}_{\boldsymbol{n}}} \tag{39}$$

The derivation of the material tangent modulus $\boldsymbol{A}$ is an important part of any implicit scheme which should be accurately calculated to solve the problem. The derivation involves computing variation of different mechanical quantities. The material tangent modulus $\boldsymbol{A}$ can then be obtained as the variation of the stress measure with respect to the variation of the deformation gradient. Since the derivation of material tangent modulus $\boldsymbol{A}$ depends on the incorporated integration algorithm, it is commonly termed as the algorithmic tangent modulus. Accordingly, an incremental scheme is incorporated to numerically solve Eq. (36) in which the deformed body $\text{ß}_{(\mathrm{time}=\mathrm{t})}$ at time $= t$ with the observable thermodynamics variables of $\{\mathbf{F}(t), \mathbf{F}^{\mathbf{e}}(t), \mathbf{F}^{\mathbf{p}}(t), \sigma(t), s^{\alpha}(t)\}$ undergoes the incremental deformation and the new deformed body $\text{ß}_{(\mathrm{time}=\tau)}$ at time $\tau = t + \Delta t$ has the updated observable thermodynamics variables of $\{\mathbf{F}(\tau), \mathbf{F}^{\mathbf{e}}(\tau), \mathbf{F}^{\mathbf{p}}(\tau), \sigma(\tau), s^{\alpha}(\tau)\}$. Again, the trial state is defined by assuming a pure elastic incremental deformation from time $= t$ to time $= \tau$. It is worth mentioning that the system involves both geometric and material nonlinearity resulting in a system of nonlinear equations which need to be solved iteratively. As a result, the constitutive model is invoked once in every iteration which provides an estimate of the observable thermodynamics variables and the tangent modulus for the subsequent step of the line search for the solution to the global FEM system of equations. The only variable that remains fixed during each iteration is the deformation gradient estimate, which is given to the constitutive equations as an input. In the next section, every variable represents the estimate computed from the constitutive model in the most recent iteration, and accordingly, the time is dropped from the notations, e.g., $\mathbf{F}^{\mathbf{e}} \equiv \mathbf{F}^{\mathbf{e}}(\tau)$

2.4. Algorithmic tangent modulus

As a first step, the PKI stress tensor **P** can be related to the Cauchy stress tensor σ as follows:

$$\mathbf{P} = \det(\mathbf{F})\sigma\mathbf{F}^{-\mathrm{T}} \tag{40}$$

The variation of PKI stress tensor can then be defined as follows:

$$\begin{aligned}\delta(\mathbf{P}) &= \delta(\det(\mathbf{F})\sigma\mathbf{F}^{-\mathrm{T}}) = \delta(\det(\mathbf{F}))\sigma\mathbf{F}^{-\mathrm{T}} + \mathbf{F}\delta(\sigma)\mathbf{F}^{-\mathrm{T}} + \mathbf{F}\sigma\delta(\mathbf{F}^{-\mathrm{T}})\\ &= \det(\mathbf{F})\mathrm{tr}[\delta(\mathbf{F})\mathbf{F}^{-1}]\sigma\mathbf{F}^{-\mathrm{T}} + \det(\mathbf{F})\delta(\sigma)\mathbf{F}^{-\mathrm{T}} - \det(\mathbf{F})\sigma\mathbf{F}^{-\mathrm{T}}\delta(\mathbf{F}^{\mathrm{T}})\mathbf{F}^{-\mathrm{T}}\end{aligned} \tag{41}$$

The relation between the Cauchy stress tensor σ and PKII stress tensor **S** can be defined as follows:

$$\sigma = \frac{1}{\det(\mathbf{F}^{\mathrm{e}})}\mathbf{F}^{\mathbf{e}}\mathbf{S}\mathbf{F}^{\mathbf{eT}} \tag{42}$$

Accordingly, one can obtain $\delta(\sigma)$ as follows:

$$\begin{aligned}&\delta(\sigma)\\ &= \delta\left(\frac{1}{\det(\mathbf{F}^{\mathrm{e}})}\right)\mathbf{F}^{\mathrm{e}}\mathbf{S}\mathbf{F}^{\mathrm{eT}} + \frac{1}{\det(\mathbf{F}^{\mathrm{e}})}[\delta(\mathbf{F}^{\mathrm{e}})\mathbf{S}\mathbf{F}^{\mathrm{eT}} + \mathbf{F}^{\mathrm{e}}\delta(\mathbf{S})\mathbf{F}^{\mathrm{eT}} + \mathbf{F}^{\mathrm{e}}\mathbf{S}\delta(\mathbf{F}^{\mathrm{eT}})]\\ &= \frac{1}{\det(\mathbf{F}^{\mathrm{e}})}[-\mathrm{tr}(\delta(\mathbf{F}^{\mathrm{e}})\mathbf{F}^{\mathrm{e}-1})\mathbf{F}^{\mathbf{e}}\mathbf{S}\mathbf{F}^{\mathrm{eT}} + \delta(\mathbf{F}^{\mathrm{e}})\\ &\quad \mathbf{S}\mathbf{F}^{\mathrm{eT}} + \mathbf{F}^{\mathrm{e}}\delta(\mathbf{S})\mathbf{F}^{\mathrm{eT}} + \mathbf{F}^{\mathbf{e}}\mathbf{S}\delta(\mathbf{F}^{\mathrm{eT}})]\end{aligned} \tag{43}$$

One need to obtain the expressions for $\delta(\mathbf{F}^{\mathrm{e}})$ and $\delta(\mathbf{S})$. The elastic deformation gradient tensor $\mathbf{F}^{\mathrm{e}}$ can be defined using Eq. (24) as follows:

$$\mathbf{F}^{\mathbf{e}} \approx \mathbf{F}^{\mathrm{etr}}\left[\mathbf{I} - \sum_{\alpha}\Delta\gamma^{\alpha}\mathbf{S}_0^{\alpha}\mathrm{sign}(\tau^{\alpha})\right] \tag{44}$$

where $\mathbf{F}^{\mathrm{etr}}$ is the trial elastic deformation gradient tensor. Accordingly, $\delta(\mathbf{F}^{\mathrm{e}})$ can be related to $\delta(\Delta\gamma^{\beta})$ using Eqs. (21), (31) and (32) as follows:

$$\begin{aligned}\delta(\mathbf{F}^{\mathbf{e}}) &\approx \delta\left(\mathbf{F}^{\mathrm{etr}}\left[\mathbf{I} - \sum_{\alpha}\Delta\gamma^{\alpha}\mathbf{S}_0^{\alpha}\mathrm{sign}(\tau^{\alpha})\right]\right) \approx \delta(\mathbf{F}^{\mathrm{etr}})\\ &\quad \left(\mathbf{I} - \sum_{\alpha}\Delta\gamma^{\alpha}\mathbf{S}_0^{\alpha}\mathrm{sign}(\tau^{\alpha})\right) - \left(\sum_{\alpha}\mathbf{F}^{\mathrm{etr}}\delta(\Delta\gamma^{\alpha})\mathbf{S}_0^{\alpha}\mathrm{sign}(\tau^{\alpha})\right)\\ &\approx \delta(\mathbf{F})(\mathbf{F}^{\mathrm{ptr}})^{-1}\left(\mathbf{I} - \sum_{\alpha}\Delta\gamma^{\alpha}\mathbf{S}_0^{\alpha}\mathrm{sign}(\tau^{\alpha})\right) - \left\{\sum_{\alpha}\mathrm{sign}(\tau^{\alpha})\mathbf{F}^{\mathrm{etr}}\mathbf{S}_0^{\alpha}\right.\\ &\quad \left.\left[\frac{\partial(\Delta\gamma^{\alpha})}{\partial\mathbf{E}^{\mathrm{etr}}}\right]:\delta(\mathbf{E}^{\mathrm{etr}})\right\} \approx \delta(\mathbf{F})\mathbf{F}^{\mathrm{p}-1} - \left\{\sum_{\alpha}\mathrm{sign}(\tau^{\alpha})\mathbf{F}^{\mathrm{etr}}\mathbf{S}_0^{\alpha}\left[\frac{\partial(\Delta\gamma^{\alpha})}{\partial\mathbf{E}^{\mathrm{etr}}}\right]\right.\\ &\quad \left.:\delta(\mathbf{E}^{\mathrm{etr}})\right\}\end{aligned} \tag{45}$$

In order to obtain the relation between $\delta\mathbf{F}^{\mathrm{e}}$ and $\delta\mathbf{F}$, one should obtain $\delta(\mathbf{E}^{\mathrm{etr}})$ and $\delta(\Delta\gamma^{\beta})$ One can use Eqs. (16) and (17) to obtain $\delta(\mathbf{E}^{\mathrm{etr}})$as follows:

$$\begin{aligned}\delta(\mathbf{E}^{\mathrm{etr}}) &= \frac{1}{2}[\delta((\mathbf{F}^{\mathrm{etr}})^{\mathrm{T}})\mathbf{F}^{\mathrm{etr}} + (\mathbf{F}^{\mathrm{etr}})^{\mathrm{T}}\delta(\mathbf{F}^{\mathrm{etr}})]\\ &= \frac{1}{2}[(\mathbf{F}^{\mathrm{ptr}})^{-\mathrm{T}}\delta\mathbf{F}^{\mathrm{T}}\mathbf{F}^{\mathrm{etr}} + (\mathbf{F}^{\mathrm{etr}})^{\mathrm{T}}\delta\mathbf{F}(\mathbf{F}^{\mathrm{ptr}})^{-1}]\end{aligned} \tag{46}$$

In order to obtain $\delta(\Delta\gamma^{\beta})$, one can use Eq. (27) as a starting point as follows:

$$\mathbf{b} = \mathbf{A}\Delta\gamma \to \delta\mathbf{b} = \delta\mathbf{A}\Delta\gamma + \mathbf{A}\delta(\Delta\gamma) \tag{47}$$

Accordingly, $\delta(\Delta\gamma)$ can be obtained as follows:

$$\delta(\Delta\gamma) = \mathbf{A}^{-1}(\delta\mathbf{b} - \delta\mathbf{A}\Delta\gamma) \tag{48}$$

The term $\delta\mathbf{b}$ can be derived by using Eq.(29) as follows:

$$\delta b^{\alpha} = \mathrm{sign}(\tau^{\alpha\mathrm{tr}})\delta\mathbf{S}^{\mathrm{tr}}:\mathbf{S}_0^{\alpha} - \delta(s^{\alpha}(t)) \approx \mathrm{sign}(\tau^{\alpha\mathrm{tr}})\mathscr{l}[\delta(\mathbf{E}^{\mathrm{etr}})]:\mathbf{S}_0^{\alpha} \tag{49}$$

Note that in the previous expression, $\delta(s^{\alpha}(t))$ can be approximated to zero because any variation of the deformation gradient tensor, i.e., $\delta(\mathbf{F})$, does not change the observable thermodynamics variables obtained in earlier time steps. Following the same procedure, $\delta A^{\alpha\beta}$ can be obtained using Eq. (28) as follows:

$$\begin{aligned}\delta A^{\alpha\beta} &= \delta h^{\alpha\beta} + \mathrm{sign}(\tau^{\alpha\mathrm{tr}})\mathrm{sign}(\tau^{\beta\mathrm{tr}})\mathbf{S}_0^{\alpha}:\mathscr{l}[\delta(\mathbf{B}^{\beta\mathrm{tr}})] \approx \mathrm{sign}(\tau^{\alpha\mathrm{tr}})\mathrm{sign}(\tau^{\beta\mathrm{tr}})\mathbf{S}_0^{\alpha}\\ &\quad :\mathscr{l}[\delta(\mathbf{B}^{\beta\mathrm{tr}})]\end{aligned} \tag{50}$$

where $\delta(\mathbf{B}^{\beta\mathrm{tr}})$ can be defined as follows:

$$\delta(\mathbf{B}^{\beta\mathrm{tr}}) = \frac{1}{2}[(\mathbf{S}_0^{\beta})^{\mathrm{T}}\delta(\mathbf{C}^{\mathrm{etr}})^{\mathrm{T}} + \delta(\mathbf{C}^{\mathrm{etr}})\mathbf{S}_0^{\beta}] = [(\mathbf{S}_0^{\beta})^{\mathrm{T}}\delta(\mathbf{E}^{\mathrm{etr}})^{\mathrm{T}} + \delta(\mathbf{E}^{\mathrm{etr}})\mathbf{S}_0^{\beta}] \tag{51}$$

One should note that $\delta h^{\alpha\beta}$ is proportional to $\delta(s^{\beta}(t))$ and accordingly can be approximated to zero. One can define $\delta(\Delta\gamma^{\alpha})$ as a function of $\delta(\mathbf{E}^{\mathrm{etr}})$ by substituting Eqs. (49)–(51) into Eq. (48). Accordingly, one can use Eq. (45) to define $\delta(\mathbf{F}^{\mathrm{e}})$ as a function of $\delta(\mathbf{F})$. Finally, a relation between $\delta(\mathbf{P})$ and $\delta(\mathbf{F})$ can be defined using Eqs. (41)–(43).

3. Twinning model

3.1. Kinematics

Twinning is an important deformation mode in many metals and alloys. The crystallography of the twinning mechanism is presented in Fig. 4, which includes the parent grain, twinned region, and the orientation of the corresponding regions. The kinematics of slip and twinning is shown in Fig. 5. Twin systems are considered as polar pseudo-slip systems and are sheared until they are reoriented. In the *PRISMS-Plasticity* CPFE software, the twinning model developed by Staroselsky et al. [27] is incorporated. However, some modifications have been applied to the twinning model compared to the original scheme (Staroselsky et al. [27]) as follows:

- In the current work, the predominant twin reorientation scheme (PTR) developed by Tomé et al. [28] is incorporated. However, Staroselsky et al. [27] used the scheme proposed by Van Houtte [29]. Van Houtte [29] modeled twinning as a polar pseudo-slip mechanism, and the volume of new grain created by twinning is proportional to the amount of accommodated shear strain. In the case of PTR scheme (Tomé et al. [28]), the entire grain is reoriented as the accommodated shear strain reaches a threshold.
- Unlike the Staroselsky et al. [27] who applied the twinning criteria to the entire grain, in *PRISMS-Plasticity* CPFE software, the PTR scheme is applied to each Gauss integration point in a way that the accommodated shear strain for each Gauss integration point is monitored. The individual Gauss integration point is reoriented if the accommodated shear strain reaches the twinning threshold as compared to reorienting the entire grain. The orientations of all

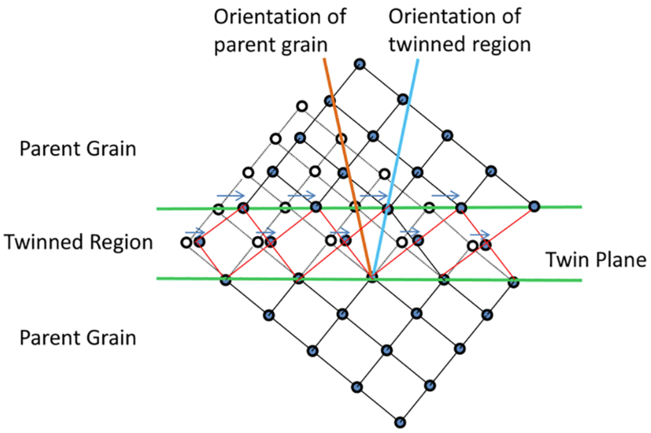


Fig. 4. Crystallography of twinning.

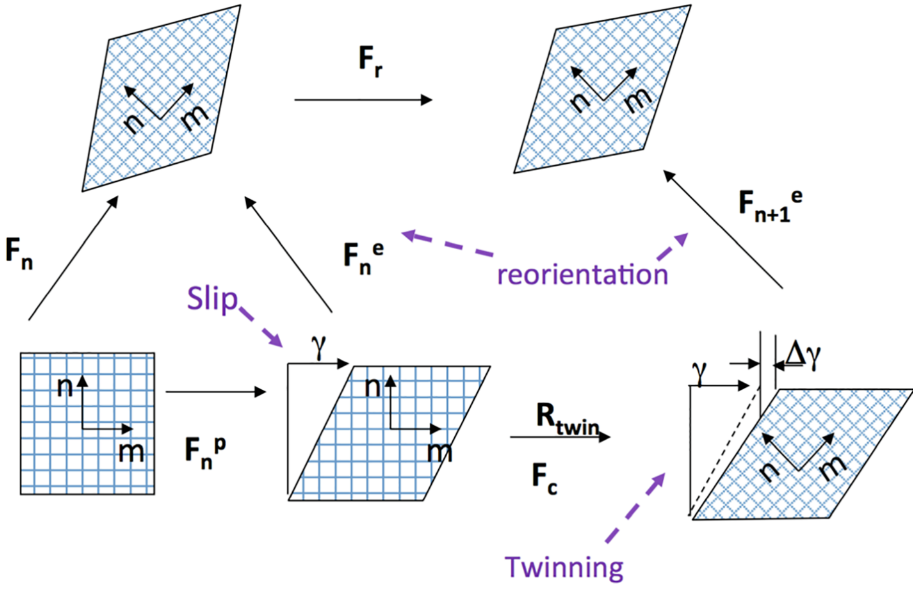


Fig. 5. Kinematics of slip and twinning.

Gauss integration points are tracked in the FE simulation and reorient the individual points, which satisfy the PTR scheme (Tomé et al. [28]).

3.2. Reorientation criteria

In the framework developed for rate-independent CPFE, the twin volume fraction of n^{th} Gauss integration point associated with the i^{th} twinning system for each time step can be described as follows:

$$\Delta g^{n,t_i} = \frac{\Delta\gamma^{n,t_i}}{S} \tag{52}$$

where S is the characteristic twin shear strain, which is calculated as 0.129 in the case of magnesium (Christian and Mahajan [30]). Accordingly, the accumulative twinning volume fraction at time step T can be obtained as follows:

$$g^{n,t_i} = \sum_{\text{timestep}=1}^{T} \Delta g^{n,t_i} \tag{53}$$

The Gauss integration point is reoriented due to twinning if the accumulative twinning volume fraction g^{n,t_i} is larger than the threshold F_T, which can be defined as follows:

$$F_T = A + B\frac{F_E}{F_R} \tag{54}$$

where A and B are material constants, F_E is the total volume fraction of reoriented Gauss integration points in which the twinning volume fraction reaches the threshold ,i.e., $g^{n,t_i} > F_T$, and F_R is the real twinned fraction, which can be defined as follows:

$$F_R = \sum_{n=1}^{N} f^n \sum_{i=1}^{N_{\text{tw}}} g^{n,t_i} \tag{55}$$

where f^n is the volume fraction of n^{th} Gauss integration point, N is the number of Gauss integration points, and N_{tw} is the number of twinning systems.

3.3. Lattice reorientation due to twinning

In the case of a Gauss integration point in which the twinning volume fraction reaches the threshold, i.e., $g^{n,t_i} > F_T$, it should be reoriented. A schematic diagram of reorientation due to twinning is shown in Fig. 6. The rotation matrix for rotating the crystal frame about the twin plane $\mathbf{R}_2$ can be obtained as below:

$$\mathbf{R}_2 = 2\mathbf{n} \otimes \ \mathbf{n} - \mathbf{I} \tag{56}$$

where $\mathbf{n}$ is the unit vector normal to the twin plane for the grain of interest, and $\mathbf{I}$ is the second order identity tensor. Also, twinning can be described an operation of rotation in which the twinned region is obtained from the untwinned one by means of rotation of π about an axis parallel to $\mathbf{n},$ which is the unit vector normal to the twin plane for the grain of interest. Accordingly, the quaternion representation of twinning operation can be described as follows:

$$\mathbf{q}_{tw} = \cos\frac{\pi}{2} + (n_x\boldsymbol{i} + n_y\boldsymbol{j} + n_z\boldsymbol{k})\sin\frac{\pi}{2} = [0,\ n_x,\ n_y,\ n_z] \tag{57}$$

Two schemes can be used to find the new orientation for the twinned grain. First, one can use Eq. (57) to obtain the Rodrigues vector of the twinned grain using the following procedure:

- Calculation of the rotation matrix for reference frame to crystal frame, i.e., $\mathbf{R}_1$.
- Calculation of the rotation matrix for rotating the crystal frame about the twin plane by 180^0, i.e., $\mathbf{R}_2$, using Eq. (57).
- Calculation of the new rotation matrix $\mathbf{Q}$ is defined as a combination of both rotations $\mathbf{R}_2$ and $\mathbf{R}_1$ as $\mathbf{Q} = \mathbf{R}_1\mathbf{R}_2$**.**
- Projection of $\mathbf{Q}$ to the fundamental region, i.e., $\mathbf{QF}$**,** based on crystal symmetries.
- Conversion of the rotation matrix $\mathbf{QF}$ to its Rodrigues vector representation.

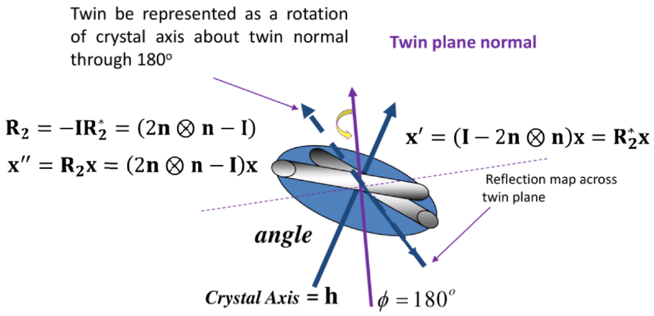

Twin be represented as a rotation of crystal axis about twin normal through 180°
Twin plane normal
$\mathbf{R_2} = -\mathbf{I}\mathbf{R}_2^* = (2\mathbf{n} \otimes \mathbf{n} - \mathbf{I})$
$\mathbf{x}'' = \mathbf{R_2}\mathbf{x} = (2\mathbf{n} \otimes \mathbf{n} - \mathbf{I})\mathbf{x}$
$\mathbf{x}' = (\mathbf{I} - 2\mathbf{n} \otimes \mathbf{n})\mathbf{x} = \mathbf{R}_2^*\mathbf{x}$
Reflection map across twin plane
angle
Crystal Axis = h
$\phi = 180^o$

Fig. 6. Reorientation due to twinning.

In the next method, one can use Eq. (58) to describe the twinning operation and obtain the Rodrigues vector of the twinned grain using the following procedure:

- Calculation of the quaternion representation of twinning operation, i.e., $\mathbf{q}_{tw}$, using Eq. (58).
- Calculation of the quaternion representation of rotation from reference frame to crystal frame, i.e., $\mathbf{q}_1$.
- Calculation of the quaternion representation of rotation from reference frame to the twinned crystal frame, i.e., $\mathbf{q}_2$, as the quaternion product of $\mathbf{q}_1$ and $\mathbf{q}_{tw}$.
- Conversion of the quaternion representation $\mathbf{q}_2$ to its Rodrigues vector representation.

4. Integration with the experimental characterization results using the open source software DREAM.3D and Neper

An important feature of the *PRISMS-Plasticity* CPFE software is the integration with experimental characterization schemes, such as electron backscatter diffraction (EBSD) and synchrotron X-ray diffraction. Here, procedures are described for generating the input files of *PRISMS-Plasticity* CPFE software using available open source software packages of DREAM.3D [20] and Neper [21]. In the case of *PRISMS-Plasticity* software, the material and simulation parameters are included within the input file, leaving two additional required input files of the mesh information and mapping of microstructure to this mesh. This is accomplished using the open source packages DREAM.3D and Neper (Fig. 7).

The integration of *PRISMS-Plasticity* with the EBSD data can be conducted through an automatic workflow using DREAM.3D [20]. DREAM.3D is an open source and modular software package to reconstruct, instantiate, quantify, mesh, handle and visualize multidimensional and multimodal data. In the present context, the first step is to use the StatsGenerator module of DREAM.3D to generate the relevant statistics associated with the input microstructure, later used to generate a synthetic representative volume element (RVE) which can mimic the main characteristics of EBSD data using DREAM.3D. To do so, the EBSD data is used as an input for a prebuilt DREAM3D pipeline to create a voxelated cuboidal synthetic microstructure which approximately obeys the supplied statistics. This pipeline, which forms an essential building block of the PRISMS Integration Framework, is available to the users through GitHub (see Appendix A). The current version of the *PRISMS-Plasticity* CPFE code admits any input mesh composed of hexahedral elements. DREAM.3D generated a voxelated RVE. Accordingly, the PRISMS-CPFE code maps the microstructure to the mesh.

Alternatively, one can use the open source software Neper [21] to link the material characterizations to the *PRISMS-Plasticity* CPFE software. Neper can generate the microstructure of the polycrystalline sample from:

- The experimental statistics of grain such as grain size and shape distributions.
- The Grain-by-grain data obtained from the synchrotron X-ray diffraction.

Neper generates polycrystalline samples by various types of Voronoi tessellation such as centroidal and Laguerre tessellations, and especially in the form of grain-boundary conforming tetrahedral meshes. However, as mentioned earlier, the input mesh to the *PRISMS-Plasticity* CPFE code needs to be composed of hexahedral elements. A tet-to-hex converter should be employed to convert the tetrahedral elements to hexahedral elements. This tet-to-hex converter code is included as a utility software within the *PRISMS-Plasticity* CPFE code. Additionally,

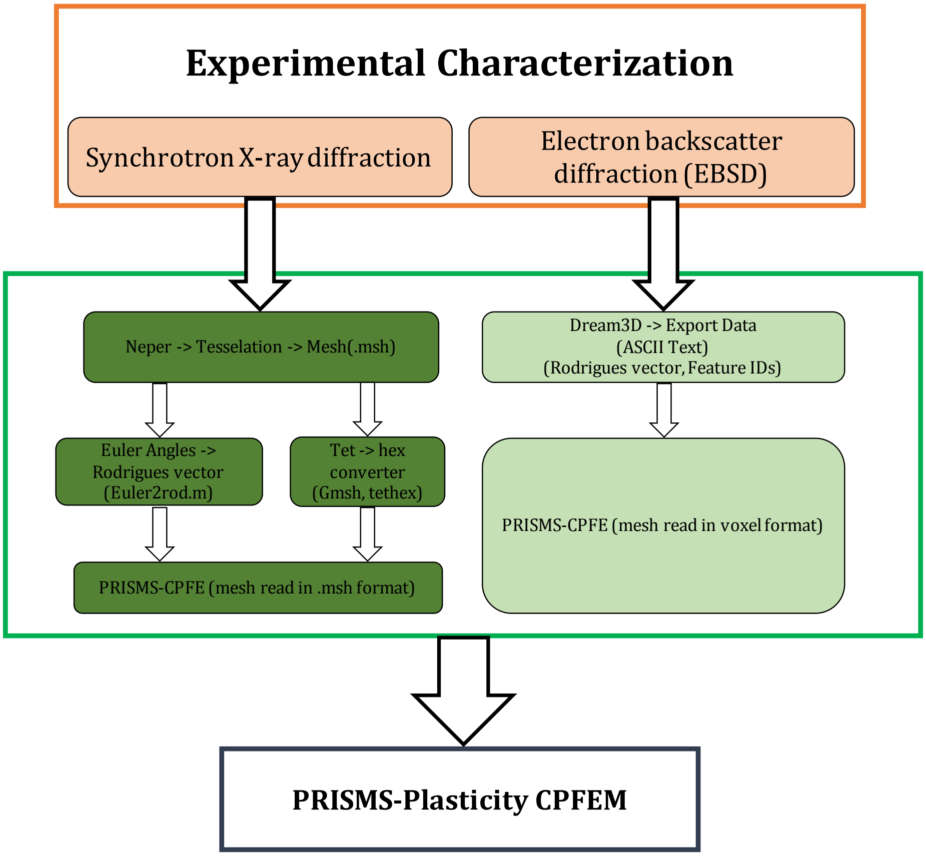

Experimental Characterization
Synchrotron X-ray diffraction
Electron backscatter diffraction (EBSD)
Neper -> Tesselation -> Mesh(.msh)
Euler Angles -> Rodrigues vector (Euler2rod.m)
Tet -> hex converter (Gmsh, tethex)
PRISMS-CPFE (mesh read in .msh format)
Dream3D -> Export Data (ASCII Text) (Rodrigues vector, Feature IDs)
PRISMS-CPFE (mesh read in voxel format)
PRISMS-Plasticity CPFEM

Fig. 7. Integration of PRISMS-Plasticity CPFE software with the experimental characterization results using the open source software of DREAM.3D and Neper.

Neper outputs orientations in the form of Euler angles, which should be converted to the Rodrigues vector. This can be done using a MATLAB script, which can be freely downloaded as utility software within the main code through GitHub (Appendix A).

5. Integration with the Materials Commons

Materials Commons [22] is a virtual collaboration space and repository for curating, archiving and disseminating information from experiments and computations. An important part of PRISMS framework is the integration of the PRISMS software with Materials Commons (Fig. 1). To do so, the Materials Commons Python Application Programming Interfaces (MCAPI) wrapper (prismscpfe_mcapi) is used to automate the integration of *PRISMS-Plasticity* CPFE simulations with the Materials Commons. Using MCAPI, the command line tool (CLI) automatically parses the software input file and the corresponding simulation results to Materials Commons. Fig. 8 shows a set of typical *PRISMS-Plasticity* CPFE input files which can be automatically uploaded to the Materials Commons repository. The prismscpfe_mcapi Python package is available to the users through the GitHub repository (see Appendix A).

6. Application examples

The application of *PRISMS-Plasticity* CPFE software is demonstrated using a set of examples for polycrystalline materials with different crystal structures of FCC, BCC, and HCP. The integration of the *PRISMS-Plasticity* CPFE software with other open source software such as DREAM.3D and Neper are elaborated. DREAM.3D and Neper were used to generate the initial microstructure. MATLAB toolbox MTEX [31] is used to plot pole figures and inverse pole figures. The displacement, stress, and strain fields are visualized using the open source visualization software ParaView [32]. The input files used in this section have been published and are available in the Materials Commons (see Appendix A).

6.1. FCC polycrystal

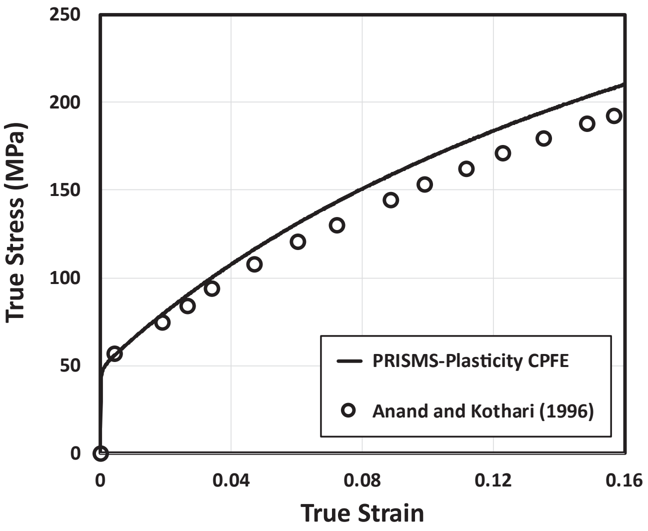


Fig. 9. The stress–strain response of polycrystalline copper sample during compression test obtained by PRISMS-Plasticity CPFE software using a block of 400 random orientations. The PRISMS-Plasticity CPFE simulation results are compared with the results from Anand and Kothari [23].

In the first set of examples, the *PRISMS-Plasticity* CPFE software is used to replicate the results presented by Anand and Kothari [23] for polycrystalline OFHC copper with initial random orientations during a compression experiment. Two different simulations are conducted. In the first simulation, the isotropic polycrystalline sample is modeled as an aggregate of 400 single crystals with random orientations, with each grain being modeled by a single eight-node linear hexahedral element. Accordingly, a 5 × 8 × 10 FE cubic mesh is generated in the x, y, and z directions, which each element represents a single grain. Following Anand and Kothari [23], Symmetry boundary conditions are imposed on adjacent x, y, and z faces and the uniaxial compression is applied in the z direction on the opposite z face. The material properties are selected similar to Anand and Kothari [23], which are $h_0^\beta = h_0 = 180\text{MPa}$, $s_0^\beta = s_0 = 16\text{MPa}$, $s_s^\beta = s_s = 148\text{MPa}$, and $a^\beta = a = 2.25$ for all slip systems [see Eq. (8)]. The latent hardening parameter of $q = 1.4$ is used for the crystal plasticity model. Fig. 9 compares

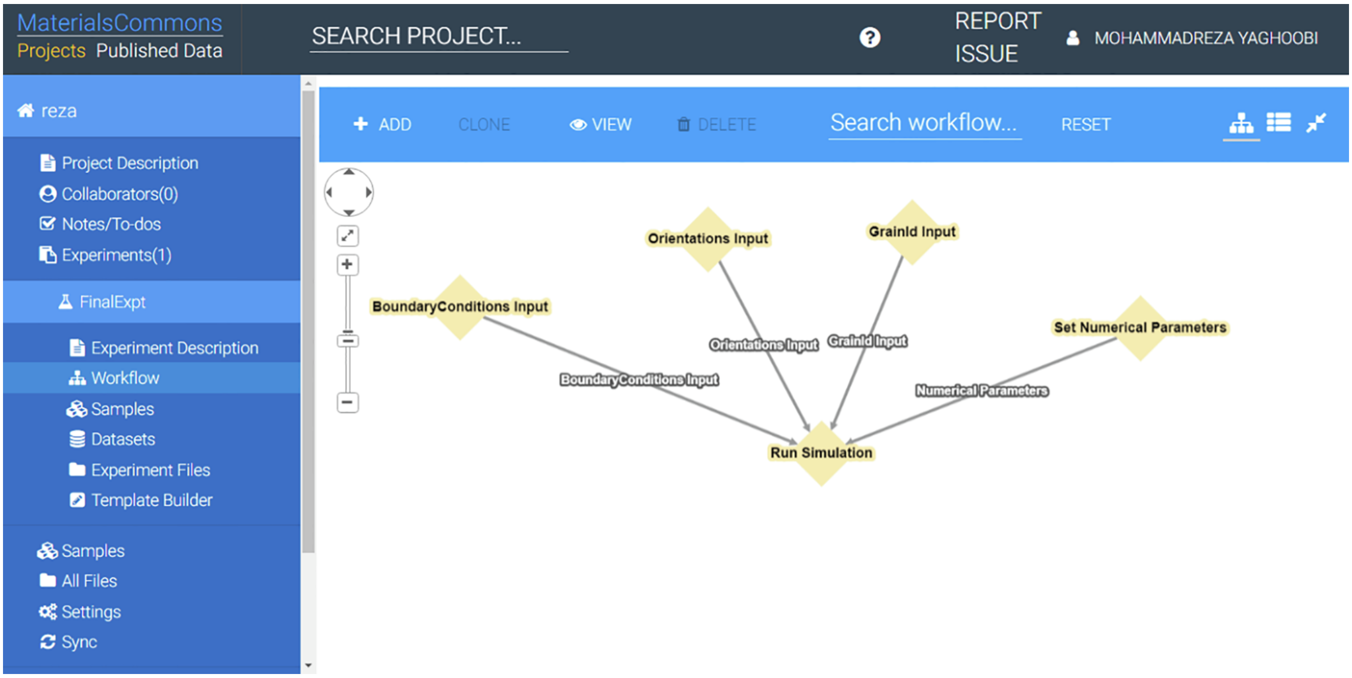


Fig. 8. A set of PRISMS-Plasticity input files which can be automatically uploaded to the Materials Commons using Python prismscpfe_mcapi.

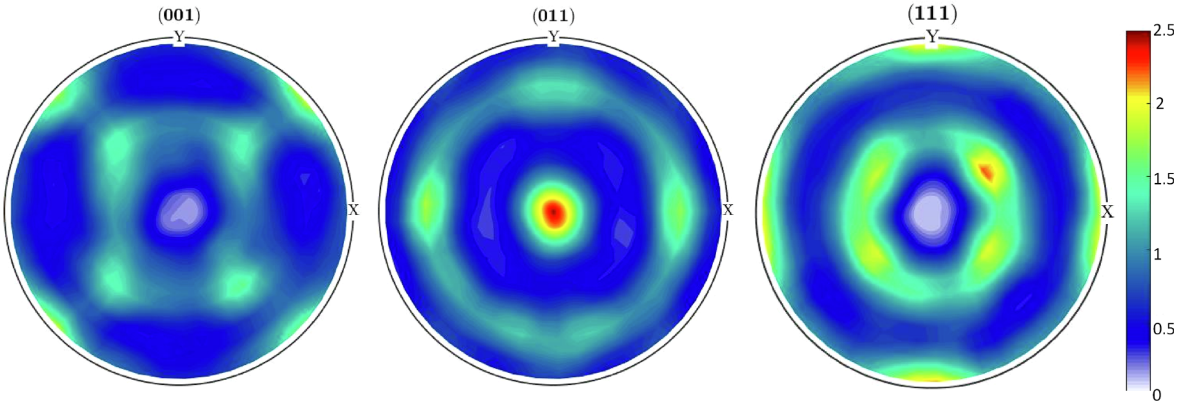


Fig. 10. The crystallographic texture of polycrystalline copper after a true strain of 40% in compression. The initial texture is input as a random texture.

the obtained stress–strain curve with the one reported by Anand and Kothari [23]. The results depict that the stresses obtained by *PRISMS-Plasticity* CPFE software are slightly higher than the results of Anand and Kothari [23]. The slight difference between the current results and with the results reported by Anand and Kothari [23] can be related to differences in the initial sets of orientations. In the next step, the pole figures of the deformed sample are plotted at true strain of 40% using MATLAB toolbox MTEX [31] (Fig. 10). The results show that the simulated pole figures capture the expected features of the FCC sample with initial random grain orientations during the compression test (see, Anand and Kothari [23]). This *PRISMS-Plasticity* CPFE simulation was performed using 16 processors on the PRISMS FLUX cluster with a simulation time (wall time) of 5400 s.

The integration of the *PRISMS-Plasticity* CPFE software with the open source software Neper is demonstrated here in the case of polycrystalline OFHC copper sample with initial random grain orientations. Accordingly, a polycrystalline copper sample consists of 200 randomly oriented grains is generated using Neper. Fig. 11 shows the initial sample, which is generated using Voronoi tessellation scheme using 6750 tetrahedral elements, which were converted to hexahedral elements as described earlier. Two different approaches were used to discretize the grain boundaries; non-conforming and conforming FE discretization schemes. In the case of non-conforming discretization, a regular 32 × 32 × 32 mesh was generated to model the polycrystalline sample. Fig. 12 compares the stress–strain curves predicted by the non-conforming mesh with the corresponding stress-stress curves from Anand and Kothari [23]. It shows that the *PRISMS-Plasticity* CPFE software with non-conforming mesh slightly underpredicts the stress–strain curve as compared to Anand and Kothari [23]. The underprediction of the results can be attributed to the non-conforming discretization used in the simulation. This *PRISMS-Plasticity* CPFE simulation was performed using 64 processors on the PRISMS FLUX cluster with a simulation time (wall time) of 16,200 s.

The conforming FE discretization was generated from the Neper output by converting the tetrahedral elements to hexahedral elements using the tet-to-hex converter which is included as a utility software within the *PRISMS-Plasticity* CPFE code. Fig. 13 (b) shows the microstructure of the conforming mesh in the cases of 28,992 hexahedral elements. The original sample with tetrahedral elements is also presented in Fig. 13 (a) for the sake of comparison. Fig. 14 compares the stress–strain curves obtained from conforming (C) non-conforming (N) meshes versus the results reported by from Anand and Kothari [23]. The results show that the stresses of the sample with the conforming mesh can accurately capture the results reported by Anand and Kothari [23]. The *PRISMS-Plasticity* CPFE simulation for a sample with conforming (C) mesh was performed using 64 processors on the PRISMS FLUX cluster with a simulation time (wall time) of 15,000 s.

CPFE simulation can capture the local microstructural information in addition to the stress–strain curve, i.e., homogenized behavior of a

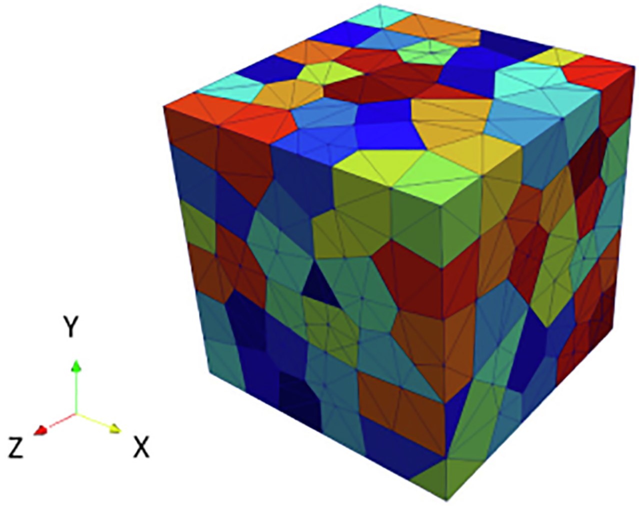


Fig. 11. The virtual microstructures which is generated using Neper consists of a 200 randomly oriented grains with a tetrahedral mesh comprising of 6750 elements.

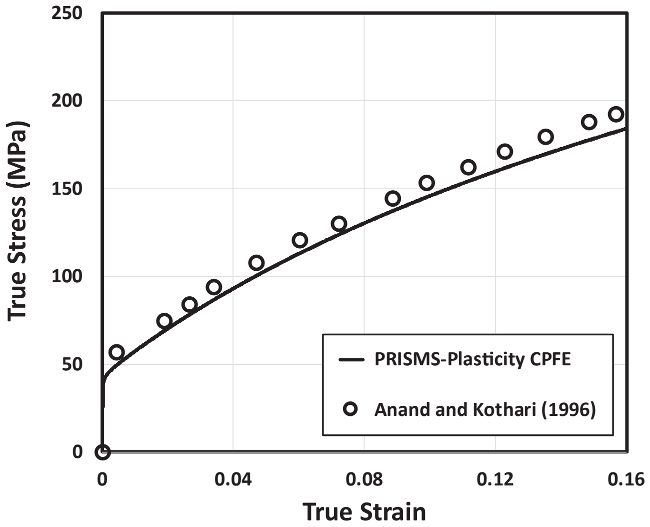


Fig. 12. The stress–strain response of polycrystalline copper sample during compression test obtained by PRISMS-Plasticity CPFE software using virtual microstructures generated with Neper discretized by non-conforming mesh of 32,768 elements. The results of this simulation are compared with those reported by Anand and Kothari [23].

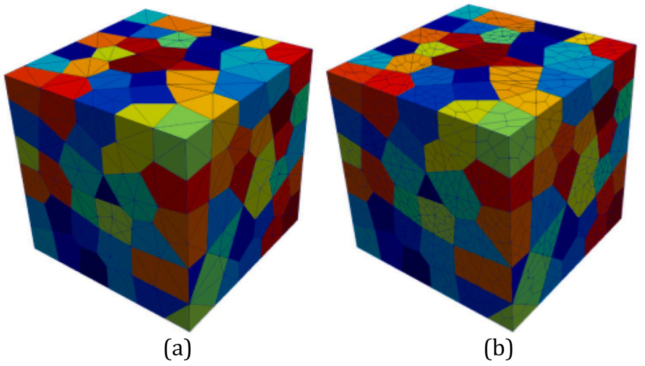


Fig. 13. Virtual microstructures generated with Neper using: (a) 6750 tetrahedral elements (b) 28,992 hexahedral elements.

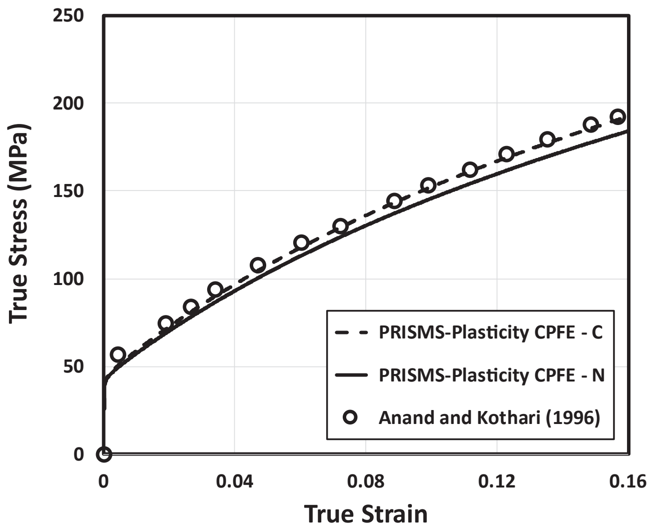


Fig. 14. Comparison the stress–strain response between the non-conforming mesh (N) with 32,768 elements and conforming (C) meshes with 28,992 elements. The results of these simulation are compared with those reported by Anand and Kothari [23].

sample. *PRISMS-Plasticity* CPFE software generates various outputs that can be visualized and processed using ParaView [32], which is an open source post-processing software. As an example, the von Mises equivalent stress and strain fields are demonstrated here which are mapped onto the deformed sample (Fig. 15). The output file can then be processed using ParaView to extract various local information. Some of these aspects such as thresholding, clipping, and slicing of the von Mises equivalent stress field are depicted in Fig. 16. These visualization outputs provide additional information for various local fields inside the microstructure, which can be an important part of a CPFE study.

6.2. BCC polycrystal

In this example, the capability of the *PRISMS-Plasticity* CPFE software to model BCC polycrystalline samples is demonstrated. To do so, the response of a polycrystalline sample of the β titanium alloy Timetal 21S during uniaxial tension is modeled and compared to simulation results reported by Qidwai et al. [33]. In BCC materials, slip may occur on 48 distinct slip systems. These systems represent three families of planes: 12 systems are in the ⟨1 1 1⟩{1 1 0} family, 12 in the ⟨1 1 1⟩{1 1 2} family, and 24 in the ⟨1 1 1⟩{1 2 3} family. Approximate material behavior can be computed by considering a subset of the total number of slip systems. Accordingly, 12 slip systems of ⟨1 1 1⟩{1 1 0} family are considered in this example.

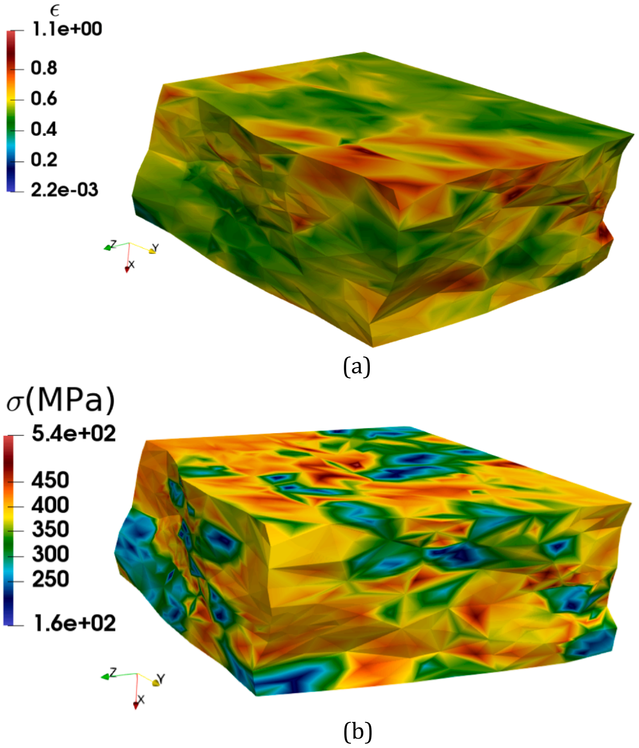
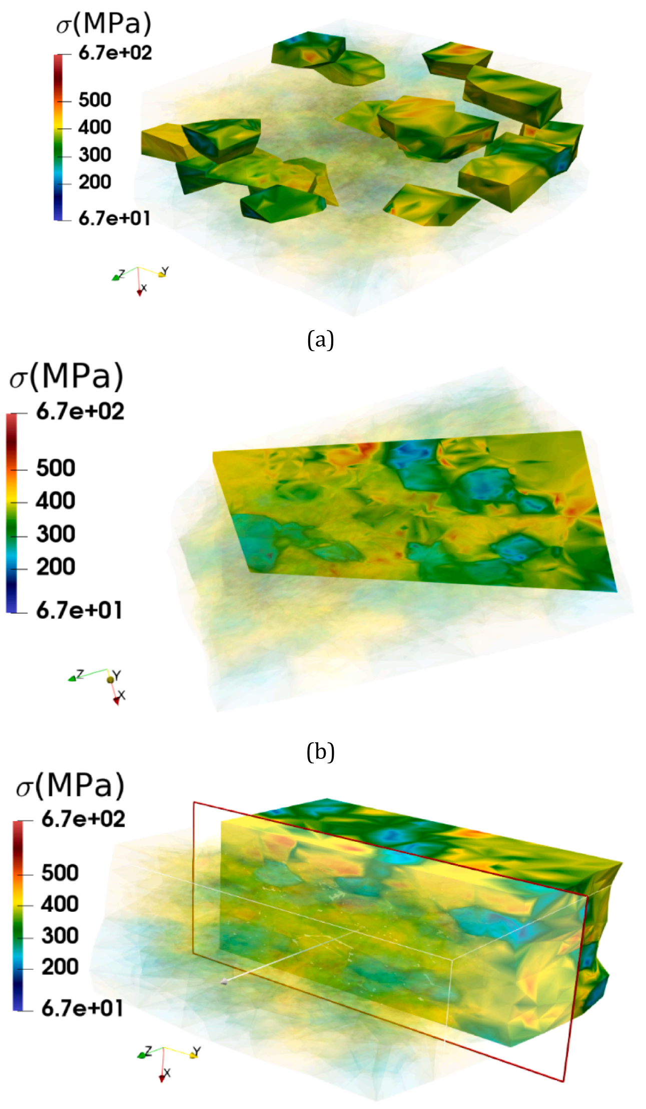


Fig. 15. Visualization of local microstructure information of polycrystalline copper after a true strain of 40% in compression using ParaView: (a) von Mises equivalent strain and (b) von Mises equivalent stress. The microstructure of the sample is generated using Neper and simulated using conforming FE discretization scheme with 28,992 hexahedral elements.

DREAM.3D [20] is incorporated here to generate the initial microstructure. Accordingly, the generated RVE is represented by a 68 × 64 × 46 mesh of microstructural data points, i.e., voxels. The RVE consists of a total of 92 grains, with 16 interior grains. Figs. 17 and 18 present the inverse pole figure and pole figure of the initial microstructure, respectively, which are generated using the MATLAB toolbox MTEX [31]. A structured linear hexahedral mesh (structured grid) is superposed on this voxelated texture output of DREAM.3D such that each hexahedral element of the finite element mesh corresponds to a voxel of the RVE. The total number of elements is 200,192. The ability to read such a voxelated mesh makes it convenient to pass the microstructures generated using DREAM3D as an input to the *PRISMS-Plasticity* CPFE software. Furthermore, in order to study convergence by changing the number of elements, two more samples with different numbers of elements including one smaller with total number of 25,024 and one larger with the total number of 1,601,536 are simulated. Symmetry boundary conditions are imposed on adjacent x, y, and z faces and a tensile displacement is applied in the y direction on the opposite face. The crystal plasticity model parameters of β Titanium are adopted from Qidwai et al. [33], which are $h_0^\beta = h_0 = 1500\text{MPa}$, $s_0^\beta = s_0 = 200\text{MPa}$, $s_s^\beta = S_s = 500\text{MPa}$, and $a^\beta = a = 0.1$ for all slip systems [see Eq. (8)]. The latent hardening value of $q = 1.4$ is used in the *PRISMS-Plasticity* CPFE simulation. Fig. 19 compares the obtained stress–strain curve with the one reported by Qidwai et al. [33]. The results show that the stresses obtained by the *PRISMS-Plasticity* CPFE simulation agrees well with the results of Qidwai et al. [33]. Furthermore, Fig. 20 shows the effect of number of elements on the stress–-strain response of the simulated sample. The results show the convergence of the stress–strain curve with increasing the number of elements. Finally, the von Mises equivalent stress and strain fields are presented in Fig. 21. Simulations were performed using 128 processors on the FLUX cluster with the simulation time (wall time) of 10800,



(a)

(b)

(c)

Fig. 16. Visualization of von Mises equivalent stress of polycrystalline copper after a true strain of 40% in compression using different options of ParaView: (a) threshold (b) slice (c) clip. The microstructure of the sample is generated using Neper and simulated using conforming FE discretization scheme with 28,992 hexahedral elements.

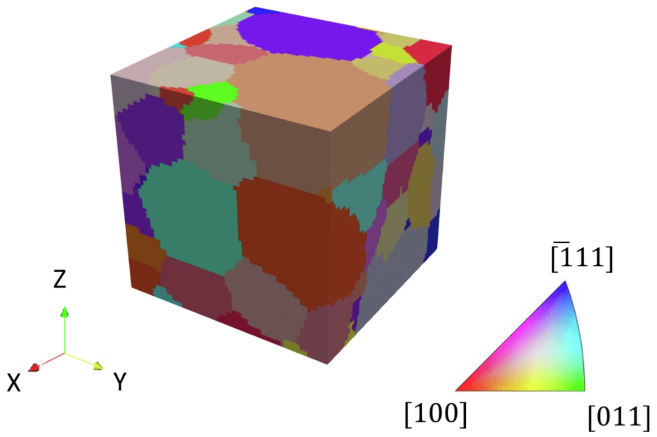


Fig. 17. Inverse pole figure (IPF) showing the initial of β titanium alloy microstructure using MATLAB toolbox MTEX.

39000, and 558,000 s in the cases of problems with 25024, 200192, and 1,601,536 elements, respectively.

6.3. Hcp polycrystal

In the case of HCP samples, the twinning deformation mechanism plays a critical role in accommodating plastic deformation. To demonstrate the twinning model used in *PRISMS-Plasticity* CPFE, simulation results are compared against the experimental results reported by Wu [34] for extruded Mg alloy ZK60A sample at room temperature. The polycrystalline sample is modeled as an aggregate of 1080 single crystals, which reproduces the extruded sample, each grain being modeled by a single eight-node linear hexahedral element. Accordingly, an 8 × 9 × 15 FE cubic mesh is generated in the x, y, and z directions, in which each element represents a single grain. The tensile and compressive displacements are applied in the z direction to mimic the uniaxial tension and compression tests, respectively, while the other face is fixed in all three directions. Fig. 22 presents the pole figures of the initial microstructure that are generated using MATLAB toolbox MTEX [31] and used as the initial texture input for *PRISMS-Plasticity* CPFE simulations. Three types of slip modes including Basal $\langle a\rangle$ ($\{0001\}11\bar{2}0$) Prismatic $\langle a\rangle$ ($\{0001\}11\bar{2}0$), and Pyramidal $\langle c+a\rangle$ ($\{\bar{1}\bar{1}22\}\bar{1}\bar{1}23$), and one extension twin mode ($\{10\bar{1}2\}\bar{1}011$) are considered to model the deformation mechanisms of ZK60A sample. Table 1 presents the selected crystal plasticity parameters for slip and twin systems. The latent hardening parameter of $q = 1$ is used for the crystal plasticity model. Also, the twinning model constants of $A = 0.7$ and $B = 0$ are selected [see Eq. (52)]. These constants are calibrated to accurately capture the length of the plateau and the steepness of the subsequent strain hardening during uniaxial compression test. Fig. 23 shows the stress–strain responses of extruded Mg alloy ZK60A sample during uniaxial tension and compression loadings obtained using the *PRISMS-Plasticity* CPFE simulation compared to the experimental values reported by Wu [34]. The results show that the compressive behavior is precisely captured using the *PRISMS-Plasticity* CPFE software, while the simulated tensile stresses are slightly lower than the experimental values. The slight discrepancy between the experimental and simulation results can be attributed to the difference between the texture used here and the original texture reported by Wu [34]. The simulation can also be easily used to study slip activity. In Fig. 24, the predicted increase in twin volume fraction as the strain increases is shown during the simulated uniaxial tension and compression tests. The results show that the twin volume is approximately 15% at 8% strain in compression, while the twin volume fraction is less than 1% in the case of an 8% uniaxial tensile strain. The simulation results demonstrates the significance of the extension twin system during compression loading. These *PRISMS-Plasticity* CPFE simulations were performed using 16 processors on the PRISMS FLUX cluster with a simulation time (wall time) of 1800 s.

7. Scaling

One of the important aspects of a parallel code is the capability to scale to a large number of degrees of freedom using multiple processors while solving problems in practicable times. Especially, in the case of CPFE simulation, problems with a few million degrees of freedom are common. Accordingly, the scalability of the CPFE software has a key role in its performance. In the case of *PRISMS-Plasticity* CPFE software, the solver is implemented on top of the deal.II [19] open source finite element library. It has been previously shown that this library is capable of scaling to tens of thousands of processors [35]. Both strong and weak scaling studies were conducted for the *PRISMS-Plasticity* CPFE software. The strong scaling study consisted of determining the simulation time (wall time) for a fixed problem size, i.e., number of elements, as the number of processors increases. The weak scaling study consisted of determining the simulation time (wall time) for a fixed

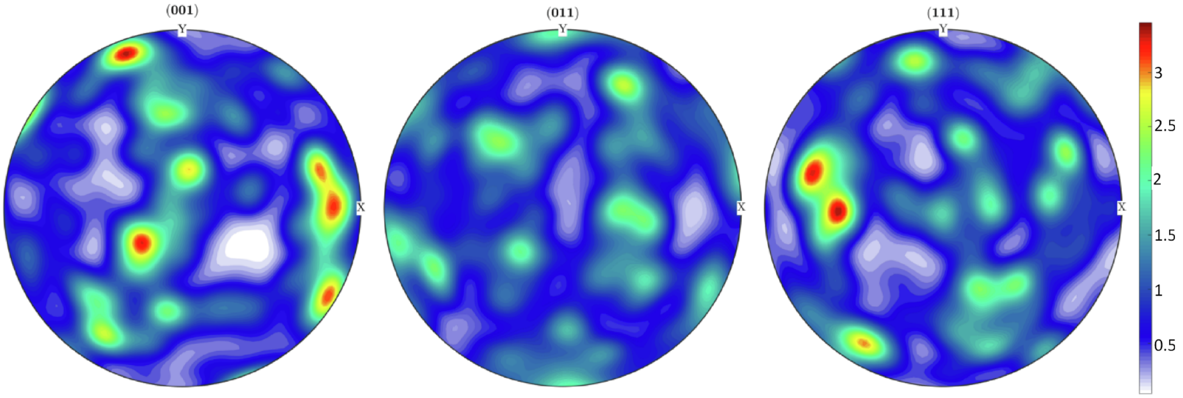


Fig. 18. Pole figures showing the initial microstructure for β titanium alloy. The pole figures were generated using the MATLAB toolbox MTEX.

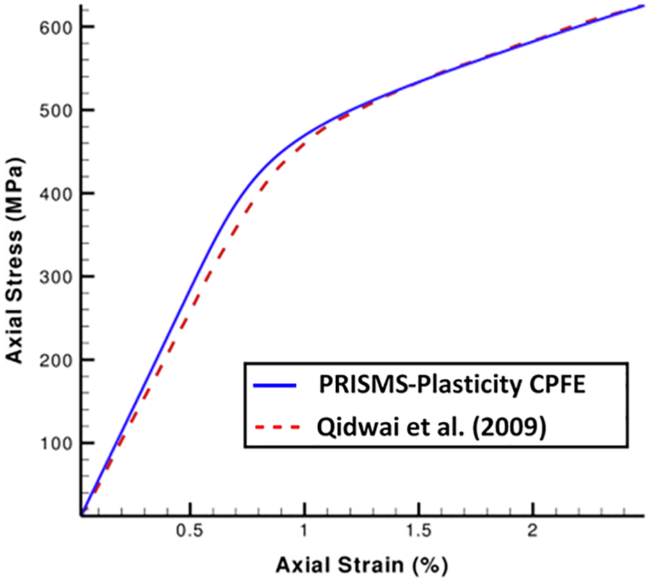


Fig. 19. The stress–strain response of polycrystalline sample of β titanium alloy Timetal 21S during uniaxial tension obtained by PRISMS-Plasticity CPFE software. These are compared with the simulation results for this microstructure reported by Qidwai et al. [33].

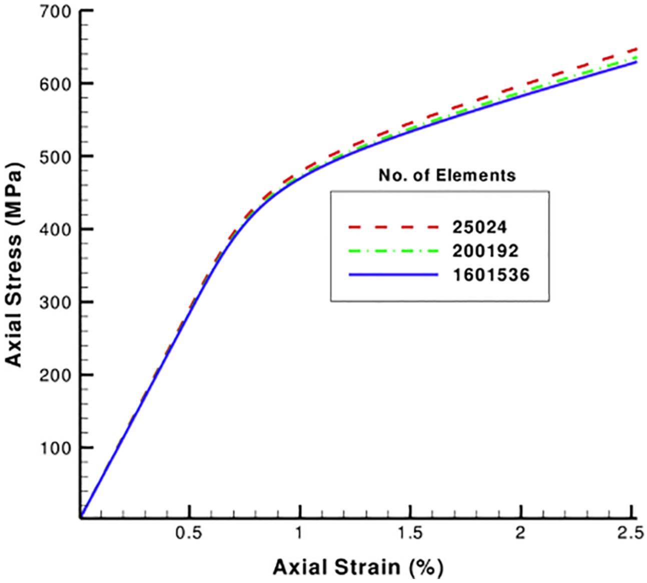


Fig. 20. Stress–strain response of polycrystalline sample of β titanium alloy Timetal 21S during uniaxial tension with different mesh densities. Note the convergence of stress–strain curves as the number of elements increases to 200,192 and greater.

number of elements per processor as the problem size increases, i.e., total number of elements increases. The simulations are performed on the PRISMS FLUX cluster at the University of Michigan across multiple nodes equipped with Intel Xeon processors and 4 GB memory per processor. The solution to an FEM problem consists of assembly, solve, and updating the variables. Assembly is the procedure in which the constitutive model is solved, algorithmic tangent modulus is calculated, and the global stiffness matrix and force vector are assembled. The solve process includes solving the assembled linear system to compute the displacements across the nodes. Finally, the constitutive model variables, displacements, and deformation gradient tensors are updated using the obtained solution. Solving the linear system is performed using the iterative solvers provided by deal.II library [19], and the scalability of this library has been demonstrated by Bangerth et al. [35]. Accordingly, the scaling study is conducted here on the code components dealing with assembly and updating the variables. The simulations are performed on polycrystalline OFHC copper sample consists of 100 randomly orientated grains during a compression experiment. The material properties are selected similar to Anand and Kothari [23] as described in section 6.1. The largest number of elements is 2,097,152, which is a regular 128 × 128 × 128 grid of eight-node linear hexahedral elements. The total equivalent strain of 5.2% is applied on the sample with constant imposed velocity gradient on the boundaries. The strong scaling study was performed on the sample with 2,097,152 elements using 64, 128, 256, and 512 processors. Fig. 25(a) shows the results of strong scaling study and demonstrates that linear scaling is obtained for both assembly and updating the variables. The results show that the software performance scales well and large CPFE problems can be readily solved using *PRISMS-Plasticity* by increasing the number of processors up to at least 500 processors. The weak scaling study was performed with a 32 × 32 × 32 grid on 8 processors, 64 × 64 × 64 grid on 64 processors, and 128 × 128 × 128 grid on 512 processors as depicted in Fig. 25(b). The weak scaling shows strong performance for both assembly and updating the variables in which the wall time does not change as the problem size increases. However, the total time increases as the problem size increases which is attributed to the communication overheads involved in the solve procedure for large problems.

8. Future developments and ongoing research

The *PRISMS-Plasticity* 3-D CPFE software package is constantly under development to enhance both modelling accuracy and efficiency and new versions are released annually. To date, four versions of the code have been released. The structure of the code is continuously and

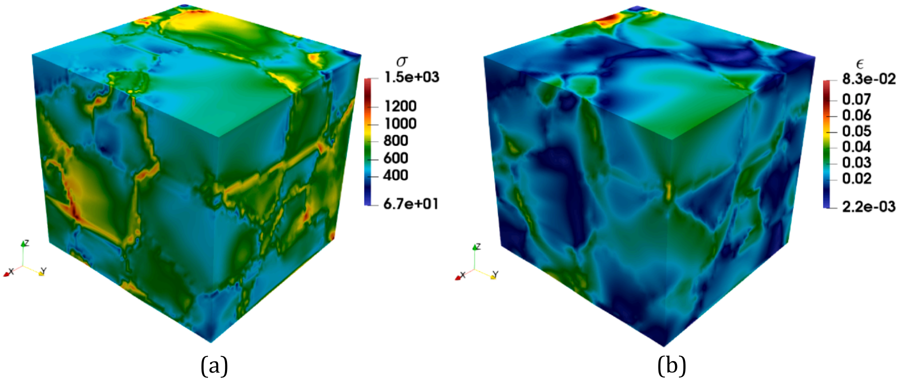


Fig. 21. Visualization of local microstructure information of β titanium alloy Timetal 21S after a strain of 2.5% in tension using ParaView: (a) von Mises equivalent strain and (b) von Mises equivalent stress. The microstructure of the sample is generated using DREAM.3D and discretized using 200,192 hexahedral elements.

professionally being updated. Major upgrades to the code are planned and released on an annual basis in the summer prior to the annual PRISMS Center workshop [4]. The 2019 release of the software will be capable of supporting multiple material constitutive models in a systematic way. The rate dependent model will be added to the software to capture the effect of temperature and strain rate during the crystal plasticity simulation of polycrystalline metallic samples. The rate-dependent implementation will be based on the work of Kalidindi et al. [36]. It will incorporate the effect of temperature and strain rate. The current version of software can capture the isotropic hardening of slip systems. Kinematic hardening implementation presented by Staroselsky and Cassenti [37] is being added to the software to model the material response under cyclic loadings. In addition, a revised twinning model is under development to capture the cyclic response of HCP polycrystalline samples. The model can capture the detwinning as well as twinning and is based on the work of Wang et al. [38]. Wang et al. [38] introduced the Twinning–DeTwinning (TDT) model in which they implemented the model in an elastic viscoplastic self-consistent (EVPSC) polycrystal model. This approach will be included in a future version of *PRISMS-Plasticity* to capture the twinning and detwinning mechanisms. The modelling of multiphase materials is another aspect which is planned to be implemented in the software. Accordingly, the software will be able to capture the samples with arbitrary number and distribution of phases. This will contribute to the development of an integrated framework for an automated alloy design procedure. Longer term planned enhancements include capturing grain size effects on material response using either a micro Hall-Petch model [39] or constitutive modeling based on the concept of geometrically necessary dislocations [40]. In addition, *PRISMS-Plasticity* CPFE software will be integrated with *PRISMS-PF* Phase-Field software to simulate phenomena such as dynamic recrystallization. An important feature of the *PRISMS-Plasticity* CPFE software is the ease of integration with the results from lower length scale simulations such as discrete dislocation dynamics (DDD) and molecular dynamics (MD) as depicted in Fig. 1 [4]. This integration is a continual process with current efforts focused on calibration of the crystal plasticity slip activity parameters using the results of DDD [41]. Finally, a fracture model is under development to capture the full deformation response from elastic to plastic deformation followed by damage nucleation and propagation, which eventually leads to fracture [42,43].

Table 1
Crystal plasticity parameters of Mg alloy ZK60A for different slip and twin system.

Mode	s_0^α(MPa)	h_0^α(MPa)	s_s^α(Mpa)	θ^α
Basal	45	150	95	0
Prismatic	105	150	135	6
Pyramidal $< c+a >$	210	150	420	6
Twinning	50	10	220	0.75

9. Summary

An open source parallel 3-D crystal plasticity finite element (CPFE) software package *PRISMS-Plasticity* is presented here as a part of PRISMS integrated framework. The software can be effectively integrated with experimental characterization techniques such as electron backscatter diffraction (EBSD) and synchrotron X-ray diffraction using available open source software packages of DREAM.3D [20] and Neper [21]. The integration of *PRISMS-Plasticity* software with an information

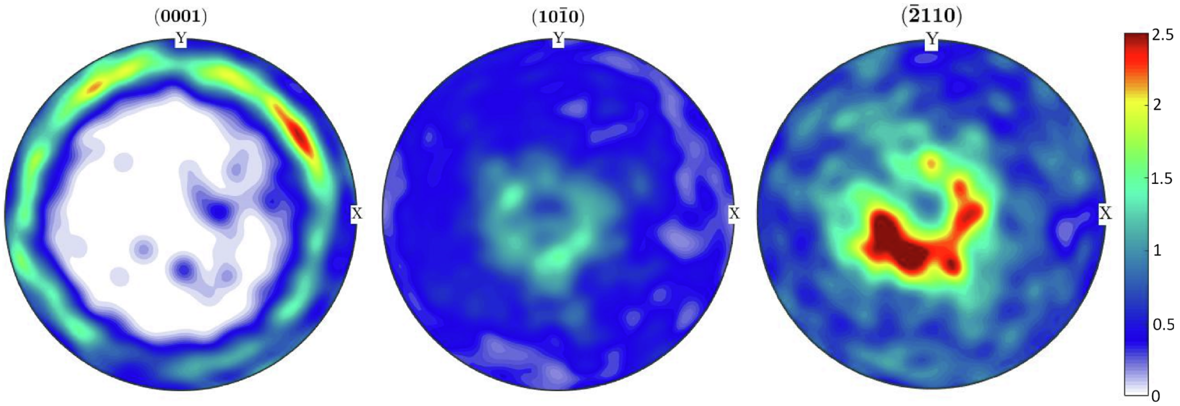


Fig. 22. Pole figures showing the initial texture of the extruded sample with the basal poles aligned perpendicular to the loading direction. LD: Loading Direction (Extrusion Direction); RD: Radial Direction (normal to the loading direction).

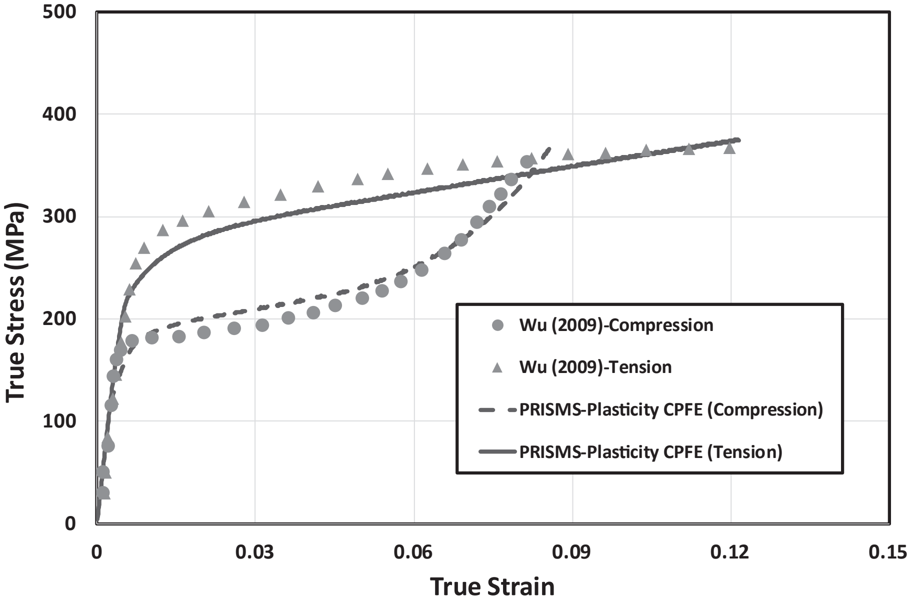


Fig. 23. The stress–strain responses of Mg alloy ZK60A sample during uniaxial tension and compression loadings obtained by PRISMS-Plasticity CPFE software. The *PRISMS-Plasticity* simulation results are compared with experimental values reported by Wu [34].

repository, the Materials Commons, is also demonstrated. The *PRISMS-Plasticity* crystal plasticity model includes a constitutive model, numerical integration, and the derivation of consistent tangent modulus. A twinning model is also incorporated in the software using an integration point sensitive PTR scheme [28].

A scaling study of the *PRISMS-Plasticity* CPFE software is presented which demonstrates the efficiency of this highly parallelized code. The results show that the software performance can be readily scaled up to hundreds of processors or more. A set of numerical examples for polycrystalline samples with different crystal structures of FCC, BCC, and HCP are simulated and compared against the available results in literature. Problem sizes of up to 6 million degrees of freedom are simulated demonstrating the scalability and efficiency of the software. The visualization procedure of the simulation results is also presented using open source software of ParaView [32] to demonstrate the capability of the code to capture common aspects of the crystal plasticity simulation.

CRediT authorship contribution statement

Mohammadreza Yaghoobi: Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Validation, Visualization, Writing - original draft, Writing - review & editing. **Sriram Ganesan:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing - original draft, Writing - review & editing. **Srihari Sundar:** Formal analysis, Investigation, Methodology, Validation, Writing - original draft, Writing - review & editing. **Aaditya Lakshmanan:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing - original draft, Writing - review & editing. **Shiva Rudraraju:** Methodology, Writing -

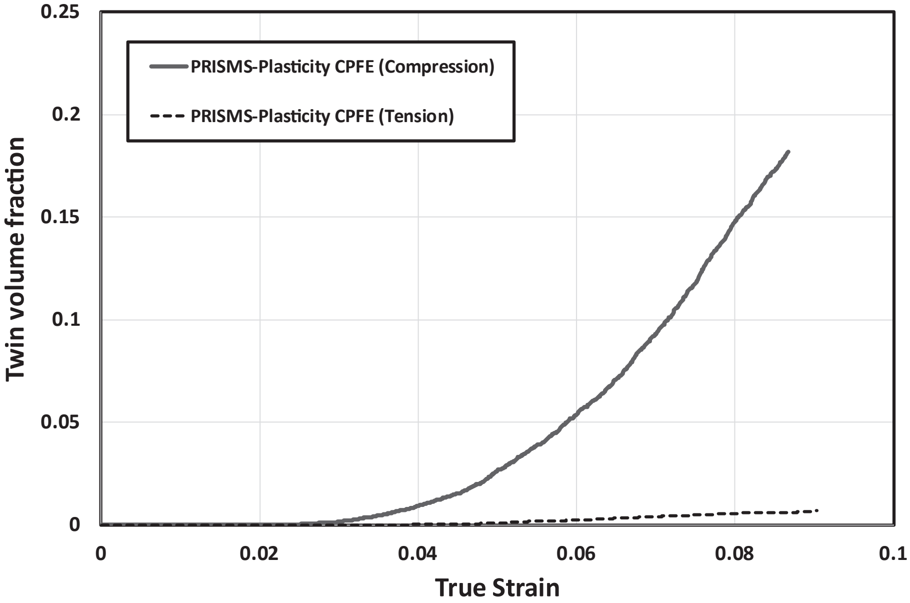
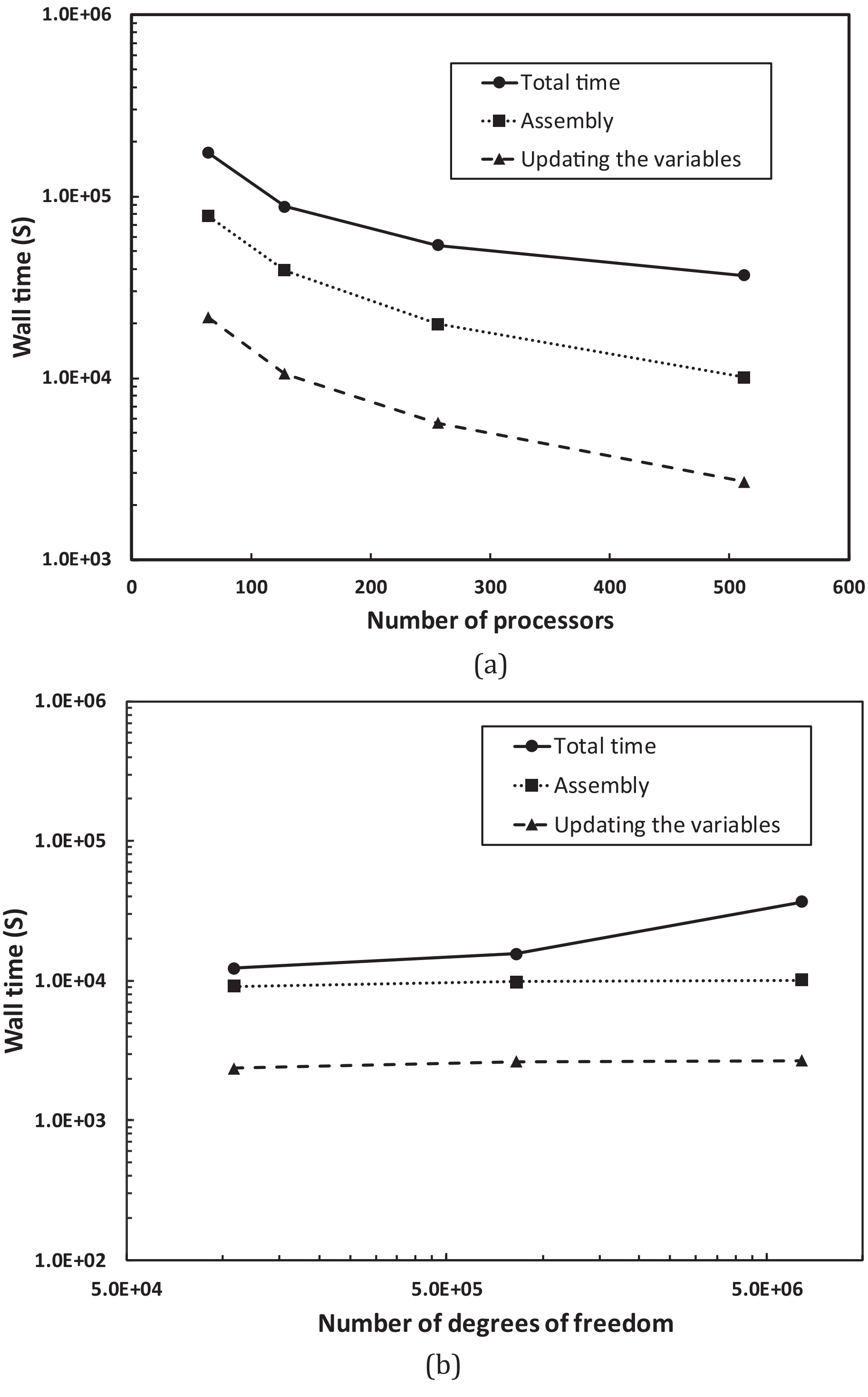


Fig. 24. PRISMS-Plasticity CPFE simulation results showing the variation of twin volume fraction versus the true strain for the Mg alloy ZK60A during uniaxial tension and compression loadings.



(a)

(b)

Fig. 25. Scaling study for PRISMS-Plasticity CPFE software: (a) strong scaling (b) weak scaling.

review & editing. **John E. Allison:** Conceptualization, Project administration, Methodology, Funding acquisition, Supervision, Writing - review & editing. **Veera Sundararaghavan:** Conceptualization, Project administration, Formal analysis, Investigation, Methodology, Supervision, Writing - review & editing.

Acknowledgment

This work was supported by the U.S. Department of Energy, Office of Basic Energy Sciences, Division of Materials Sciences and Engineering under Award#DE-SC0008637 as part of the Center for Predictive Integrated Structural Materials Science (PRISMS Center) at University of Michigan. We also acknowledge the financial cost-share support of University of Michigan College of Engineering and Office of the Vice President for Research. The authors acknowledge helpful discussions and efforts of Stephen DeWitt, Terry Weymouth, Brian Puchala, and Glenn Tarcea on the integration of *PRISMS-Plasticity* CPFE software with Materials Commons.

Appendix A

- *PRISMS-Plasticity* software can be freely downloaded from the following link: https://github.com/prisms-center/plasticity.
- prismscpfe_mcapi Python package can be freely downloaded from the following link: https://github.com/prisms-center/prismscpfe_mcapi.
- A DREAM.3D pipeline is built to generate input files for *PRISMS-Plasticity* CPFE simulation from a typicalexperimental EBSD file, which can be freely downloaded from the following link:

https://github.com/prisms-center/PRISMS-Plasticity_Training_Materials.

- Input files for the CPFEM simulations presented in the section 6 can be downloaded from Materials Commons using the following link:

https://materialscommons.org/mcapp/#/data/dataset/8125b518-4955-4a5f-b887-24bc5544af38.

Also, the dataset is published for these input files at Materials Commons at https://doi.org//10.13011/m3-fh45-7w84.

References

[1] J. Allison, D. Backman, L. Christodoulou, Integrated computational materials engineering: a new paradigm for the global materials profession, JOM 58 (2006) 25–27.

[2] J. Allison, M. Li, C. Wolverton, X.M. Su, Virtual aluminum castings: an industrial application of ICME, JOM 58 (2006) 28–35.

[3] J. Allison, Integrated computational materials engineering: a perspective on progress and future steps, JOM 63 (2011) 15–18.

[4] L.K. Aagesen, J.F. Adams, J.E. Allison, W.B. Andrews, V. Araullo-Peters, PRISMS: an integrated, open-source framework for accelerating predictive structural materials science, JOM 70 (2018) 2298–2314.

[5] F. Roters, P. Eisenlohr, L. Hantcherli, D.D. Tjahjanto, T.R. Bieler, D. Raabe, Overview of constitutive laws, kinematics, homogenization and multiscale methods in crystal plasticity finite-element modeling: theory, experiments, applications, Acta Mater. 58 (2010) 1152–1211.

[6] G.I. Taylor, Plastic strain in metals, J. Inst. Metals 62 (1938) 307–324.

[7] H. Honeff, H. Mecking, Analysis of the deformation texture at different rolling conditions, in: S. Nagashima (Ed.), Proc. ICOTOM 6, The Iron and Steel Institute of Japan, 1981, pp. 347–355.

[8] U.F. Kocks, H. Chandra, Slip geometry in partially constrained deformation, Acta Metall. 30 (1982) 695–709.

[9] P. Van Houtte, On the equivalence of the relaxed Taylor theory and the Bishop-Hill theory for partially constrained plastic deformation of crystals, Mat. Sci. Eng. 55 (1982) 69–77.

[10] P. Van Houtte, L. Delannay, I. Samajdar, Quantitative prediction of cold rolling textures in low-carbon steel by means of the LAMEL model, Texture Microstruct. 31 (1999) 109–149.

[11] P. Van Houtte, L. Delannay, S.R. Kalidindi, Comparison of two grain interaction models for polycrystal plasticity and deformation texture prediction, Int. J. Plasticity 18 (2002) 359–377.

[12] L. Delannay, S.R. Kalidindi, P. Van Houtte, Quantitative prediction of textures in aluminum cold rolled to moderate strains, Mat. Sci. Eng. A 336 (2002) 233–244.

[13] M. Crumbach, M. Goerdeler, G. Gottstein, L. Neumann, H. Aretz, R. Kopp, Through-process texture modelling of aluminium alloys, Modell. Simul. Mater. Sci. Eng. 12 (2004) S1–S18.

[14] M. Crumbach, M. Goerdeler, G. Gottstein, Modelling of recrystallisation textures in aluminum alloys: I. Model set-up and integration, Acta Mater. 54 (2006) 3275–3289.

[15] M. Crumbach, M. Goerdeler, G. Gottstein, Modelling of recrystallisation textures in aluminum alloys: II. Model performance and experimental validation, Acta Mater. 54 (2006) 3291–3306.

[16] D.D. Tjahjanto, P. Eisenlohr, F. Roters, A novel grain cluster-based homogenization scheme, Modell. Simul. Mater. Sci. Eng. 18 (2010) 015006.

[17] H. Moulinec, P. Suquet, A numerical method for computing the overall response of nonlinear composites with complex microstructure, Comput. Methods Appl. Mech. Eng. 157 (1998) 69–94.

[18] R.A. Lebensohn, N-site modeling of a 3D viscoplastic polycrystal using Fast Fourier Transform, Acta Mater. 49 (2001) 2723–2737.

[19] W. Bangerth, R. Hartmann, G. Kanschat, deal. II-A general purpose object-oriented finite element library, ACM Trans Math. Softw. (TOMS) 33 (2007) 24.

[20] M.A. Groeber, M.A. Jackson, DREAM 3D: a digital representation environment for the analysis of microstructure in 3D, Integ, Mater. Manuf. Innov. 3 (2014) 5.

[21] R. Quey, P. Dawson, F. Barbe, Large-scale 3D random polycrystals for the finite element method: generation, meshing and remeshing, Comput. Methods Appl. Mech. Eng. 200 (2011) 1729–1745.

[22] B. Puchala, G. Tarcea, E.A. Marquis, M. Hedstrom, H.V. Jagadish, J.E. Allison, The Materials Commons: a collaboration platform and information repository for the global materials community, JOM 68 (2016) 2035.

[23] L. Anand, M. Kothari, A computational procedure for rate independent crystal plasticity, J. Mech. Phys. Solids 44 (1996) 525–558.

[24] V. Sundararaghavan, N. Zabaras, A multi-length scale sensitivity analysis for the control of texture-dependent properties in deformation processing, Int. J. Plasticity 24 (2008) 1581–1605.

[25] A. Kumar, V. Sundararaghavan, Simulation of magnetostrictive properties of Galfenol under thermomechanical deformation, Finite Elem. Anal. Des. 127 (2017) 1–5.

[26] P. Acar, A. Ramazani, V. Sundararaghavan, Crystal plasticity modeling and experimental validation with an orientation distribution function for Ti-7Al alloy, Metals 7 (2017) 459.

[27] A. Staroselsky, L. Anand, A constitutive model for hcp materials deforming by slip and twinning: application to magnesium alloy AZ31B, Int. J. Plasticity 19 (2003) 1843–1864.

[28] C.N. Tomé, R.A. Lebensohn, U. Kocks, A model for texture development dominated by deformation twinning: application to zirconium alloys, Acta Metall. 39 (1991) 2667–2680.

[29] P. Van Houtte, Simulation of the rolling and shear texture of brass by the Taylor theory adapted for mechanical twinning, Acta Metall. 26 (1978) 591–604.

[30] J.W. Christian, S. Mahajan, Deformation twinning, Prog. Mater Sci. 39 (1995) 1–157.

[31] F. Bachmann, R. Hielscher, H. Schaeben, Texture analysis with MTEX-free and open source software toolbox, Solid State Phenom. 160 (2010) 63–68.

[32] A. Henderson, ParaView guide, A Parallel Visualization Application, Kitware Inc., 2007.

[33] M.S. Qidwai, A.C. Lewis, A.B. Geltmacher, Using image-based computational modeling to study microstructure-yield correlations in metals, Acta Mater. 57 (2009) 4233–4247.

[34] L. Wu, Mechanical behavior and the role of deformation twinning in wrought magnesium alloys investigated using neutron and synchrotron X-ray diffraction, Ph.D. Thesis The University of Tennessee, Knoxville, 2009.

[35] W. Bangerth, C. Burstedde, T. Heister, M. Kronbichler, Algorithms and data structures for massively parallel generic adaptive finite element codes, ACM Trans. Math. Softw. 38 (2011) 14.

[36] S.R. Kalidindi, C.A. Bronkhorst, L. Anand, Crystallographic texture evolution during bulk deformation processing of fcc metals, J. Mech. Phys. Solids 40 (1992) 537–569.

[37] A. Staroselsky, B.N. Cassenti, Combined rate-independent plasticity and creep model for single crystal, Mech. Mater. 42 (2010) 945–959.

[38] H. Wang, P.D. Wua, C.N. Tomé, J. Wang, A constitutive model of twinning and detwinning for hexagonal close packed polycrystals, Mat. Sci. Eng. A 555 (2012) 93–98.

[39] G. Proust, C.N. Tomé, A. Jain, S.R. Agnew, Modeling the effect of twinning and detwinning during strain-path changes of magnesium alloy AZ31, Int. J. Plasticity 25 (2009) 861–880.

[40] W.A. Counts, M.V. Braginsky, C.C. Battaile, E.A. Holm, Predicting the Hall-Petch effect in fcc metals using non-local crystal plasticity, Int. J. Plasticity 24 (2008) 1243–1263.

[41] M.C. Messner, M. Rhee, A. Arsenlis, N.R. Barton, A crystal plasticity model for slip in hexagonal close packed metals based on discrete dislocation simulations, Modell. Simul. Mater. Sci. Eng. 25 (2017) 044001.

[42] A. Spear, Numerical and experimental studies of three-dimensional crack evolution in aluminum alloys: macroscale to microscale, PhD thesis Cornell University, 2014.

[43] J.D. Clayton, Dynamic plasticity and fracture in high density polycrystals: constitutive modeling and numerical simulation, J. Mech. Phys. Solids 53 (2005) 261–301.