Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 393 (2022) 114831

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

A robust algorithm for rate-independent crystal plasticity

T. Mánik*, H.M. Asadkandi, B. Holmedal

Department of Materials Science and Engineering, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway

Received 20 December 2021; received in revised form 23 February 2022; accepted 28 February 2022
Available online 25 March 2022

Abstract

A new stable return-mapping algorithm enables crystal-plasticity solutions by using a regularized yield surface with very large exponents, for which the rate-independent limit of the Schmid assumption in practice is reached. Numerical stability is enabled by an improved initial guess for the stress solution and by applying a line search for each Newton iteration. A hypo-elastic–plastic corotational formulation is chosen, where the tensors are contracted in a way that naturally degenerate to the rigid plastic formulation. The consistent algorithmic tangent modulus is derived, and a fast and very stable open-source implicit implementation into a finite element software is explained and demonstrated for simulations of the necking of a single crystal and for deformation of a polycrystalline representative volume element. The simulations run stable allowing large time steps. Hence, the simulation times are significant shorter than for explicit finite element simulations. The framework enables use of arbitrary types of slip systems. As a demonstration, the importance and interpretation of the yield surface exponent and the asymptotic limit of very large exponent are discussed for bcc crystals with $\left\{1\bar{1}0\right\}\langle 111\rangle$, $\left\{\bar{1}2\bar{1}\right\}\langle 111\rangle$ and $\left\{1\bar{3}2\right\}\langle 111\rangle$ slip systems.

Keywords: Crystal plasticity; Rate independent; Return mapping algorithm; Line search; Finite element method; Open-source code

1. Introduction

The detailed modeling of metal plasticity, e.g. by finite element codes, requires a crystal-plasticity framework that can efficiently incorporate arbitrary slip systems as well as account for crystal elasticity. In practice, compromises must be made between model complexity and calculation time. This work will be limited to the formulation and demonstration of a stable numerical algorithm for a rate-independent crystal plasticity, without complex latent hardening of the slip systems [1], and without non-Schmid effects, see e.g. Soare [2]. However, the framework is not limited to these simplifications.

The starting point for rate-independent crystal-plasticity theories is the Schmid assumption, [3]. Mathematically, this can be expressed as a multi-surface formulation with one yield criterion for each shear stress τ_α, with a corresponding critical resolved shear stress, τ_{c_α}, on each slip system, α.

$$\tau_\alpha \leq \tau_{c_\alpha} \tag{1}$$

With infinitely many slip systems, the criteria would correspond exactly to the isotropic Tresca yield criterion. However, slip is restricted to certain crystallographic planes and the densely packed atom directions in the crystal.

* Corresponding author.
E-mail address: tomas.manik@ntnu.no (T. Mánik).

https://doi.org/10.1016/j.cma.2022.114831

Hence, the yield stress of a crystal is highly anisotropic, where the inner envelope of the Schmid criteria defines a crystal yield surface with sharp corners.

At sufficiently low temperatures in densely packed crystal structures, dislocation glide will occur on the densely packed atom planes, due to their low Peierls barrier. However, at elevated temperatures the Peierls–Nabarro contribution to the critical resolved shear stress for glide on non-densely packed slip planes decreases, enabling slip to occur also on these more narrowly spaced slip planes. However, in non-densely packed structures without densely packed slip planes, e.g. ferritic steel with BCC structure, several slip systems with their respective critical resolved shear stress, will compete, even at room temperature. Employing additional slip systems results in a more complex crystal yield surface with more corners and facets. In hcp metals there are only three slip systems available in the densely packed planes, hence also here slip in less densely packed planes must be included in a model [4,5]. Furthermore, twinning can be incorporated as pseudo-slip systems [6,7]. To handle mathematically all mentioned cases, critical resolved shear stresses for slip to occur on different slip planes will be distinguished and be allowed to have individual work hardening in the model considered in this work.

The rule of normality i.e. associated flow, where the plastic rate-of-deformation is normal to the yield surface, holds for each facet on the crystal yield surface. However, in the rate-independent limit, several solutions meet in a yield-surface corner, and the Taylor ambiguity occurs. In the rate-independent theories, a variety of ambiguity solutions have been suggested, as discussed in the review by Mánik and Holmedal [8].

In general, the critical resolved shear stress depends on the shear rates of the slip systems, but at room temperature the strain-rate sensitivity is low, and the metal is commonly assumed to be rate-independent, which will be assumed in this work. However, the rate-independent models will always be simplifications of rate-sensitive models, hence it is important to understand the simplifications being made. Firstly, it is important to distinguish the instant strain-rate sensitivity from the strain-rate sensitivity that influences the work hardening and consequently needs a certain amount of strain to change the critical resolved shear stress. The latter can be captured by a rate-insensitive model.

The origin of the instant strain-rate sensitivity is that the critical shear stress of a slip system depends on the shear rate of the same slip system. The popular viscoplastic power law [9,10] in Eq. (2) is an example, for recent CPFEM applications, see e.g. [11–15].

$$\tau_{c_\alpha} = \tau_\alpha^0 \left(\frac{|\dot{\gamma}_\alpha|}{\dot{\gamma}_0} \right)^m \tag{2}$$

In this model there is no threshold for the critical resolved shear stress, and $\tau_\alpha = \tau_{c_\alpha}$ is the resolved shear stress. Furthermore, τ_α^0 and $\dot{\gamma}_0$ are constants (that can be strain dependent) and m is the instant strain-rate sensitivity. For the CPFEM with the viscoplastic model, efficient implementations [16–22], comparison studies of different algorithms [23–25] and an extensive review [26] have been reported. Even higher numerical efficiency has been achieved by spectral solvers utilizing the fast Fourier transform, but then limited to cyclic boundary conditions [27–33]. However, the viscoplastic model equations are increasingly difficult to solve numerically for small values of the strain rate sensitivity, m. So far, most of the numerical algorithms for solving the viscoplastic equations have not been optimized for dealing with the rate independent limit $m \to 0$. The speed and stability of the CPFEM and spectral implementations worsen as m decreases. One approximation used to deal with small m, is to first perform expensive calculations for the crystal in the crystal coordinate system, and then map these solutions by a spectral representation, as suggested by Knezevic, Al-Harbi and Kalidindi [34] and applied in CPFEM by Zecevic, McCabe and Knezevic [35]. Another approach to deal with arbitrarily small m, was suggested by Knezevic, Zecevic, Beyerlein and Lebensohn [36], by first calculating solutions by a relatively large $m \approx 0.05$, and then obtain solutions for lower values of m by a scaling relation that applies to the viscoplastic law. As pointed out by Mánik and Holmedal [37] for the case of 12 slip systems in fcc, the slip systems that are most activated, in most cases do not change for $m \leq 0.1$. Since the 56 corner solutions in fcc are quite well separated, this method works for most combinations of a given strain path and grain orientation. However, when more slip systems are activated, e.g. in bcc, some of the corners will disappear with lower values of m. Anyhow, even with $m = 0.05$, the time steps that can be made by implicit finite element integration are limited, and the line-search and choice of the initial guess, as proposed in the current work, will be very beneficial.

The instant strain-rate sensitivity, m, influences the crystal yield surface in two different ways. Firstly, it rounds off the corners of the crystal yield surface, which for rate-dependent models can be defined as the iso-surface with constant internal work and internal work rate. Note, that even when the strain-rate sensitivity is very low, say at room temperature, the round-off of the corners may still be significant. Secondly, the rate dependency causes the crystal yield surface to expand with increasing strain rate. At room temperature this expansion is very weak for most metals and justifies the use of a rate-independent yield surface.

However, the rounded corners must still be accounted for, as argued by Holmedal [38]. Eq. (2) can be derived from a potential [38,39] proportional to the internal work rate and shape invariant for different internal work rates. As pointed out by Holmedal [38], an iso value of this potential corresponds to the regularized yield surface proposed by Gambin and Arminjon [40–44]. Since the plastic strain rate equals the gradient of the potential, the associated flow rule then applies. Hence, in the limit of $m \to 0$, this model degenerates to a solution of the rate-independent Schmid model. However, since the Schmid model suffers from the Taylor ambiguity, this rate-independent solution is not unique. The physical interpretation of that is that different models for the strain-rate dependency gives different ambiguity solutions.

To avoid the Taylor ambiguity and the corresponding singularities, the regularization by Gambin/Arminjon has been applied in many rate-independent crystal-plasticity finite element model (CPFEM) implementations [45–47]. Alternatively, a regularized yield surface, based on the approach by Kreisselmeier and Steinhauser [48], has also been commonly applied [49–54]. Note that in the limit of large yield surface exponents, this approach becomes similar as the yield surface by Gambin and therefor degenerates to the same Taylor ambiguity solution, i.e. the one corresponding to the viscoplastic power law in Eq. (2).

Another rate-dependent model is the viscous over-stress model.

$$\tau_\alpha = \hat{\tau}_\alpha^0 + \eta^{-1} |\dot{\gamma}_\alpha| \tag{3}$$

Here $\hat{\tau}_\alpha^0$ is a true athermal yield stress and η^{-1} is the viscosity of the metal. In the limit $\eta \to \infty$ this model degenerates to a solution of the rate-independent Schmid model. Note, that this corresponds to another ambiguity solution than the viscoplastic power law in Eq. (2). Implementations have been made by Schmidt-Baldassari [55] and by [55,56], in which a minimization of an augmented Lagrangian is used to approximately obtain the rate-independent limit ($\eta \to \infty$) of the viscous assumption in Eq. (3). Following the arguments by Mánik and Holmedal [8], this ambiguity solution must correspond to the one obtained by either quadratic programming [8,57] or by singular value decomposition. [58,59].

Not all ambiguity solutions have a physical interpretation. Many are simply efficient mathematical means to obtain a well-posed, non-singular mathematical problem. For an overview of various approaches, the reader is referred to recent reviews, [37,56,60].

In models for predicting texture evolution during fabrication, e.g. rolling or extrusion, rate-independent statistical aggregate models are useful, i.e. the classical full-constraint Taylor model [61], and the more recent advanced Taylor type models; the ALamel model [62], the ALamel3 model [8] the GIA/RGC model [63,64] and rate-independent self-consistent models, e.g. [65–68]. In texture-prediction applications, elasticity is not important and plastic formulations without elasticity are commonly applied. A newly proposed way of contracting tensors [69] will be applied in this paper, where a rigid plastic formulation will follow naturally as a special case.

Even with the computer capacity available today, CPFEM simulations are challenging. The coupling between the elements leads to a large system of non-linear coupled equations to be numerically integrated one time-step at a time. This can be obtained, using either explicit or implicit finite element methods. Due to the stiff nature of the involved system of equations, explicit time stepping is restricted to very small time-steps, even with carefully use of mass scaling. In explicit CPFEM solvers, a major part of the computer time required for the calculations is related to calculating locally for each integration point the stress tensor and the lattice rotation for a given time step, i.e. as prescribed in a user-defined subroutine. However, with implicit time stepping, most of the computer time is spent solving the global finite element equations. To compete with explicit numerical integration schemes, the implicit schemes must be sufficiently stable to allow order of magnitude larger time steps. A stable return mapping algorithm is the key.

At room temperature, a realistic exponent for this yield surface is orders of magnitude larger than for high-exponent yield surfaces applied in continuum plasticity. Numerical convergence of the return mapping algorithm is more and more difficult with increasing yield surface exponents and has until now been a major numerical challenge. However, recent progress has been reported within continuum plasticity, reporting stable return mapping algorithms, using a line-search approach [69–73].

In the current work, these algorithms are further developed and applied for the Gambin/Arminjon regularized crystal yield surface, enabling for the first time an implicit return mapping algorithm, allowing stable, effective calculations of the rate-independent limit with yield surface exponents as large as one million and strain steps as large as unity. Physical mechanisms, modeled by various types of slip systems, such as twinning, phase transformations, latent and reverse hardening etc., have not been included here. At the current stage, the paper presents a proof-of-concept of a rate-independent framework that enables this, without compromising numerical robustness.

The UMAT developed in this work can be freely downloaded from the following link:
https://gitlab.com/ntnu-physmet/crystal-plasticity

2. Regularized single crystal plasticity model

Two coordinate systems will be considered for expressing vectors and tensors of the crystal plasticity model. The global (sample) system has basis vectors $\boldsymbol{e}_i$, while the co-rotated (crystal) coordinate system has basis vectors $\hat{\boldsymbol{e}}_i$, $i = 1, 2, 3$, that coincides with the crystal lattice after deformation. The orthogonal transformation tensor from the global to the crystal coordinate system is denoted $\boldsymbol{Q} = \boldsymbol{R}^{\mathrm{T}}$. The transformation rules read

$$\hat{\boldsymbol{v}} = \boldsymbol{Q} \cdot \boldsymbol{v}, \hat{\boldsymbol{T}} = \boldsymbol{Q} \cdot \boldsymbol{T} \cdot \boldsymbol{Q}^{\mathrm{T}} \tag{4}$$

where $\boldsymbol{v}$ is a vector and $\boldsymbol{T}$ is a second-order tensor. The initial orientation of the crystal coordinate system is given by the initial transformation matrix $\boldsymbol{Q}_0 = \boldsymbol{R}_0^{\mathrm{T}}$, which can be calculated for a given set of Euler angles.

The imposed velocity gradient $\boldsymbol{L}$ is given in the global system, while the constitutive equations are formulated in the co-rotated crystal system. The hypoelastic approach is employed, with additive decomposition of the rate-of-deformation tensor. The hypoelastic–plastic models are typically used when elastic strains are small compared to the plastic strains. Except for some cases of complex elastic-dominated closed-loop cyclic loading, the non-conserved energy is negligible and the hypoelastic description is adequate.

In the co-rotated system $\hat{\boldsymbol{D}} = \operatorname{sym}\left(\hat{\boldsymbol{L}}\right)$ is split into its elastic and plastic parts

$$\hat{\boldsymbol{D}} = \hat{\boldsymbol{D}}^{\mathrm{e}} + \hat{\boldsymbol{D}}^{\mathrm{p}} \tag{5}$$

The rate of the co-rotated Cauchy stress is given by Hooke's law

$$\dot{\hat{\boldsymbol{\sigma}}} = \hat{\mathbb{C}} : \hat{\boldsymbol{D}}^{\mathrm{e}} = \hat{\mathbb{C}} : \left(\hat{\boldsymbol{D}} - \hat{\boldsymbol{D}}^{\mathrm{p}}\right) \tag{6}$$

where $\hat{\mathbb{C}}$ is the fourth-order elastic stiffness tensor, given in the co-rotated system. The unity slip direction vector $\hat{\boldsymbol{b}}_\alpha$ and the unity slip plane normal vector $\hat{\boldsymbol{n}}_\alpha$ for each considered slip system α define the Schmid tensor $\hat{\boldsymbol{M}}_\alpha$ in the co-rotated system.

$$\hat{\boldsymbol{M}}_\alpha = \hat{\boldsymbol{b}}_\alpha \otimes \hat{\boldsymbol{n}}_\alpha \tag{7}$$

The plastic rate-of-deformation tensor, $\hat{\boldsymbol{D}}^{\mathrm{p}}$, is related to the symmetric part of the Schmid matrix, $\hat{\boldsymbol{P}}_\alpha = \operatorname{sym}\left(\hat{\boldsymbol{M}}_\alpha\right)$ as

$$\hat{\boldsymbol{D}}^{\mathrm{p}} = \sum_{\alpha=1}^{N} \dot{\gamma}_\alpha \hat{\boldsymbol{P}}_\alpha \tag{8}$$

where $\dot{\gamma}_\alpha$ is the slip rate on slip system α.

Following Holmedal [38], the strain-rate independent regularized crystal yield surface is employed here,

$$f\left(\hat{\boldsymbol{\sigma}}\right) = \varphi\left(\hat{\boldsymbol{\sigma}}\right) - 1 = \left(\sum_{\alpha=1}^{N} \xi_\alpha \left|\frac{\hat{\boldsymbol{\sigma}} : \hat{\boldsymbol{P}}_\alpha}{\tau_{c_\alpha}}\right|^n\right)^{\frac{1}{n}} - 1 = 0 \tag{9}$$

where $\varphi\left(\hat{\boldsymbol{\sigma}}\right)$ is the yield function. The plastic rate-of-deformation tensor obeys the normality rule

$$\hat{\boldsymbol{D}}^{\mathrm{p}} = \dot{\lambda}\varphi_{\hat{\sigma}} = \frac{\dot{\lambda}}{(f+1)^{n-1}} \sum_{\alpha=1}^{N} \frac{\xi_\alpha}{\tau_{c_\alpha}} \left|\frac{\hat{\boldsymbol{\sigma}} : \hat{\boldsymbol{P}}_\alpha}{\tau_{c_\alpha}}\right|^{n-1} \hat{\boldsymbol{P}}_\alpha \tag{10}$$

where $\varphi_{\boldsymbol{\sigma}} = \partial\varphi/\partial\boldsymbol{\sigma}$ is the gradient of the yield function. When the exponent n is large, the parameters ξ_α may be set to unity. It follows from Eqs. (8) and (10), that on the yield surface, i.e. for $f = 0$,

$$\dot{\gamma}_\alpha = \dot{\lambda}\frac{\xi_\alpha}{\tau_{c_\alpha}}\left|\frac{\hat{\boldsymbol{\sigma}} : \hat{\boldsymbol{P}}_\alpha}{\tau_{c_\alpha}}\right|^{n-1} \operatorname{sign}\left(\hat{\boldsymbol{\sigma}} : \hat{\boldsymbol{P}}_\alpha\right) \tag{11}$$

Due to that f is a homogeneous function of the first order, it follows that

$$\dot{\lambda} = \hat{\boldsymbol{\sigma}} : \hat{\boldsymbol{D}}^{\mathrm{p}} = \dot{W} \tag{12}$$

where $\dot{W}$ is the plastic work rate.

To determine the single crystal rotation, the total spin tensor $\hat{\boldsymbol{W}} = \operatorname{skw}\left(\hat{\boldsymbol{L}}\right)$ is additively decomposed into its elastic, plastic, and lattice-rotation parts.

$$\hat{\boldsymbol{W}} = \hat{\boldsymbol{W}}^{\mathrm{p}} + \hat{\boldsymbol{W}}^{\mathrm{e}} + \hat{\boldsymbol{W}}^{\mathrm{c}} \tag{13}$$

Here, $\hat{\boldsymbol{W}}^{\mathrm{p}}$ is the plastic spin tensor. The elastic deformations that contribute to the elastic spin tensor $\hat{\boldsymbol{W}}^{\mathrm{e}}$, are very small and neglected here. The constitutive lattice spin, $\hat{\boldsymbol{W}}^{\mathrm{c}}$, generates the lattice rotation.

The origin of the plastic spin is the contribution to the spin from the shape-change caused by the slip activity, i.e.

$$\hat{\boldsymbol{W}}^{\mathrm{p}} = \sum_{\alpha=1}^{N} \dot{\gamma}_\alpha \hat{\boldsymbol{\Omega}}_\alpha \tag{14}$$

The skew-symmetric part of the Schmid tensor $\hat{\boldsymbol{\Omega}}_\alpha = \operatorname{skw}(\hat{\boldsymbol{M}}_\alpha)$ and $\dot{\gamma}_\alpha$ are the slip rates. Hence, the constitutive spin can be estimated as $\hat{\boldsymbol{W}}^{\mathrm{c}} \approx \hat{\boldsymbol{W}} - \hat{\boldsymbol{W}}^{\mathrm{p}}$. The constitutive spin tensor dictates the crystal rotation, according to

$$\dot{\boldsymbol{R}} = \boldsymbol{W}^{\mathrm{c}} \cdot \boldsymbol{R} = \boldsymbol{R} \cdot \hat{\boldsymbol{W}}^{\mathrm{c}} \cdot \boldsymbol{R}^{\mathrm{T}} \cdot \boldsymbol{R} = \boldsymbol{R} \cdot \hat{\boldsymbol{W}}^{\mathrm{c}} \tag{15}$$

The work hardening of the critical resolved shear stresses is assumed to be functions, $\tau_{c_\alpha}(\Gamma)$, that depend on the accumulated slip Γ, which is defined by the differential equation

$$\dot{\Gamma} = \sum_{\alpha=1}^{N} |\dot{\gamma}_\alpha| = |\dot{\lambda}| \sum_{\alpha=1}^{N} \frac{\xi_\alpha}{\tau_{c_\alpha}} \left|\frac{\hat{\boldsymbol{\sigma}} : \hat{\boldsymbol{P}}_\alpha}{\tau_{c_\alpha}}\right|^{n-1}, \Gamma(0) = 0 \tag{16}$$

In this paper, a simple model for the work hardening [74] is applied for demonstration

$$\frac{d\tau_{c_\alpha}}{dt} = h_\alpha(\Gamma)\,\dot{\Gamma} = \frac{\Delta\tau_\alpha^{\mathrm{sat}}}{\Delta\gamma_\alpha^{\mathrm{sat}}} \exp\left(-\frac{\Gamma}{\Delta\gamma_\alpha^{\mathrm{sat}}}\right)\dot{\Gamma},\ \tau_{c_\alpha}(0) = \tau_0 \tag{17}$$

where $h_\alpha(\Gamma)$ is the hardening moduli for each slip system. In this case, it can be integrated as the Voce equation

$$\tau_{c_\alpha} = \tau_0 + \Delta\tau_\alpha^{\mathrm{sat}}\left(1 - \exp\left(-\frac{\Gamma}{\Delta\gamma_\alpha^{\mathrm{sat}}}\right)\right) \tag{18}$$

Note, that replacing || by ⟨⟩, where ⟨⟩ denotes the Macaulay brackets, allows for distinguishing forward and backward slip activities as explained by Holmedal [38]. This is necessary to model Bauschinger effect on the slip system level [75–77].

3. Rotation update

The update of the rotation tensor $\boldsymbol{R}$ is given by the differential equation (15). An analytical solution exists for the case of a constant $\boldsymbol{W}$ and can be written using the Euler–Rodriguez formula (for details see Brannon [78]), as

$$\boldsymbol{R}(t) = \left(\boldsymbol{I} + \frac{\sin\left(\sqrt{\boldsymbol{W} : \boldsymbol{W}}t\right)}{\sqrt{\boldsymbol{W} : \boldsymbol{W}}}\boldsymbol{W} + \frac{\left(1 - \cos\left(\sqrt{\boldsymbol{W} : \boldsymbol{W}}t\right)\right)}{\boldsymbol{W} : \boldsymbol{W}}\boldsymbol{W} \cdot \boldsymbol{W}\right) \cdot \boldsymbol{R}_0 \tag{19}$$

where $\boldsymbol{R}_0 = \boldsymbol{R}(0)$. In general, the spin $\boldsymbol{W}$ changes with time. Eq. (19) can then be applied as an approximation during each time increment Δt, as

$$\boldsymbol{R}_{n+1} = \Delta \boldsymbol{R} \cdot \boldsymbol{R}_n = \left(\boldsymbol{I} + \frac{\sin\left(\sqrt{\boldsymbol{W}_n : \boldsymbol{W}_n t}\right)}{\sqrt{\boldsymbol{W}_n : \boldsymbol{W}_n}} \boldsymbol{W}_n + \frac{\left(1 - \cos\left(\sqrt{\boldsymbol{W}_n : \boldsymbol{W}_n t}\right)\right)}{\boldsymbol{W}_n : \boldsymbol{W}_n} \boldsymbol{W}_n \cdot \boldsymbol{W}_n \right) \cdot \boldsymbol{R}_n \tag{20}$$

where $\boldsymbol{W}_n$ is kept constant during the time increment.

Another alternative, is the symmetric numerical second-order update scheme by Hughes and Winget (1980), which is often employed (e.g. for calculating rotation increment matrix in UMAT in Abaqus/Standard), It assumes that the spin $\boldsymbol{W}$ is known at time $t_n + \Delta t/2$, and the Cayley–Hamilton theorem [79] can be applied to avoid inverting matrices [78].

$$\begin{aligned} \boldsymbol{R}_{n+1} &= \left(\boldsymbol{I} - \frac{\Delta t}{2} \boldsymbol{W}_{n+\frac{1}{2}} \right)^{-1} \cdot \left(\boldsymbol{I} + \frac{\Delta t}{2} \boldsymbol{W}_{n+\frac{1}{2}} \right) \cdot \boldsymbol{R}_n = \\ &= \left(\boldsymbol{I} + \frac{\Delta t}{1 + \frac{\Delta t^2}{4} \boldsymbol{W}_{n+\frac{1}{2}} : \boldsymbol{W}_{n+\frac{1}{2}}} \left(\boldsymbol{W}_{n+\frac{1}{2}} + \frac{\Delta t}{2} \boldsymbol{W}_{n+\frac{1}{2}} \cdot \boldsymbol{W}_{n+\frac{1}{2}} \right) \right) \cdot \boldsymbol{R}_n \end{aligned} \tag{21}$$

When using the corotated constitutive spin $\hat{\boldsymbol{W}}^c$, the integration of Eq. (15) reads

$$\boldsymbol{R}_{n+1} = \boldsymbol{R}_n \cdot \left(\boldsymbol{I} + \frac{\Delta t}{1 + \frac{\Delta t^2}{4} \left(\hat{\boldsymbol{W}}^c\right)_{n+\frac{1}{2}} : \left(\hat{\boldsymbol{W}}^c\right)_{n+\frac{1}{2}}} \left(\left(\hat{\boldsymbol{W}}^c\right)_{n+\frac{1}{2}} + \frac{\Delta t}{2} \left(\hat{\boldsymbol{W}}^c\right)_{n+\frac{1}{2}} \cdot \left(\hat{\boldsymbol{W}}^c\right)_{n+\frac{1}{2}} \right) \right) \tag{22}$$

4. Vector/matrix notation

Symmetric second-order tensors and fourth-order tensors with minor symmetry can be mapped into a vector and matrix representation, respectively. The most widely used vector/matrix representations are the Voigt and Mandel notations. The main purpose of a vector/matrix representation is to exploit the tensor symmetries, allowing symmetric stress and strain tensors to be stored as vectors, and fourth-order elastic modulus or plastic anisotropy tensors to be stored as matrices. In implementations of computational mechanics, this type of vector–matrix representation significantly reduces the number of operations and the computation cost. In rigid plastic crystal plasticity, other notations have been used (see Mánik [69] for a recent overview).

In this paper, the natural vector/matrix notation, originally suggested by Kocks, Tomé and Wenk [80] for use in crystal elasticity, is applied. This notation was recently adapted for use in continuum plasticity in a return mapping algorithm by Mánik [69]. Due to its explicit representation of the deviator, this notation advantageously separates the deviatoric plasticity from the elasticity, i.e. the plastic part is equivalent to the notation proposed by Lequeu, Gilormini, Montheillet, Bacroix and Jonas [81]. This notation enables a more concise algorithm formulation and, due to separation of pressure dependency, it reduces dimension of equation system and makes numerical computation more effective. Like the Mandel- but unlike the Voigt notation, it represents both stress and strain tensors equally, see Eq. (23). The brief description and the essentials of the natural notation is given in Appendix A. For an exhaustive description, see Mánik [69].

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{12} & A_{22} & A_{23} \\ A_{13} & A_{23} & A_{33} \end{pmatrix} \longrightarrow \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{3}(A_{11} + A_{22} + A_{33}) \\ 1/\sqrt{6}(2A_{33} - A_{11} - A_{22}) \\ 1/\sqrt{2}(A_{22} - A_{11}) \\ \sqrt{2}A_{23} \\ \sqrt{2}A_{13} \\ \sqrt{2}A_{12} \end{pmatrix} \tag{23}$$

5. Return mapping algorithm

At time $t^{(n)}$, we consider the Cauchy stress $\hat{\boldsymbol{\sigma}}^{(n)}$ and the internal variables $\hat{\boldsymbol{q}}^{(n)}$ expressed in the corotational crystal coordinate system. In the backward Euler integration scheme, the total rate-of-deformation $\hat{\boldsymbol{D}}^{(n+1)}$ is required at $t^{(n+1)}$ in the corotational crystal coordinate system. However, the orientation of the crystal coordinate system at

$t^{(n+1)}$ is not known, hence $\hat{\boldsymbol{D}}^{(n+1)}$ needs to be calculated. What is known in the finite element code is $\boldsymbol{D}^{(n+1/2)}$, which is assumed to be constant in the reference system throughout the time step. To limit the complexity of the algorithmic modulus, $\hat{\boldsymbol{D}}^{(n+1)}$ is extrapolated within the accuracy of the numerical scheme (see Appendix B).

The basic problem to be solved by a return mapping algorithm, is to find the Cauchy stress $\hat{\boldsymbol{\sigma}}^{(n+1)}$ and the internal variables $\hat{\boldsymbol{q}}^{(n+1)}$ at time $t^{(n+1)} = t^{(n)} + \Delta t$, which satisfy the Kuhn–Tucker complementarity conditions

$$f\left(\hat{\boldsymbol{\sigma}}^{(n+1)}, \hat{\boldsymbol{q}}^{(n+1)}\right) \leq 0, \quad \dot{\lambda} \geq 0, \quad \dot{\lambda} f\left(\hat{\boldsymbol{\sigma}}^{(n+1)}, \hat{\boldsymbol{q}}^{(n+1)}\right) = 0 \tag{24}$$

The fully implicit backward Euler return mapping algorithm is employed in this paper. In the literature, return-mapping algorithms are generally formulated in their tensorial form, while numerical implementations employ either Voigt or Mandel vector/matrix notation. In the following, the return-mapping algorithm will be expressed directly in the natural notation [69]. For the sake of clarity, the hat ($\hat{}$) designating the corotational aspect, will be omitted in the following, for all the vectorial and tensorial quantities.

Given a total rate-of-deformation vector $\vec{\mathbf{d}}^{(n+1)}$, time increment Δt and Cauchy stress $\vec{\boldsymbol{\sigma}}^{(n)}$, the trial stress is obtained by applying an elastic predictor.

$$\vec{\boldsymbol{\sigma}}^{\text{tr}\,(n+1)} = \vec{\boldsymbol{\sigma}}^{(n)} + \mathbf{C}\vec{\mathbf{d}}^{(n+1)}\Delta t \tag{25}$$

Here, in the natural notation, $\mathbf{C}$ is a 6 × 6 diagonal matrix representing the elastic moduli with cubic symmetry.

If $\varphi\left(\vec{\boldsymbol{\sigma}}^{\text{tr}\,(n+1)}, \Gamma^{(n)}\right) - 1 \leq 0$, the strain increment is elastic and $\vec{\boldsymbol{\sigma}}^{(n+1)} = \vec{\boldsymbol{\sigma}}^{\text{tr}\,(n+1)}$ and $\Gamma^{(n+1)} = \Gamma^{(n)}$. Otherwise, when $\varphi\left(\vec{\boldsymbol{\sigma}}^{\text{tr}\,(n+1)}, \Gamma^{(n)}\right) - 1 > 0$, plastic deformation occurs and the return mapping algorithm will be employed to integrate the model and to obtain $\vec{\boldsymbol{\sigma}}^{(n+1)}$, $\dot{\lambda}^{(n+1)}$ and $\Gamma^{(n+1)}$ at $t^{(n+1)}$.

The plastic corrector part of the implicit backward-Euler return mapping algorithm finds, in an iterative manner, the Cauchy stress $\vec{\boldsymbol{\sigma}}^{(n+1)}$, the plastic multiplier $\dot{\lambda}^{(n+1)}$, and the slip $\Gamma^{(n+1)}$ at time $t^{(n+1)} = t^{(n)} + \Delta t$, by solving the following system of nonlinear equations:

$$\vec{\mathbf{r}} = -\vec{\mathbf{d}}^{\text{p}} + \dot{\lambda}\varphi_{\vec{\boldsymbol{\sigma}}} = -\mathbf{C}^{-1}\left(\vec{\boldsymbol{\sigma}}^{\text{tr}} - \vec{\boldsymbol{\sigma}}\right)\frac{1}{\Delta t} + \dot{\lambda}\varphi_{\vec{\boldsymbol{\sigma}}} = \vec{0} \tag{26}$$

$$f = \varphi\left(\vec{\boldsymbol{\sigma}}, \Gamma\right) - 1 = 0 \tag{27}$$

$$q = -\frac{\left(\Gamma - \Gamma^{(n)}\right)}{\Delta t} + \dot{\Gamma}\left(\vec{\boldsymbol{\sigma}}, \Gamma, \dot{\lambda}\right) = 0 \tag{28}$$

Instead of using the trial stress $\vec{\boldsymbol{\sigma}}^{\text{tr}}$ as the initial guess, the radially returned stress onto the yield surface, $\vec{\boldsymbol{\sigma}}^{(0)}$, is calculated.

$$\vec{\boldsymbol{\sigma}}^{(0)} = k\frac{\vec{\boldsymbol{\sigma}}^{\text{tr}}}{\varphi(\vec{\boldsymbol{\sigma}}^{\text{tr}})} \tag{29}$$

For large trial stresses, this first guess is the key to numerically stable and robust return-mapping algorithm. The parameter k controls the distance of the initial stress guess from the yield surface, for $k = 1$, $\vec{\boldsymbol{\sigma}}^{(0)}$ lies on the yield surface, as $\varphi\left(\vec{\boldsymbol{\sigma}}^{(0)}\right) = 1$. For a given $\vec{\boldsymbol{\sigma}}^{(0)}$, the initial guess for the plastic multiplier, $\dot{\lambda}^{(0)}$ is given by Eq. (12)

$$\dot{\lambda}^{(0)} = \vec{\mathbf{d}}^{\text{p}^{\text{T}}}\vec{\boldsymbol{\sigma}}^{(0)} = \frac{k}{\varphi(\vec{\boldsymbol{\sigma}}^{\text{tr}})}\left(1 - \frac{k}{\varphi\left(\vec{\boldsymbol{\sigma}}^{\text{tr}}\right)}\right)\vec{\boldsymbol{\sigma}}^{\text{tr}^{\text{T}}}\mathbf{C}^{-1}\vec{\boldsymbol{\sigma}}^{\text{tr}} \tag{30}$$

The return mapping is solved using a Newton–Raphson algorithm with a line search. The solution is sought in an iterative manner as

$$\begin{aligned}\vec{\boldsymbol{\sigma}}^{(k+1)} &= \vec{\boldsymbol{\sigma}}^{(k)} + \alpha^{(k)}\,\Delta\vec{\boldsymbol{\sigma}}\\ \dot{\lambda}^{(k+1)} &= \dot{\lambda}^{(k)} + \alpha^{(k)}\Delta\dot{\lambda}\\ \Gamma^{(k+1)} &= \Gamma^{(k)} + \alpha^{(k)}\Delta\Gamma\end{aligned} \tag{31}$$

where (k) is the iteration, $\Delta\vec{\boldsymbol{\sigma}}$, $\Delta\dot{\lambda}$ and $\Delta\Gamma$ are the increment of the Cauchy stress, the plastic multiplier and the accumulated slip, respectively, and $\alpha^{(k)}$ is a constant to be determined by the line search algorithm. By linearizing

Eqs. (26), (27) and (28), the increment $\Delta\dot{\lambda}$ is calculated as

$$\Delta\dot{\lambda} = \frac{f^{(k)} - {\varphi_{\vec{\boldsymbol{\sigma}}}^{(k)}}^{\mathrm{T}} \mathbf{L}\vec{\mathbf{r}}^{(k)} - Y\left(\dot{\lambda}^{(k)} {\varphi_{\vec{\boldsymbol{\sigma}}}^{(k)}}^{\mathrm{T}} \mathbf{L}\varphi_{\vec{\boldsymbol{\sigma}}\Gamma}^{(k)} - \varphi_{\Gamma}^{(k)}\right)\left({\dot{\Gamma}_{\vec{\boldsymbol{\sigma}}}^{(k)}}^{\mathrm{T}} \mathbf{L}\vec{\mathbf{r}}^{(k)} - q^{(k)}\right)}{{\varphi_{\vec{\boldsymbol{\sigma}}}^{(k)}}^{\mathrm{T}} \mathbf{L}\varphi_{\vec{\boldsymbol{\sigma}}}^{(k)} + Y\left(\dot{\lambda}^{(k)} {\varphi_{\vec{\boldsymbol{\sigma}}}^{(k)}}^{\mathrm{T}} \mathbf{L}\varphi_{\vec{\boldsymbol{\sigma}}\Gamma}^{(k)} - \varphi_{\Gamma}^{(k)}\right)\left({\dot{\Gamma}_{\vec{\boldsymbol{\sigma}}}^{(k)}}^{\mathrm{T}} \mathbf{L}\varphi_{\vec{\boldsymbol{\sigma}}}^{(k)} - \dot{\Gamma}_{\dot{\lambda}}^{(k)}\right)} \tag{32}$$

The latter is then used for calculating the increment of the accumulated slip

$$\Delta\Gamma = Y\left({\dot{\Gamma}_{\vec{\boldsymbol{\sigma}}}^{(k)}}^{\mathrm{T}} \mathbf{L}\vec{\mathbf{r}}^{(k)} - q^{(k)}\right) + Y\left({\dot{\Gamma}_{\vec{\boldsymbol{\sigma}}}^{(k)}}^{\mathrm{T}} \mathbf{L}\varphi_{\vec{\boldsymbol{\sigma}}}^{(k)} - \dot{\Gamma}_{\dot{\lambda}}^{(k)}\right)\Delta\dot{\lambda} \tag{33}$$

and both $\Delta\dot{\lambda}$ and $\Delta\Gamma$ are finally used for obtaining the increment of the Cauchy stress

$$\Delta\vec{\boldsymbol{\sigma}} = -\mathbf{L}\left(\vec{\mathbf{r}}^{(k)} + \varphi_{\vec{\boldsymbol{\sigma}}}^{(k)}\Delta\dot{\lambda} + \dot{\lambda}^{(k)}\varphi_{\vec{\boldsymbol{\sigma}}\Gamma}^{(k)}\,\Delta\Gamma\right) \tag{34}$$

In Eqs. (32), (33) and (34), the matrix $\mathbf{L}$ and Y are given as

$$\begin{aligned} \mathbf{L} &= \left(\frac{\mathbf{C}^{-1}}{\Delta t} + \dot{\lambda}^{(k)}\varphi_{\vec{\boldsymbol{\sigma}}\vec{\boldsymbol{\sigma}}}^{(k)}\right)^{-1} \\ Y &= \left(\dot{\Gamma}_{\Gamma}^{(k)} - \frac{1}{\Delta t} - \dot{\lambda}^{(k)} {\dot{\Gamma}_{\vec{\boldsymbol{\sigma}}}^{(k)}}^{\mathrm{T}} \mathbf{L}\varphi_{\vec{\boldsymbol{\sigma}}\Gamma}^{(k)}\right)^{-1} \end{aligned} \tag{35}$$

For the calculations above, the following partial derivatives need to be calculated (note that the resolved shear stress $\tau_\alpha = \vec{\boldsymbol{\sigma}}^{\mathrm{T}}\vec{\mathbf{p}}_\alpha$):

$$\begin{aligned} \frac{\partial\varphi}{\partial\vec{\boldsymbol{\sigma}}} &= \varphi_{\vec{\boldsymbol{\sigma}}} = \sum_{\alpha=1}^{N}\frac{\xi_\alpha}{\tau_{c_\alpha}}\left(\frac{|\tau_\alpha|}{\varphi\tau_{c_\alpha}}\right)^{n-1}\operatorname{sign}(\tau_\alpha)\,\vec{\mathbf{p}}_\alpha \\ \frac{\partial^2\varphi}{\partial\vec{\boldsymbol{\sigma}}^2} &= \varphi_{\vec{\boldsymbol{\sigma}}\vec{\boldsymbol{\sigma}}} = \frac{(n-1)}{\varphi}\left(\sum_{\alpha=1}^{N}\frac{\xi_\alpha}{\tau_{c_\alpha}^2}\left(\frac{|\tau_\alpha|}{\varphi\tau_{c_\alpha}}\right)^{n-2}\vec{\mathbf{p}}_\alpha\vec{\mathbf{p}}_\alpha^{\mathrm{T}} - \varphi_{\vec{\boldsymbol{\sigma}}}\varphi_{\vec{\boldsymbol{\sigma}}}^{\mathrm{T}}\right) \\ \frac{\partial\varphi}{\partial\Gamma} &= \varphi_{\Gamma} = -\varphi\sum_{\alpha=1}^{N}\frac{\xi_\alpha}{\tau_{c_\alpha}}\left(\frac{|\tau_\alpha|}{\varphi\tau_{c_\alpha}}\right)^{n}h_\alpha \\ \frac{\partial^2\varphi}{\partial\vec{\boldsymbol{\sigma}}\,\partial\Gamma} &= \varphi_{\vec{\boldsymbol{\sigma}}\Gamma} = (1-n)\,\frac{\varphi_\Gamma}{\varphi}\varphi_{\vec{\boldsymbol{\sigma}}} - n\sum_{\alpha=1}^{N}\frac{h_\alpha\xi_\alpha}{\tau_{c_\alpha}^2}\left(\frac{|\tau_\alpha|}{\varphi\tau_{c_\alpha}}\right)^{n-1}\operatorname{sign}(\tau_\alpha)\,\vec{\mathbf{p}}_\alpha \\ \frac{\partial\dot{\Gamma}}{\partial\dot{\lambda}} &= \dot{\Gamma}_{\dot{\lambda}} = \operatorname{sign}\left(\dot{\lambda}\right)\sum_{\alpha=1}^{N}\frac{\xi_\alpha}{\tau_{c_\alpha}}\left(\frac{|\tau_\alpha|}{\varphi\tau_{c_\alpha}}\right)^{n-1} \\ \frac{\partial\dot{\Gamma}}{\partial\vec{\boldsymbol{\sigma}}} &= \dot{\Gamma}_{\vec{\boldsymbol{\sigma}}} = \frac{(n-1)}{\varphi}\left(\left|\dot{\lambda}\right|\sum_{\alpha=1}^{N}\frac{\xi_\alpha}{\tau_{c_\alpha}^2}\left(\frac{|\tau_\alpha|}{\varphi\tau_{c_\alpha}}\right)^{n-2}\operatorname{sign}(\tau_\alpha)\,\vec{\mathbf{p}}_\alpha - \dot{\Gamma}\varphi_{\vec{\boldsymbol{\sigma}}}\right) \\ \frac{\partial\dot{\Gamma}}{\partial\Gamma} &= \dot{\Gamma}_{\Gamma} = -n\left|\dot{\lambda}\right|\sum_{\alpha=1}^{N}\frac{\xi_\alpha h_\alpha}{\tau_{c_\alpha}^2}\left(\frac{|\tau_\alpha|}{\varphi\tau_{c_\alpha}}\right)^{n-1} + (1-n)\,\frac{\dot{\Gamma}\varphi_\Gamma}{\varphi} \end{aligned} \tag{36}$$

For convergence, measures of the three residuals $\vec{\mathbf{r}}^{(k)}$, $f^{(k)}$ and $q^{(k)}$ for iteration (k) are defined as

$$\psi_r^{(k)} = \frac{1}{2}{\vec{\mathbf{r}}^{(k)}}^{\mathrm{T}}\vec{\mathbf{r}}^{(k)}\Delta t^2, \quad \psi_f^{(k)} = \frac{1}{2}{f^{(k)}}^2, \quad \psi_q^{(k)} = \frac{1}{2}{q^{(k)}}^2\Delta t^2 \tag{37}$$

and

$$\psi^{(k)} = \psi_r^{(k)} + \psi_f^{(k)} + \psi_q^{(k)} \tag{38}$$

If for an iteration (k)

$$\psi_r^{(k)} < \varepsilon_r \text{ and } \psi_f^{(k)} < \varepsilon_f \text{ and } \psi_q^{(k)} < \varepsilon_q \tag{39}$$

then the convergence is achieved. Recommended error tolerances, used throughout this work, are $\varepsilon_r = 10^{-20}$, $\varepsilon_f = 10^{-8}$, $\varepsilon_q = 10^{-16}$.

5.1. Line search

In each Newton–Raphson iteration, the line search is used to determine the step size $\alpha^{(k)}$. For this, $\psi^{(k)}$ serves as the merit function [72]. For the search direction given by the increments $\Delta\vec{\boldsymbol{\sigma}}$, $\Delta\dot{\lambda}$ and $\Delta\Gamma$, the step length $\alpha^{(k)}$ needs to be found such that $0 < \alpha^{(k)} \leq 1$ and for which the merit function $\psi^{(k)}$ is minimized. In this paper, two methods for line search are employed and tested. A line search, calculating the minimum of the quadratic approximation to the merit function, is adopted here, similarly as previously applied in return mapping algorithms for continuum plasticity [69,71,72]. This is very efficient method for the exponents of the regularized yield function up to ∼100. For large exponents up to 1 000 000, the quadratic approximation of the merit function becomes too poor, leading to $\alpha^{(k)}$'s far from the optimal ones. More effective line search is developed in this paper, employing new minimization algorithm which solves more efficiently crystal plasticity models with large exponents up to 1 000 000.

5.1.1. Quadratic line search

For each iteration, the Newton step is performed as the first attempt, i.e. $\alpha^{(k)} = 1$ in Eq. (31). This step will be accepted only if $\psi^{(k+1)}$ is lower than some fraction of the merit function $\psi^{(k)}$, achieved in the previous Newton–Raphson iteration.

$$\psi^{(k+1)} < (1 - 2\beta)\,\psi^{(k)} \tag{40}$$

The parameter $\beta = 10^{-4}$, as recommended by Perez-Foguet and Armero [72]. If the Newton step does not meet the condition in Eq. (40), it is instead considered as a first iteration, $j = 0$, for which $\psi^{(k+1)}_{(j)} = \psi^{(k+1)}$ and $\alpha^{(k)}_{(j)} = 1$. Next, an improved guess $\alpha^{(k)}_{(j+1)}$ is calculated based on a quadratic series expansion of the function $\psi^{(k+1)}\left(\alpha^{(k)}\right)$, utilizing that $\psi^{(k+1)}(0) = \psi^{(k)}$ and the derivative $\psi^{(k+1)'}(0) = -2\psi^{(k+1)}(0) = -2\psi^{(k)}$.

$$\alpha^{(k)}_{(j+1)} = \max\left(\eta\alpha^{(k)}_{(j)}, \frac{\left(\alpha^{(k)}_{(j)}\right)^2 \psi^{(k)}}{\psi^{(k+1)}_{(j)} - \left(1 - 2\alpha^{(k)}_{(j)}\right)\psi^{(k)}}\right) \tag{41}$$

The index (j) indicates iterations within the line search. The parameter $\eta = 0.1$ limits the size of the new estimate $\alpha^{(k)}_{(j+1)}$. For each line search iteration (j), the merit function $\psi^{(k+1)}_{(j)}$ is calculated according to Eqs. (31) and (32) using the updates of $\vec{\boldsymbol{\sigma}}^{(k+1)}_{(j)}$, $\dot{\lambda}^{(k+1)}_{(j)}$ and $\Gamma^{(k+1)}_{(j)}$ calculated by using the step $\alpha^{(k)}_{(j)}$.

$$\begin{aligned}
\vec{\boldsymbol{\sigma}}^{(k+1)}_{(j)} &= \vec{\boldsymbol{\sigma}}^{(k)} + \alpha^{(k)}_{(j)}\,\Delta\vec{\boldsymbol{\sigma}} \\
\dot{\lambda}^{(k+1)}_{(j)} &= \dot{\lambda}^{(k)} + \alpha^{(k)}_{(j)}\Delta\dot{\lambda} \\
\Gamma^{(k+1)}_{(j)} &= \Gamma^{(k)} + \alpha^{(k)}_{(j)}\Delta\Gamma
\end{aligned} \tag{42}$$

Within the line search iterations, the residual $\psi^{(k+1)}_{(j)}$ has to satisfy the Goldstein's condition [72,82]

$$\psi^{(k+1)}_{(j)} < \left(1 - 2\beta\alpha^{(k)}_{(j)}\right)\psi^{(k)} \tag{43}$$

If Eq. (43) is not satisfied, the line search iterations continue by calculating new $\alpha^{(k)}_{(j+1)}$ using Eq. (41). On the other hand, if Eq. (43) is satisfied, then both the line search iteration (j) and the Newton–Raphson iteration (k) are terminated by setting $\alpha^{(k)} = \alpha^{(k)}_{(j)}$, updating $\vec{\boldsymbol{\sigma}}^{(k+1)}$, $\dot{\lambda}^{(k+1)}$ and $\Gamma^{(k+1)}$ using Eq. (31). If the convergence is not reached, i.e. Eq. (39) is not satisfied, then the algorithm continues with the $(k{+}1)$th Newton–Raphson iteration.

5.1.2. Line search by minimization

An alternative approach for finding the step length $\alpha^{(k)}$ for which $\psi^{(k)}\left(\alpha^{(k)}\right) < \psi^{(k)}(0)$, is to find the minimum within some tolerance ε. For this, the standard and general 1D minimization Brent's method [83] can be applied. The way the function $\psi^{(k)}\left(\alpha^{(k)}\right)$ is constructed by Eqs. (37) and (38) gives rise to some known properties e.g. the derivative at $\alpha^{(k)} = 0$, reads $\psi^{(k)'}(0) = -2\psi^{(k)}(0)$. A new minimization algorithm was tailor-made to utilize the known characteristics of $\psi^{(k)}\left(\alpha^{(k)}\right)$ making it faster than Brent's method. See Appendix C for the detailed description of this line search algorithm.

Table 1
Constitutive parameters used in the convergence study.

	C_{11} (GPa)	C_{12} (GPa)	C_{44} (GPa)	τ_c (MPa)
FCC case	106.75	60.41	28.34	10
BCC case	233.3	135.5	128.0	100

5.2. Consistent algorithmic modulus

The consistent algorithmic modulus is essential to calculate when the return mapping algorithm is employed as a part of an outer iteration. It must be provided as part of the user subroutine in the implicit FE solver. For the implicit backward Euler return mapping algorithm of the regularized single crystal plasticity model described above, it is calculated as

$$\mathbf{C}^{\mathrm{alg}} = \left(\mathbf{M} - \frac{\left(\mathbf{M}\varphi_{\vec{\boldsymbol{\sigma}}} - \dot{\lambda} Y \dot{\Gamma}_{\dot{\lambda}} \mathbf{L} \varphi_{\vec{\boldsymbol{\sigma}}\Gamma}\right)\left(\varphi_{\Gamma} Y \dot{\Gamma}_{\vec{\boldsymbol{\sigma}}}^{\mathrm{T}} \mathbf{L} - \varphi_{\vec{\boldsymbol{\sigma}}}^{\mathrm{T}} \mathbf{M}\right)}{-\varphi_{\vec{\boldsymbol{\sigma}}}^{\mathrm{T}}\left(\mathbf{M}\varphi_{\vec{\boldsymbol{\sigma}}} - \dot{\lambda} Y \dot{\Gamma}_{\dot{\lambda}} \mathbf{L} \varphi_{\vec{\boldsymbol{\sigma}}\Gamma}\right) + \varphi_{\Gamma} Y \dot{\Gamma}_{\vec{\boldsymbol{\sigma}}}^{\mathrm{T}} \mathbf{L} \varphi_{\vec{\boldsymbol{\sigma}}} - \varphi_{\Gamma} Y \dot{\Gamma}_{\dot{\lambda}}} \right) \frac{1}{\Delta t} \tag{44}$$

in which matrix $\mathbf{M}$ is calculated as

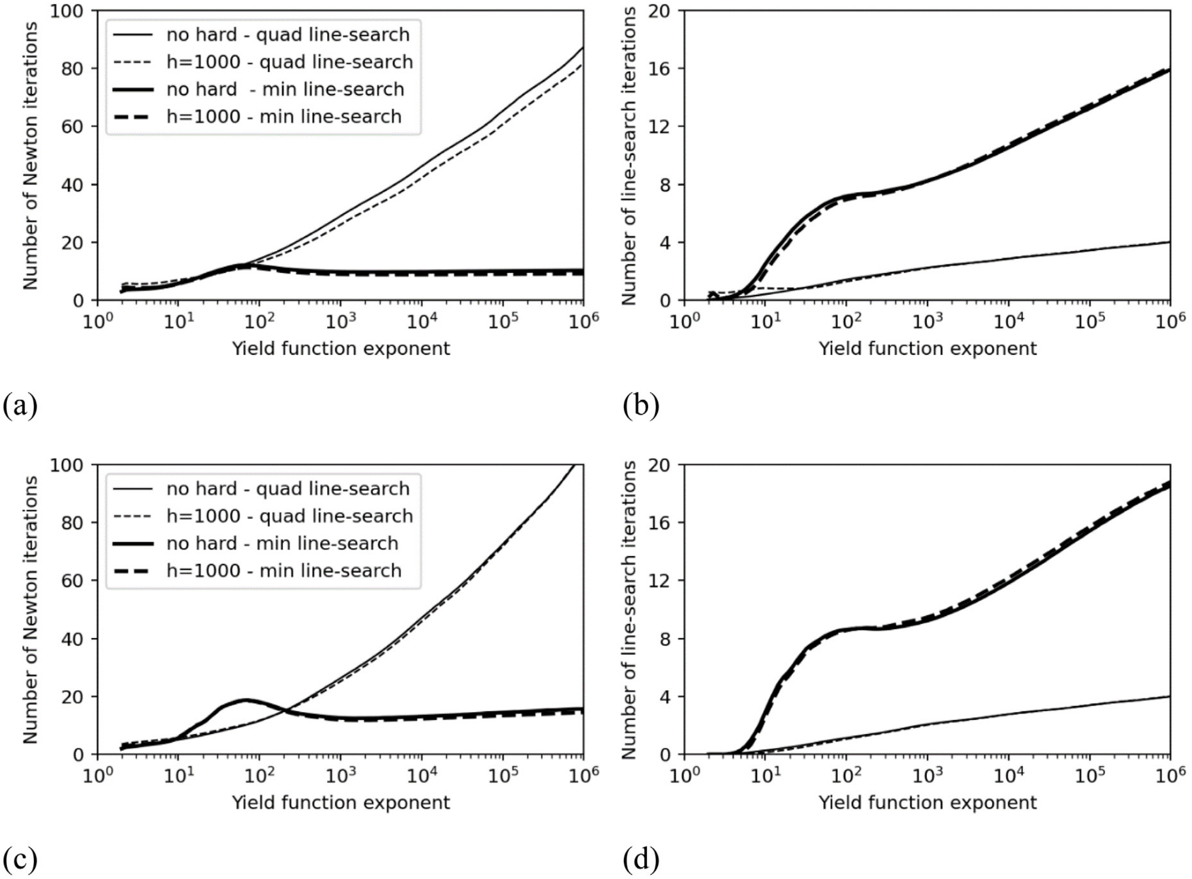
$$\mathbf{M} = \left(\dot{\lambda} Y \mathbf{L} \varphi_{\vec{\boldsymbol{\sigma}}\Gamma} \dot{\Gamma}_{\vec{\boldsymbol{\sigma}}}^{\mathrm{T}} + \mathbf{I}\right) \mathbf{L} \tag{45}$$

6. Convergency results

Convergency behavior of the implicit, backward-Euler return-mapping algorithm with line search is examined as a function of the yield-surface exponent n. Note that according to Eq. (25), an arbitrary strain increment occurring during a time increment Δt, is uniquely prescribed by $\vec{\boldsymbol{\sigma}}^{\mathrm{tr}} - \vec{\boldsymbol{\sigma}}^{(n)}$. Hence, to effectively cover the space of possible strain-increment directions, an evenly distributed set of 10 000 trial stresses $\left\{\vec{\boldsymbol{\sigma}}_i^{\mathrm{tr}}\right\}_{i=1,10000}$ was generated from a set of 5-dimensional vectors, approximately uniformly distributed on the 5-dimensional hypersphere, being generated by the algorithm described in Appendix D. The stress state before the strain increment to be iterated, $\vec{\boldsymbol{\sigma}}^{(n)}$ can, without loss of generality, be set equal to $\vec{0}$, i.e. strain path changes are also covered by this set. Each trial stress was chosen so that $f\left(\vec{\boldsymbol{\sigma}}_i^{\mathrm{tr}}\right) = s$, where s is a chosen constant. The magnitude of s represents the magnitude of the trial stress, which implicitly corresponds to the magnitude of the total strain increment $\Delta\vec{\boldsymbol{\varepsilon}} = \vec{\mathbf{d}}\Delta t$, for a given elastic modulus matrix $\mathbf{C}$ and a set of critical resolved shear stresses τ_{c_α}. In this manner, a large set of possible strain paths can be probed, independently of what the previous stress solution was, and arbitrary strain-path changes are included as well. To test different strain-increment magnitudes, four selected values of $s = 2\tau_C, 10\tau_C, 100\tau_C$ and $1000\tau_C$ were included in the set of trial stresses. The largest amongst these trial stresses corresponds to a strain increment of order unity and in practice provides an ultimate stability challenge for the algorithm. In total, the 40 000 tested strain increments provide a large set that effectively covers the space of realistic strain increments in an implicit FE simulation.

The efficiency and stability of the return-mapping algorithm will in principle depend on the chosen slip systems and their corresponding work hardening. A BCC structure with $\left\{1\bar{1}0\right\}\langle 111\rangle$, $\left\{\bar{1}2\bar{1}\right\}\langle 111\rangle$ and $\left\{1\bar{3}2\right\}\langle 111\rangle$ slip systems, and an FCC structure with $\left\{1\bar{1}1\right\}\langle 011\rangle$ slip systems were tested. Without loss of generality, the Euler angles used were $(\varphi_1, \Phi, \varphi_2) = (0, 0, 0)$. The elastic constants and the critical resolved shear stress applied for the FCC and BCC case are listed in Table 1. Both a case without work hardening and a case of strong work hardening (linear hardening with $h = 1000$ MPa) were tested. For each yield surface exponent, the number of Newton iterations and the number of line-search iterations were counted for the 40 000 probed strain increments.

It turned out that the algorithm gave similar iteration statistics for all cases tested. Examples of the average number of Newton iterations and the average number of line-search iterations required for each Newton iteration to reach convergency, are shown in Fig. 1 for two of the cases tested. The largest system, for the BCC case with 48 slip systems, requires only slightly more iterations to converge. The results with and without work hardening are very similar. Note that for yield-surface exponents up to about 100, the quadratic line search is always faster, with less or equal number of Newton iterations and significantly fewer line-search iterations each. The simpler quadratic line-search algorithm is competitive up to a yield-surface exponent of the order of 1000, which in practice is sufficient for a very good rate-independent approximation by the regularized yield surface.



(a)

(b)

(c)

(d)

Fig. 1. Average number of Newton iterations for FCC in (a) and BCC in (c). Average number of line-search iterations for each Newton iteration for FCC in (b) and BCC in (d). Cases with and without hardening are compared for iterations by quadratic or full minimization during the line search.

It is a remarkable result, that both line-search algorithms converge for all cases tested, up to an exponent of one million, keeping in mind that the largest strain steps included in the test set is of order unity. For these extremely high exponents, the full minimization requires significantly fewer iterations than the quadratic algorithm.

For low exponents, most cases converge within a few iterations. For larger exponents, some of the tested strain steps converge fast, while other require more iterations. In Fig. 2, examples of iteration statistics for the case of the quadratic line-search algorithm are shown, for cases where $n = 10, 100$ and 10^6. The FCC crystal without hardening and the BCC crystal with 48 slip systems and hardening show very similar distributions. The average number of Newton iterations, as well as the spread of the distribution, increase with increasing exponent. For the cases of $n = 10$ and 100, the peak is at zero line-search iteration, i.e. a full Newton step, while for $n = 10^6$ several repeated quadratic line-search iterations are required for most cases.

Fig. 3 shows iteration statistics for examples where the full minimization is applied during the line search. Again, cases of $n = 10$ and 100 are compared for an FCC crystal without hardening and a BCC crystal with 48 slip systems and hardening. Unlike the cases with quadratic line search, the mean value and spread of the number of Newton iterations saturate at large exponents and the curves are similar for $n = 100$ and 10^6. However, the number of line search iterations per Newton iteration slowly increases with increasing n, showing a broad peak at the largest number of iterations, as well as a narrow peak, for which convergency is accepted by one Newton step.

7. Application to CPFEM

The return-mapping algorithm was implemented into the user-material subroutine (UMAT) in the FE software Abaqus/Standard and tested for two cases covering simulation of single and polycrystal behavior of an FCC material

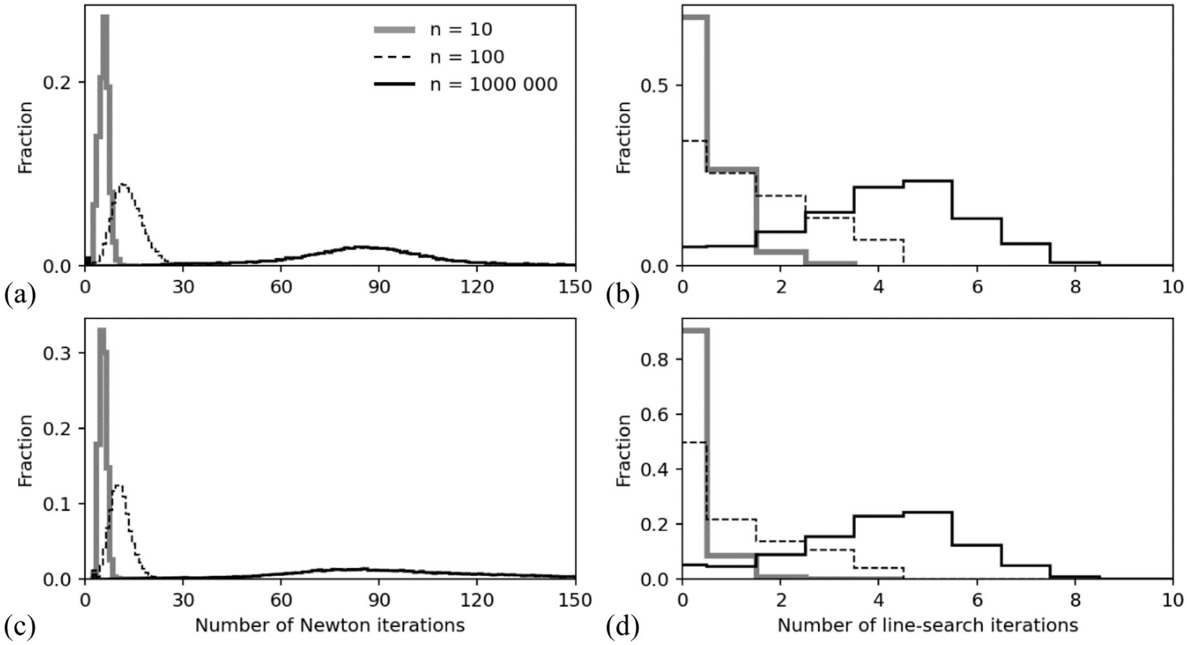


Fig. 2. Line search with quadratic approximation. Distributions of Newton and line-search iterations for the test case of an FCC crystal without hardening (a), (b) and a BCC crystal with hardening (c), (d).

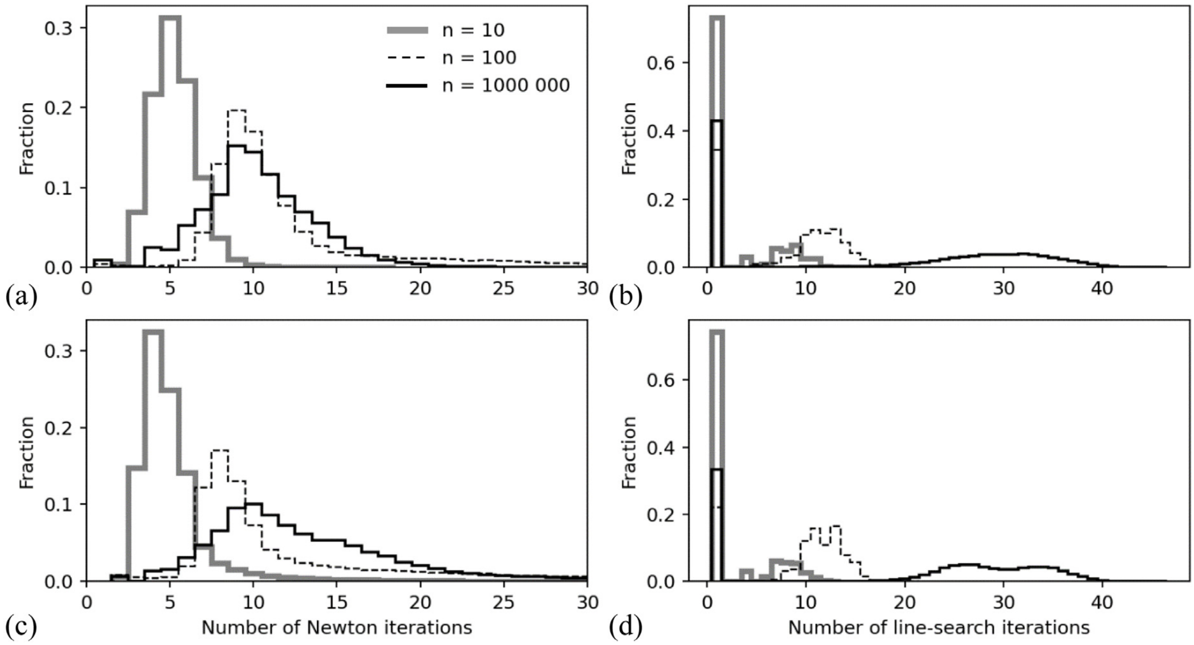


Fig. 3. Line search with new minimizing algorithm. Distributions of Newton and line-search iterations for the test case of an FCC crystal without hardening (a), (b) and a BCC crystal with hardening (c), (d).

with $\left\{1\bar{1}1\right\}\langle 011\rangle$ slip systems. Elastic constants for aluminum were used as given in Table 1. The initial critical resolved shear stress was 10 MPa. The hardening law by Eq. (17) was applied. The exponent $n = 100$ of the yield function was used. The tolerances ε_r, ε_f and ε_q in Eq. (39), for the return-mapping algorithm convergence, were

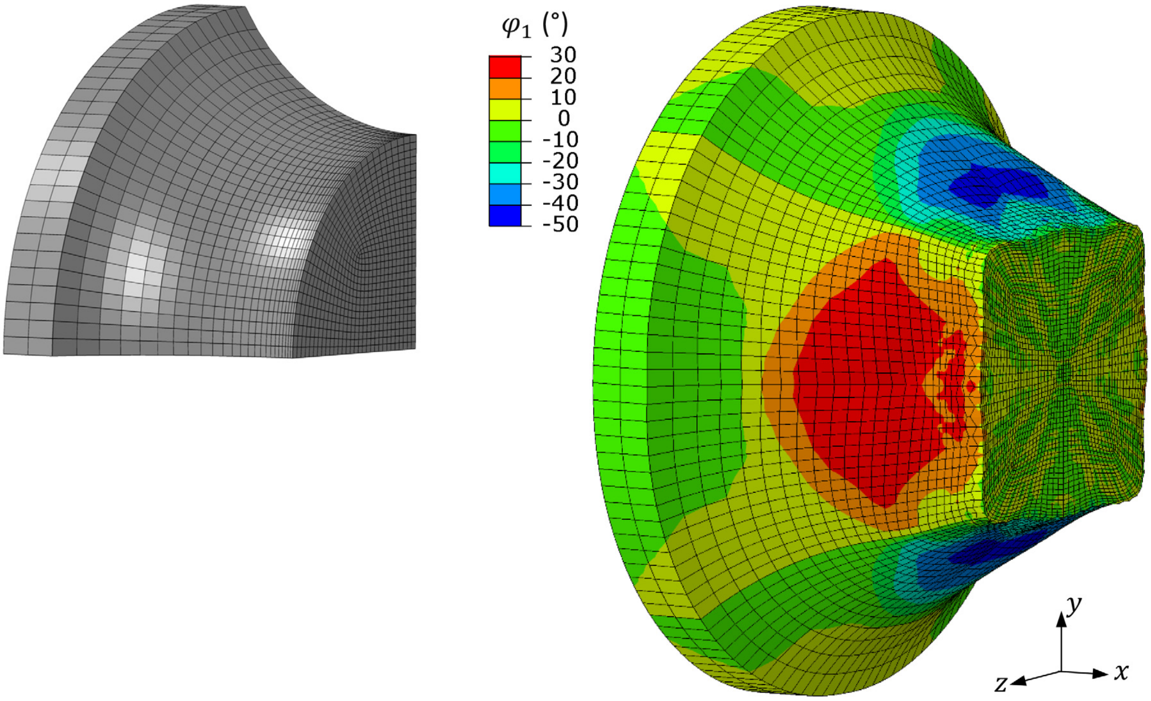


Fig. 4. Uniaxial tension of a notched single crystal in a Goss orientation, shown half of the sample.

10^{-20}, 10^{-8} and 10^{-16}, respectively. The CPFEM calculations were run on a Windows 10 system, using 6 of the 8 threads of the Intel Core i7-7700 processor with 3.6 GHz CPU. The user-subroutine was applied in Abaqus/Standard 2020, ran using the direct solver. Note that rotations in the user-material subroutine (UMAT) in the FE software Abaqus/Standard are by default calculated by the Jaumann spin. Details how to instead impose the constitutional lattice spin are given in Appendix B.

7.1. A Goss-oriented single crystal

Uniaxial tension of a notched single crystal specimen with an initial Goss orientation was simulated, i.e. with the crystal cube rotated 45° around the tensile axis. The diameter of the specimen was 10 mm, the diameter inside the notch was 6.4 mm and the notch radius was 3.6 mm. Note, that due to the single crystal's material model symmetries, only the one-eighth of the specimen was computed. The FE mesh is shown in Fig. 4a. The smooth specimen was meshed with ~18 000 linear three-dimensional eight-nodes elements with selective reduced integration (C3D8). A smaller element size was used close to the mid-section of the specimen to ensure an accurate description of the necking. Kinematic boundary conditions were imposed to the nodes located at the end of the specimen by prescribing an axial displacement of 0.8 mm. The average time increment used was ~0.01 s. On average 7 iterations were needed for the return-mapping algorithm to converge. Hardening constants used for this case, were $R_\alpha^{sat} = 20$ MPa and $\Delta\gamma_\alpha^{sat} = 0.15$ for all α.

The Goss orientation of a crystal is defined by the Euler angles (φ_1, Φ, φ_2) in Bunge's notation as (0°, 45°, 0). Its strong plastic anisotropy gives rise to a pronounced approximately square-shaped neck, as observed experimentally in strongly textured aluminum alloys [84,85].

7.2. A polycrystalline representative volume element (RVE)

The second case simulated by the CPFEM, was uniaxial tension of a RVE for a polycrystalline material. The RVE was modeled as a 1mm^3 cube consisting of 30 grains and was generated in the open source software DREAM.3D,

	Displacement (mm)		
Nodes	u_x	u_y	u_z
A	0	0	0
B	0.4	0	free
C	free	free	0
D	0	free	free

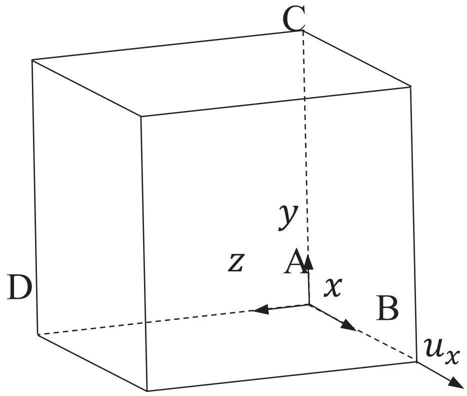


Fig. 5. Kinematic boundary conditions defining uniaxial tension for the RVE.

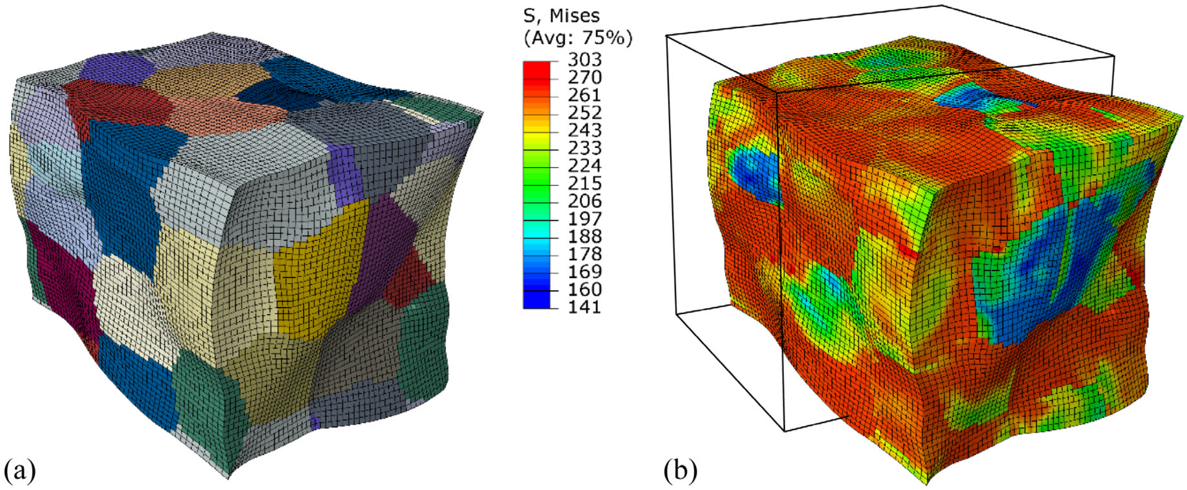


Fig. 6. Uniaxial tension of a polycrystalline RVE consisting of 30 grains meshed with 50 × 50 × 50 C3D8R elements.

see Groeber and Jackson [86]. It was discretized by 50 × 50 × 50 reduced-integration elements (C3D8R). The deformed FE model of the RVE with grain morphology is shown in Fig. 6a. Periodic boundary conditions were applied to the nodes on the exterior boundaries to ensure periodicity [15,87]. The uniaxial tensile mode for the RVE is defined by prescribing boundary conditions to the nodes A, B, C and D (Fig. 5). The nominal strain reached at the end of the simulation was 40%. The von Mises stress in the RVE at the end of the simulation is shown in Fig. 6b. The average time increment used was ~0.004 s. On average 10 Newton iterations and 7 line-search iterations for each Newton iteration were needed for the return-mapping algorithm to converge. Hardening constants used for this case were $R_\alpha^{sat} = 63$ MPa and $\Delta\gamma_\alpha^{sat} = 0.1$ for all α. The simulation took 24 h.

7.3. Comparison of computing times with explicit and implicit FE solvers

The computational efficiency of the implicit CPFEM calculations with the new algorithms presented here, is assessed by comparing to an explicit rate-dependent CPFEM implementation, which is efficient due the use of mass scaling in combination with an adaptive sub-stepping integration scheme using the modified Euler method [22]. This explicit integration scheme is extremely robust and efficient, allowing using an instantaneous strain rate sensitivity m as low as 10^{-5} to explore the very nearly rate-independent stress–strain response. The case chosen for the comparison is a simulation of uniaxial tension up to 1% strain, using an RVE consisting of 60 × 60 × 60 linear elements with full integration. The details of the explicit CPFEM simulation are given in [15]. Note, that the explicit CPFEM is

Table 2
Comparison of computing times with explicit and implicit CPFEM solvers for near-faceted (Schmid) crystal plasticity.

n or $1/m$	Explicit	New implicit
100	16 h	6 h
1 000	120 h	12 h
1 000 000	–	13 h

a rate-dependent formulation, and the FE solver used for this one was LS-DYNA. The purpose here is a coarse comparison of the total computational time. When $m \ll 1$, the rate sensitivity, m, corresponds to the exponent $n \approx 1/m$. In both cases, the simulations were run on the same PC using all 8 cores (see Section 7). The timing results are shown in Table 2.

8. Discussion

8.1. Natural notation

Numerical implementations of return-mapping algorithms involve fourth-order tensors for the elastic stiffness and for the consistent modulus. However, the implementations and numerical schemes at the end of the day must be expressed in terms of matrices and vectors. To make the tensor contractions as simple as possible to handle, the natural vector/matrix notation is applied to represent the tensors involved in the considered crystal-plasticity model. This notation has an orthonormal basis, providing all the convenient properties of the Mandel notation and overcoming the cumbersome distinguishing of stress- and strain-like tensors, as opposed to the Voigt notation. Moreover, it allows more concise algorithm formulations with higher computational efficiency [69]. In this matrix representation notation, the elastic stiffness tensor for crystals with cubic symmetry has a diagonal form. Hence the double contraction in the tensorial version of Hooke's law is reduced to a simple scalar multiplication. Furthermore, it results in an explicit split of the deviatoric and volumetric parts of symmetric second-order tensors, which is advantageous when applied to classical pressure-independent crystal plasticity. Consequently, only the deviatoric part of the constitutive equations enters the plastic corrector of the return mapping algorithm, which reduces by one the system of equations to be solved by the Newton–Raphson iterative method. For the volumetric part, only the elastic predictor, and no plastic corrector, is needed. The natural notation thus enables use of the same algorithm and numerical set-up for cases with only rigid plasticity (e.g. texture calculations), as for cases requiring full elasto-plastic calculations (e.g. CPFEM).

8.2. Line-search algorithms

Similar as for continuum plasticity, [71], a limited convergence of the pure Newton–Raphson method is observed with the regularized crystal yield surface, even with low exponents. For the relatively large strain increments, relevant for implicit FEM as tested in Section 6, the algorithm diverged for $\sim$ 10% of the strain paths with an exponent $n = 5$, without line-search. For a given exponent n, a certain maximum strain increment, $|\Delta\varepsilon|_{\max}$ exists, allowing convergence for all strain paths. Applying the parameters of the two materials in Table 1, it was found that $n \propto 1/|\Delta\varepsilon|_{\max}$, both for the fcc aluminum and the bcc iron. To obtain convergence for $n = 100$, $|\Delta\varepsilon|_{\max} \approx 2.5{\cdot}10^{-5}$ and 10^{-4} for fcc aluminum and bcc iron, respectively. When running implicit FEM simulations, the strain increments required for the desired accuracy, would be considerable larger than that. The overall efficiency of an implicit FEM implementation relies on that the strain increments can be sufficiently large, being controlled by global accuracy rather than the stability of the local iterations in the user subroutine, since most of the computer time is spent between the user subroutine calls.

To ensure stable convergence of the return-mapping algorithm presented here, the line-search algorithm plays a crucial role. Its purpose is to find a scaling α of the increment, $\Delta\boldsymbol{x}$, suggested by the Newton algorithm, i.e. $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \alpha\Delta\boldsymbol{x}$. It makes sure that the scalar merit function, ψ, is always reduced compared to the previous step, i.e. $\psi\left(\boldsymbol{x}^{(n+1)}\right) < \psi\left(\boldsymbol{x}^{(n)}\right)$. In continuum plasticity, line search has been used in several works, see e.g. [69–73]. In this work, two different line-search algorithms were employed and tested: the quadratic line search and line search by a minimization algorithm. The quadratic one returns the α that minimizes a quadratic polynomial, interpolating $\psi(\alpha)$, so that $\psi(0)$, $\psi(1)$ and $\psi'(0)$ are exactly matched. As this approximation becomes less and less precise for large values of exponent n, the number of Newton iteration is increased (see Figs. 1 and 2).

The line search by minimization returns, within the prescribed numerical accuracy, the value of α that minimizes ψ. This new minimization algorithm is a tailor-made version of Brent's algorithm, (see Appendix C). The relative and absolute tolerance, ϵ and ϵ_a, control the precision of finding the minimum of ψ. In general, tight tolerances require a larger number of line-search iterations but lead to a lower number of Newton iterations. This strongly depends on the exponent n. For large n, i.e. $n > 10000$, the number of Newton iterations is greatly reduced, when small tolerances are prescribed. For $n < 100$, the number of Newton iterations is less sensitive to how precisely the minimum of ψ is estimated, and coarser tolerances are more beneficial, reducing the number of line-search iterations to be performed. The relationships $\epsilon = \min(0.3, 1/n)$ and $\epsilon_a = 10^{-2}\epsilon$ are found to work optimally. Using this, the number of Newton iterations remains almost constant for all $n > 100$, while the number of line-search iterations still increases gradually (see Fig. 1 and Fig. 2).

A Newton iteration involves computing the Jacobian and solving a 6 × 6 linear system and is therefore about 4 times more computationally expensive as a line-search iteration. Hence, the monotonically increasing number of Newton iterations as a function of n makes the quadratic algorithm less competitive. The overall timing reveals that the quadratic line search converges faster than the line search by minimization for n up to about 500.

8.3. Approaching the limit of a rate-independent solution

Solutions of the rate-independent crystal plasticity, obeying Schmid's law, suffers from non-uniqueness, i.e. the Taylor ambiguity. Several attempts have been made to obtain a unique solution [37]. As discussed by Holmedal [38] an equivalence exist between the rate-sensitive unique solution, using the viscoplastic law (Eq. (2)) with strain rate sensitivity m, and the unique solution provided by the regularized yield surface defined by Eq. (9) with exponent n, i.e.

$$n = 1 + \frac{1}{m} \tag{46}$$

The proposed return-mapping algorithm enables, for the first time, calculations of extreme solutions a yield surface with exponent n up to a million, corresponding to a strain-rate sensitivity $m = 10^{-6}$. Slip solutions for 10 000 different crystals with a uniformly distribution if their orientations (same as in Section 6) were calculated for each exponent n. For each crystal and each exponent, the relative error of the solutions compared to a limit solution $\dot{\gamma}_\alpha^{\text{lim}}$ can be quantified by

$$\text{rel. error} = \sqrt{\frac{\sum_\alpha \left(\dot{\gamma}_\alpha - \dot{\gamma}_\alpha^{\text{lim}}\right)^2}{\sum_\alpha \left(\dot{\gamma}_\alpha^{\text{lim}}\right)^2}} \tag{47}$$

The limit $\dot{\gamma}_\alpha^{\text{lim}}$ was calculated using $n = 1000\,000$, which in practice is equivalent to the ambiguity limit within the numerical precision. This was done for an FCC structure with $\left\{1\bar{1}1\right\}\langle 011\rangle$ slip systems (Fig. 7a), a BCC structure with $\left\{1\bar{1}0\right\}\langle 111\rangle$ slip systems (Fig. 7b), to which $\left\{\bar{1}2\bar{1}\right\}\langle 111\rangle$ slips (Fig. 7c) and $\left\{1\bar{3}2\right\}\langle 111\rangle$ slips (Fig. 7d) were added. The color map in Fig. 7 shows the distribution of the relative errors for all crystals for each exponent n. Fig. 7 demonstrates the existence of a smallest exponent, for which the solutions still practically are equal to the rate-independent Taylor ambiguity limit, i.e. the same set of slip systems is activated. This is in line with previous findings [37], who showed for an FCC polycrystal, that the texture change is not sensitive to the strain rate sensitivity, m, up to a value less than ∼0.1 (or correspondingly n larger than ∼10).

The results for BCC crystals (Fig. 7b, c and d) show that the smallest exponent giving this limit solution, increases with the number of slip systems. For crystals with 48 slip systems it may be as high as ∼400. This may influence texture calculations. As an example, solutions for $n = 50$ and 400 were calculated by the Taylor model for a rolling reduction of 90%. The $\varphi_2 = 45°$ section of the ODF is shown in Fig. 7e. As expected from Fig. 7d, the texture for the limit case, i.e. with $n = 1000\,000$, was identical to the case of $n = 400$. However, the texture intensity distribution in this section with $n = 50$ is significantly different. According to Larour, Baumer, Dahmen and Bleck [88], the strain-rate sensitivities for various steel grades at room temperature may vary from $m = 0.001$ to 0.02 corresponding to n between 50 and 1000. If a rate-independent simulation of steel is desired, the exponent n should be chosen

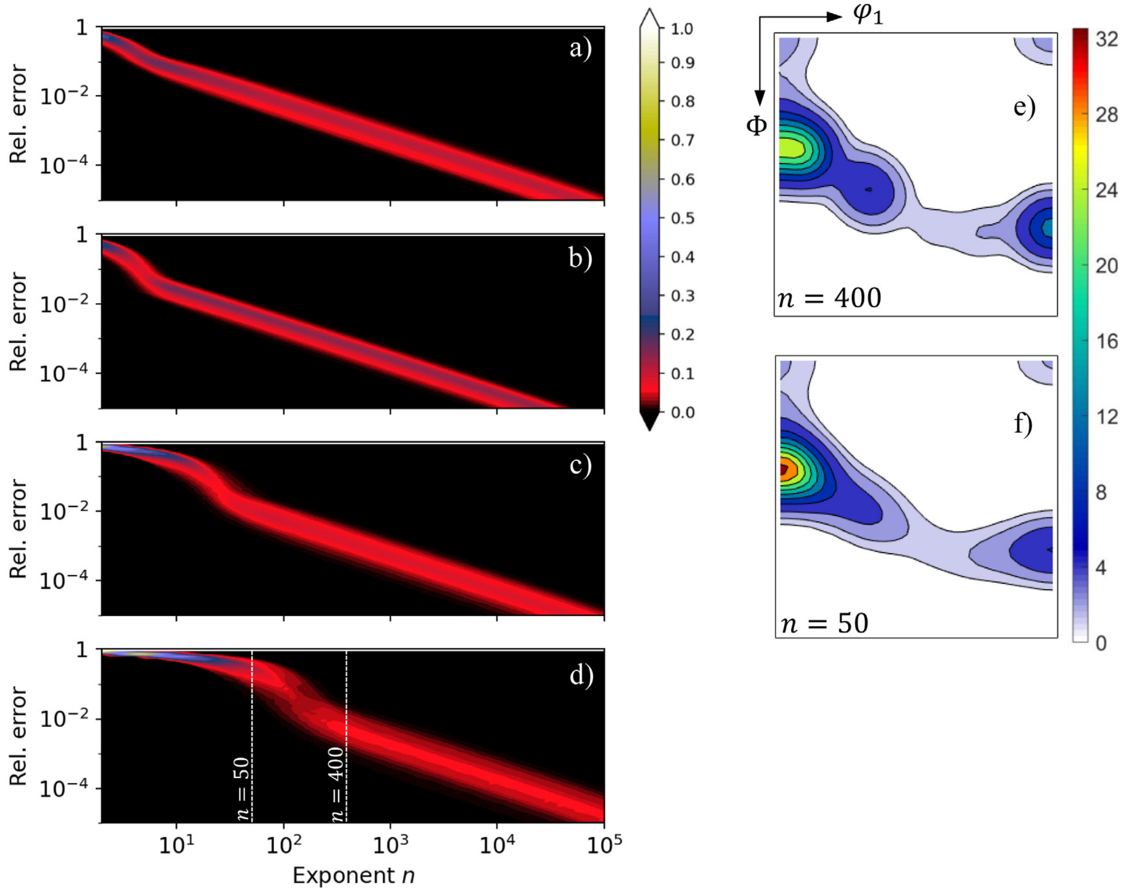


Fig. 7. Density distribution of the relative error (Eq. (47)) of the slip solutions for 10 000 single crystals calculated for (a) FCC structure with $\{1\bar{1}1\}\langle 011\rangle$ slip systems, (b) BCC structure with $\{1\bar{1}0\}\langle 111\rangle$ slip systems, (c) BCC structure with $\{1\bar{1}0\}\langle 111\rangle + \{\bar{1}2\bar{1}\}\langle 111\rangle$ slip systems, and (d) and BCC structure with $\{1\bar{1}0\}\langle 111\rangle + \{\bar{1}2\bar{1}\}\langle 111\rangle + \{1\bar{3}2\}\langle 111\rangle$ slip systems. Rolling textures for a polycrystal with uniform initial texture (1000 orientations) with 48 slip systems same as in figure (d) simulated by the Taylor model for 90% rolling reduction, using (e) $n = 400$ and (f) $n = 50$ Sections for $\varphi_2 = 45°$ are shown.

larger than ∼400 (m less than ∼0.0025) when accounting for the 48 slip systems. However, in many cases it is better to account for a realistic strain-rate sensitivity by choosing an appropriate exponent n in the range below 400 (m above 0.0025). Note that the scaling technique to speed up the viscoplastic calculations [36,89,90], would for bcc miss the correct corner solutions in these cases. This illustrates the need for line-search in the more general cases, to obtain an exact solution.

8.4. Comparison of computing times with explicit and implicit FE solvers

The model implemented in CPFEM, either as part of an implicit or an explicit solver for handling the stress balance and compatibility between the finite elements, is solved incrementally. For each increment, in the implicit approach, iterations are made to find a solution of the set of non-linear finite element equations. In the explicit approach, the dynamic inertial forces applied to the finite elements allow explicit time discretization without iterations at each time increment. Regardless of the choice of an explicit or implicit FE framework, the crystal-plasticity equations are solved for each node, one time-step ahead, as prescribed inside the user subroutine (UMAT). In there, the equations may be solved without iterations in an explicit form, in a fully implicit form or by some type of semi-implicit scheme, in which only some of the variables are solved implicitly. The implicit solution is considerably more expensive, which amounts to the major part of the computational time by explicit FE solvers, whereas for implicit FE solvers the major part of the calculations is related to iterations on the balance between the elements. To compete, the implicit solvers must make significantly larger time steps. For the rate-dependent CPFEM, a thorough comparison of merits of both explicit and implicit CPFEM solvers were reported by Harewood and McHugh [24]. They concluded that when the material deformation is the main part of the simulation, the implicit solver is several times faster. However, in problems with complex contact and sliding conditions, the implicit solver becomes significantly slower. Furthermore, they concluded that their implicit solver struggled to converge and required decreasing time steps with lower strain-rate sensitivities. The latter would not be an issue with similar line-search algorithms as reported here.

The limited conditional stability of the explicit time integration forces the explicit FE solver to use very short time increments, leading to large number of increments to be calculated. For quasi-static problems, a careful, proper mass scaling can help increasing the time step. Then the instability locally related to the integration of the equations for each node in the UMAT subroutine, becomes the bottle neck for reducing the time step. Stability can efficiently be gained by sub-stepping in the UMAT, as reported by Zhang, Hopperstad, Holmedal and Dumoulin [22]. However, with decreasingly small strain-rate sensitivity, increasingly many sub-steps will be required. Hence, it will be beneficial at some small strain-rate sensitivity, to replace the sub-stepping approach by an implicit scheme with line-search, like the one suggested here for the rate-insensitive case. Further investigations are required but are beyond the scope of this work.

When using an implicit solver, the extra computing time required for the return mapping algorithm to solve the material model constitutive equations, is small compared to the time spent by the FE solver to solve the global finite element equations. The stability of the implicit scheme suggested here allows almost arbitrary large time steps and makes this the fastest alternative for e.g. the calculation of the RVE. The time increment is usually controlled by an automatic incrementation routine in the FE software and is limited by the desired accuracy rather than stability requirements. However, in some cases, contact or sliding conditions might limit the allowed time steps significantly.

In non-linear crystal plasticity analysis, Abaqus/Standard uses some variant of Newton's iterative solution method. For each iteration, it is necessary to solve a set of linear equations, which for 3D problems corresponds to a matrix with dimensions proportional to the number of nodes in the power of two. The direct matrix solver in Abaqus/Standard uses a sparse Gauss elimination method for each solution of these linear matrix problems involved. This is the most time-consuming part of the implicit analysis, especially for large models. Unlike the explicit solver, the storage of this matrix during each iteration requires a lot of computer memory, which is the limiting factor for large models with current computers. However, the available computer memory has increased rapidly during the last decades, providing increasing amounts of internal RAM memory and fast serial buffer memory on solid-state disks, even on regular PCs.

9. Conclusion

A numerically stable and efficient fully implicit return-mapping algorithm for rate-independent crystal plasticity was obtained by including a line-search algorithm as part of the Newton iterations and utilizing an improved first guess for the iterations. Fast convergence of the algorithm was proved for any realistic strain step and for very high exponents of the regularized yield surface. Full stability was maintained for an exponent of one million, allowing the Schmid limit to be approached.

It was found that with 12 slip systems, either in BCC or FCC, the set of active slip systems corresponding to the ambiguity solution is obtained whenever the yield surface exponent is larger than $\sim$10. However, for BCC with 48 slip systems, an exponent larger than 1000 will be required. The choice of the exponent is equivalent to prescribing an instant strain-rate sensitivity. To choose the correct exponent for the simulations, or alternatively to run a rate-dependent implementation with the correct strain-rate sensitivity, can therefore be an important issue in BCC texture simulations with 48 slip systems.

A co-rotational hypo-elastic–plastic implementation was made into the user material subroutine of Abaqus/Standard (made available as open source). Efficient computations were demonstrated for uniaxial tension of a polycrystalline representative volume element deformed up to large strains. It is concluded, based on timing of crystal plasticity finite element simulations, that such simulations are significantly faster with the new algorithm in an implicit FE solver than with a state-of-the-art explicit formulation in an explicit FE solver.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Bjorn Holmedal reports financial support was provided by Research Council of Norway.

Acknowledgments

This work was part of the METPLAST project, supported by the Research Council of Norway, FRIPRO grant 315727.

Appendix A. Natural vector/matrix notation

The six independent components of a symmetric 2nd order stress or strain tensor can be represented by a 6×1 vector. Correspondingly, a 4th order symmetric tensor can be expressed as a 3×3 matrix. The Voigt stress vector contains a one-to-one list of the stress components, $\vec{\boldsymbol{\sigma}}^V = (\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{23}, \sigma_{13}, \sigma_{12})^T$. However, in the corresponding Voigt strain tensor, $\vec{\boldsymbol{\varepsilon}}^V = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, 2\varepsilon_{23}, 2\varepsilon_{13}, 2\varepsilon_{12})^T$, the shear strains must be multiplied by a factor two to preserve the inner product of stress and strain. Since the Voigt notation is frequently used in the literature, the natural notation as defined by Eq. (23), will be related to the Voigt notation. However, the natural notation has an orthonormal basis, which makes the notation the same for both strain and stress tensors, like the Mandel notation but unlike the Voigt notation. Furthermore, the natural notation has the advantage that it decouples the hydrostatic pressure, and it diagonalizes the stiffness matrix for cubic crystals, as shown below.

The transformations matrices $\mathbf{T}_\sigma$ and $\mathbf{T}_\varepsilon$, transform the stress vector $\vec{\boldsymbol{\sigma}}^V$ and the strain vector $\vec{\boldsymbol{\varepsilon}}^V$ from the Voigt notation to the vectors $\vec{\boldsymbol{\sigma}} = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)^T$ and $\vec{\boldsymbol{\varepsilon}} = (\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5, \varepsilon_6)^T$ in the natural notation, as

$$\vec{\boldsymbol{\sigma}} = \mathbf{T}_\sigma \vec{\boldsymbol{\sigma}}^V \text{ and } \vec{\boldsymbol{\varepsilon}} = \mathbf{T}_\varepsilon \vec{\boldsymbol{\varepsilon}}^V, \tag{48}$$

where $\vec{\boldsymbol{\sigma}}$ and $\vec{\boldsymbol{\varepsilon}}$ are vector representations of stress and strain in the natural notation, respectively. These transformations are related, as $\mathbf{T}_\sigma = \mathbf{T}_\varepsilon^{-\mathrm{T}}$. The transformation matrices $\mathbf{T}_\sigma$ and $\mathbf{T}_\varepsilon$ read

$$\mathbf{T}_\sigma = \frac{1}{\sqrt{6}} \begin{pmatrix} \sqrt{2} & \sqrt{2} & \sqrt{2} & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ -\sqrt{3} & \sqrt{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\sqrt{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\sqrt{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\sqrt{3} \end{pmatrix}, \mathbf{T}_\varepsilon = \frac{1}{\sqrt{6}} \begin{pmatrix} \sqrt{2} & \sqrt{2} & \sqrt{2} & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ -\sqrt{3} & \sqrt{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{3} \end{pmatrix} \tag{49}$$

Fourth order elasticity stiffness tensors are linear mappings of a second order strain tensor to a second order stress tensor, which in the contracted notations is a 6×6 elastic stiffness matrix, relating the strain vector to the stress vector. From the Voigt notation, they transform into the natural notation as

$$\mathbf{C} = \mathbf{T}_\sigma \mathbf{C}^V \mathbf{T}_\varepsilon^{-1}. \tag{50}$$

The elastic stiffness matrix, for both isotropic and cubic symmetry, are given in the Voigt notation as

$$\mathbf{C}^V_{\text{iso}} = \begin{pmatrix} 2\mu+\lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2\mu+\lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2\mu+\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix}, \mathbf{C}^V_{\text{cub}} = \begin{pmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{pmatrix} \tag{51}$$

where λ is the Lamé's first parameter, μ the shear modulus and C_{11}, C_{12}, C_{22} are cubic elastic constants. In the natural notation, both transform into a diagonal matrix

$$\mathbf{C}_{\text{iso}} = \begin{pmatrix} 3K & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\mu \end{pmatrix}, \mathbf{C}_{\text{cub}} = \begin{pmatrix} C_{11}+2C_{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & C_{11}-C_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{11}-C_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2C_{44} \end{pmatrix} \tag{52}$$

where $K = \lambda + \frac{2}{3}\mu$ is the bulk modulus. This is convenient for numerical computation, e.g. computing of the matrix inversion and for a matrix storage.

Assume an orthonormal transformation tensor $\boldsymbol{R}$ in the cartesian orthonormal basis as $\boldsymbol{R} = R_{ij}\boldsymbol{e}_i \otimes \boldsymbol{e}_j$. The coefficients R_{ij} can be arranged in a 3×3 matrix $\mathbf{R}$ as

$$\mathbf{R} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \tag{53}$$

A 2nd- and 4th order tensor, $\boldsymbol{A}$ and $\mathcal{A}$, respectively, transform as

$$\begin{aligned} \hat{A}_{kl} &= R_{ik}R_{jl}A_{ij} \left(\text{or in tensorial form } \hat{\boldsymbol{A}} = \boldsymbol{R}^{\mathrm{T}} \cdot \boldsymbol{A} \cdot \boldsymbol{R}\right) \text{ and} \\ \hat{\mathcal{A}}_{mnop} &= R_{im}R_{jn}R_{ko}R_{lp}\mathcal{A}_{ijkl} \end{aligned} \tag{54}$$

For expressing 2nd order tensors $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$ and 4th order tensors $\mathcal{A}$ and $\hat{\mathcal{A}}$ in the natural notation as $\vec{\mathbf{a}}$ and $\hat{\vec{\mathbf{a}}}$ (6×1 vectors) and $\mathbb{A}$ and $\hat{\mathbb{A}}$ (6×6 matrix), respectively, there exist a 6×6 matrix $\mathbb{R}$ so that

$$\begin{aligned} \hat{\vec{\mathbf{a}}} &= \mathbb{R}^{\mathrm{T}}\vec{\mathbf{a}} \text{ and} \\ \hat{\mathbb{A}} &= \mathbb{R}^{\mathrm{T}}\mathbb{A}\mathbb{R} \end{aligned} \tag{55}$$

The matrix $\mathbb{R}$ and its transpose $\mathbb{R}^{\mathrm{T}}$ read

$$\mathbb{R} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \left(3R_{33}^2 - 1\right)/2 & \sqrt{3}\left(R_{32}^2 - R_{31}^2\right)/2 & \sqrt{3}R_{32}R_{33} & \sqrt{3}R_{31}R_{33} & \sqrt{3}R_{31}R_{32} \\ 0 & \sqrt{3}\left(R_{23}^2 - R_{13}^2\right)/2 & (R_{11}^2 + R_{22}^2 - R_{12}^2 - R_{21}^2)/2 & R_{22}R_{23} - R_{12}R_{13} & R_{23}R_{21} - R_{13}R_{11} & R_{21}R_{22} - R_{11}R_{12} \\ 0 & \sqrt{3}R_{23}R_{33} & R_{22}R_{32} - R_{21}R_{31} & R_{22}R_{33} + R_{23}R_{32} & R_{21}R_{33} + R_{23}R_{31} & R_{22}R_{31} + R_{21}R_{32} \\ 0 & \sqrt{3}R_{13}R_{33} & R_{32}R_{12} - R_{31}R_{11} & R_{12}R_{33} + R_{13}R_{32} & R_{11}R_{33} + R_{13}R_{31} & R_{12}R_{31} + R_{11}R_{32} \\ 0 & \sqrt{3}R_{13}R_{23} & R_{12}R_{22} - R_{11}R_{21} & R_{12}R_{23} + R_{13}R_{22} & R_{11}R_{23} + R_{13}R_{21} & R_{11}R_{22} + R_{12}R_{21} \end{pmatrix} \tag{56}$$

Appendix B. Co-rotational systems in UMAT in Abaqus/Standard

At the beginning of the time increment spanned between the times t_n and t_{n+1}, the key variables supplied by Abaqus/Standard needed for the crystal plasticity model in UMAT are: the Cauchy stress tensor $\boldsymbol{\sigma}_n$ stored as a 6×1 vector $\vec{\boldsymbol{\sigma}}_n^{\mathrm{V}}$ in Voigt notation (STRESS), the incremental rotation tensor $\Delta\boldsymbol{R}_{n+\frac{1}{2}}$ (DROT) stored as a 3×3 matrix, and the deformation gradient tensor with respect to the initial configuration computed both at the beginning and at the end of the increment, $\boldsymbol{F}_n$ (DFGRD0) and $\boldsymbol{F}_{n+1}$ (DFGRD1), respectively, both stored as 3×3 matrices. At the end of this increment, i.e. at t_{n+1}, the Cauchy stress vector $\vec{\boldsymbol{\sigma}}_{n+1}^{\mathrm{V}}$ (STRESS) needs to be updated and the Jacobian matrix of the constitutive model, i.e. the algorithmic modulus $\left(\frac{\partial \Delta\boldsymbol{\sigma}}{\partial \Delta\boldsymbol{\varepsilon}}\right)_{n+1}$ (DDSDDE) needs to be computed as a 6×6 matrix in Voigt notation.

When a user defined material model is used for continuum elements Abaqus/Standard employs Jaumann stress rate (note that Green–Naghdi stress rate is employed for structural elements in Abaqus/Standard and for any type of elements in VUMAT in Abaqus/Explicit). A local orientation is not applied (*ORIENTATION keyword is not used), hence the components of all tensors are given in the reference (global) coordinate system. Note, that with use of a local orientation, Abaqus/Standard provides the components of all tensors in the local corotated coordinate system at time t_{n+1} rotated from t_n to t_{n+1} by $\Delta\boldsymbol{R}_{n+\frac{1}{2}}$.

The rotation $\Delta\boldsymbol{R}_{n+\frac{1}{2}}$ is calculated based on the Jaumann spin $\boldsymbol{W}_{n+\frac{1}{2}}$ at time $t_{n+\frac{1}{2}} = t_n + \frac{1}{2}\Delta t$ and is passed into the UMAT as DROT. $\Delta\boldsymbol{R}_{n+\frac{1}{2}}$ is calculated by the second-order update scheme by Hughes and Winget [91] as

$$\Delta\boldsymbol{R}^{\mathrm{T}}_{n+\frac{1}{2}} = \left(\boldsymbol{I} - \frac{\Delta t}{2}\boldsymbol{W}_{n+\frac{1}{2}}\right)^{-1} \cdot \left(\boldsymbol{I} + \frac{\Delta t}{2}\boldsymbol{W}_{n+\frac{1}{2}}\right) \tag{57}$$

where $\boldsymbol{W}_{n+\frac{1}{2}}$ is the skew-symmetric part of the velocity gradient $\boldsymbol{L}_{n+\frac{1}{2}}$ at time $t_{n+\frac{1}{2}}$. To account for the rigid body rotations, Abaqus/Standard applies the rotation $\Delta\boldsymbol{R}_{n+\frac{1}{2}}$ of the Cauchy stress before passed to the UMAT as $\boldsymbol{\sigma}_n$ for the next time increment.

However, the crystal constitutive equations and the stress update are calculated in the corotated crystal lattice coordinate system. Hence, the Jaumann rotation of the Cauchy stress by $\Delta\boldsymbol{R}_{n+\frac{1}{2}}$ done by Abaqus must be undone, followed by a rotation $\boldsymbol{R}_n$ into the crystal lattice system at time t_n as

$$\hat{\boldsymbol{\sigma}}_n = \boldsymbol{R}_n^{\mathrm{T}} \cdot \Delta\boldsymbol{R}^{\mathrm{T}}_{n+\frac{1}{2}} \cdot \boldsymbol{\sigma}_n \cdot \Delta\boldsymbol{R}_{n+\frac{1}{2}} \cdot \boldsymbol{R}_n \tag{58}$$

Here $\hat{\boldsymbol{\sigma}}_n$ is the Cauchy stress tensor in the corotated crystal lattice coordinate system, $\boldsymbol{R}_n$ is the rotation defining crystal lattice orientation in the reference system at time t_n. Moreover, the vector $\vec{\boldsymbol{\sigma}}_n^{\mathrm{V}}$ needs to be transformed from Voigt notation into the natural notation, $\vec{\boldsymbol{\sigma}}_n$, by transformation matrix $\mathbf{T}_\sigma$ given in Eq. (49).

$$\begin{aligned} \vec{\boldsymbol{\sigma}}_n &= \mathbf{T}_\sigma \vec{\boldsymbol{\sigma}}_n^{\mathrm{V}} \\ \vec{\hat{\boldsymbol{\sigma}}}_n &= \mathbb{R}_n^{\mathrm{T}} \mathbb{R}_{n+\frac{1}{2}}^{\mathrm{T}} \vec{\boldsymbol{\sigma}}_n \end{aligned} \tag{59}$$

where $\mathbb{R}_n$ and $\mathbb{R}_{n+\frac{1}{2}}$ are 6 × 6 matrices in the natural notation build from $\boldsymbol{R}_n$ and $\Delta\boldsymbol{R}_{n+\frac{1}{2}}$ according to Eq. (56).

As the velocity gradient $\boldsymbol{L}_{n+\frac{1}{2}}$ is not made available by Abaqus/Standard in the UMAT, it can be calculated in the reference system at $t_{n+\frac{1}{2}}$ as

$$\boldsymbol{L}_{n+\frac{1}{2}} = \frac{1}{2}\left(\boldsymbol{L}_n + \boldsymbol{L}_{n+1}\right) = \frac{1}{2\Delta t}\left(\boldsymbol{F}_{n+1} - \boldsymbol{F}_n\right) \cdot \left(\boldsymbol{F}_{n+1}^{-1} + \boldsymbol{F}_n^{-1}\right) \tag{60}$$

The assumption that the velocity gradient is constant between t_n and t_{n+1} in the reference system makes it rotating in the crystal lattice system. Being consistent with the backward Euler numerical scheme, $\hat{\boldsymbol{L}}_{n+1}$ is required in the crystal lattice system at time t_{n+1}. As the crystal lattice rotation, $\boldsymbol{R}_{n+1}$, is not known at the beginning of the time step, it must be found in an iterative manner. A simpler alternative is to assume that the velocity gradient is constant in the crystal lattice system rotating by the Jaumann spin. This will properly account for any rigid body rotation of the crystal which then will not contribute to the inaccuracy of this estimate. The error made is due to difference between the Jaumann and crystal rotation, which is limited by the slip activity during the time increment and is of the order of Δt.

Hence, $\boldsymbol{L}_{n+\frac{1}{2}}$ is rotated into the crystal lattice system by $\boldsymbol{R}_{n+1}^{\mathrm{est}} = \Delta\boldsymbol{R}_{n+\frac{1}{2}} \cdot \boldsymbol{R}_n$ as

$$\hat{\boldsymbol{L}}_{n+1} \approx \left(\boldsymbol{R}_{n+1}^{\mathrm{est}}\right)^{\mathrm{T}} \cdot \boldsymbol{L}_{n+\frac{1}{2}} \cdot \boldsymbol{R}_{n+1}^{\mathrm{est}} \tag{61}$$

Correspondingly, the rate-of-deformation tensor in the crystal coordinate system is $\hat{\boldsymbol{D}}_{n+1} = \operatorname{sym}\left(\hat{\boldsymbol{L}}_{n+1}\right)$, which is converted to a vector in the natural notation by Eq. (23).

After integration at t_{n+1}, the crystal lattice system has rotated by $\boldsymbol{R}_{n+1}$ calculated by Eq. (22). Both the Cauchy stress $\hat{\boldsymbol{\sigma}}_{n+1}$ and the algorithmic modulus $\hat{\mathbf{C}}_{n+1}^{\mathrm{alg}}$ must be rotated from the crystal lattice system back to the reference coordinate system as

$$\begin{aligned} \boldsymbol{\sigma}_{n+1} &= \boldsymbol{R}_{n+1} \cdot \hat{\boldsymbol{\sigma}}_{n+1} \cdot \boldsymbol{R}_{n+1}^{\mathrm{T}} \\ \left(C_{n+1}^{\mathrm{alg}}\right)_{ijkl} &= \left(\hat{C}_{n+1}^{\mathrm{alg}}\right)_{mnop} \left(R_{n+1}\right)_{im} \left(R_{n+1}\right)_{jn} \left(R_{n+1}\right)_{ko} \left(R_{n+1}\right)_{lp} \end{aligned} \tag{62}$$

Finally, the transformation from the natural to Voigt notation is done. The fourth order algorithmic modulus tensor transforms from the natural into Voigt notation by using Eq. (50).

$$\begin{aligned} \vec{\boldsymbol{\sigma}}_{n+1} &= \mathbb{R}_{n+1} \vec{\hat{\boldsymbol{\sigma}}}_{n+1} \\ \vec{\boldsymbol{\sigma}}_{n+1}^{\mathrm{V}} &= \mathbf{T}_\sigma^{-1} \vec{\boldsymbol{\sigma}}_{n+1} \\ \mathbf{C}_{n+1}^{\mathrm{alg}} &= \mathbb{R}_{n+1} \hat{\mathbf{C}}_{n+1}^{\mathrm{alg}} \mathbb{R}_{n+1}^{\mathrm{T}} \end{aligned} \tag{63}$$

$$\left(\mathbf{C}_{n+1}^{\mathrm{alg}}\right)^{\mathrm{V}} = \mathbf{T}_\sigma^{-1} \mathbf{C}_{n+1}^{\mathrm{alg}} \mathbf{T}_\varepsilon \tag{64}$$

Appendix C. Minimization algorithm

The problem is to find iteratively $\alpha_{\min}$ that minimizes $\psi(\alpha)$. Note that the index k for the Newton iterations is dropped from here on out, i.e. $\psi = \psi^{(k+1)}$. The converged value from the previous Newton iteration step corresponds to $\alpha_{(0)} = 0$, and will be counted as iteration $j = 0$ i.e. $\psi_{(0)} = \psi(0) = \psi^{(k)}$. The first new point to be calculated will be a full Newton step with $\alpha_{(1)} = 1$ and $\psi_{(1)} = \psi(1)$. This step will be accepted as the converged line-search solution, if the residual decreases compared to $\psi^{(k)}$, i.e. the algorithm returns $\alpha_{\min} = \psi_{(1)}$ if

$$\psi_{(1)} < 0.99\psi_{(0)} \tag{65}$$

However, if the condition in Eq. (65) is not satisfied, an improved guess is made, based on a quadratic series expansion of the function ψ around $\alpha_{(0)} = 0$. However, to avoid a very small correction, this guess for α is not allowed to be smaller than 0.1.

$$\alpha_{(2)} = \max\left(0.1, \frac{\psi_{(0)}}{\psi_{(1)} + \psi_{(0)}}\right) \tag{66}$$

These two first points are used as the starting point for further iterations to estimate the minimum value of $\psi(\alpha)$. The algorithm book-keeps the trust region and the iterations. The trust borders are initialized as $(\alpha_L, \psi_L) = (0, \psi_{(0)})$ and $(\alpha_R, \psi_R) = (1, \psi_{(1)})$. If $\psi_{(2)} < \min(\psi_{(0)}, \psi_{(1)})$, it is not a border point and is bookkept as a current middle-point, $(\alpha_M, \psi_M) = (\alpha_{(2)}, \psi_{(2)})$. Furthermore, two temporary variables are stored:

$$\alpha_{\text{old}} = \alpha_{(1)}; \alpha_{\text{new}} = \alpha_{(2)}$$

The main algorithm starts at iteration number $j = 3$, and for each proceeding iteration, the following steps are performed:

1. The numbering of the previously calculated points is updated:

   $$\alpha_{\text{oldest}} = \alpha_{\text{old}}$$

   $$\alpha_{\text{old}} = \alpha_{\text{new}}$$

2. Check if trust boundaries can be adjusted.
   If $\psi_{(j)} > \psi_L$ then $(\alpha_R, \psi_R) = (\alpha_{(j)}, \psi_{(j)})$. Proceed directly to point 3
   If $\psi_{(j)} > \psi_R$ then $(\alpha_L, \psi_L) = (\alpha_{(j)}, \psi_{(j)})$.
3. From this point it always holds, that $\psi_M \leq \min(\psi_L, \psi_R)$. Hence, the trust borders are updated as follows:
   If $\alpha_{(j)} < \alpha_M$ then
       If $\psi_{(j)} < \psi_M$ then $(\alpha_R, \psi_R) = (\alpha_{(j)}, \psi_{(j)})$
       Else $(\alpha_L, y_L) = (\alpha_M, \psi_M)$; $(\alpha_M, \psi_M) = (\alpha_{(j)}, \psi_{(j)})$
   If $\alpha_{(j)} > \alpha_M$ then
   If $\psi_{(j)} > \psi_M$ then $(\alpha_L, y_L) = (\alpha_{(j)}, \psi_{(j)})$
   Else $(\alpha_R, \psi_R) = (\alpha_M, \psi_M)$; $(\alpha_M, \psi_M) = (\alpha_{(j)}, \psi_{(j)})$
4. The minimum point of a second order polynomial that passes exactly through the three latest points, is calculated

   $$\alpha_{\text{new}} = \frac{1}{2}\left(\frac{\left(\psi_{(j-2)} - \psi_{(j)}\right)\left(\alpha_{(j-1)} - \alpha_{(j-2)}\right)\left(\alpha_{(j-1)} - \alpha_{(j)}\right)}{\left(\psi_{(j-2)} - \psi_{(j)}\right)\left(\alpha_{(j-1)} - \alpha_{(j)}\right) - \left(\psi_{(j-1)} - \psi_{(j)}\right)\left(\alpha_{(j-2)} - \alpha_{(j)}\right)} + \alpha_{(j-2)} + \alpha_{(j)}\right)$$

5. If α_{new} cannot be calculated due to dividing by zero, the golden section method will be used instead by jumping to point 8.
6. If this new estimate α_{new} will make the algorithm converge slower than the golden section method, that method will be chosen instead. The criterion to estimate this is:
   If $|\alpha_{\text{new}} - \alpha_{\text{old}}| / |\alpha_{\text{old}} - \alpha_{\text{older}}| > g$, go to point 8.
7. At this point the estimate by the quadratic polynomial is accepted. The next $\alpha_{(j+1)}$ is chosen so that α_{new} will be located between $\alpha_{(j+1)}$ and $\alpha_{(j)}$, i.e.

   $$\alpha_{(j+1)} = \alpha_{(j)} + 1.5\left(\alpha_{\text{new}} - \alpha_{(j)}\right)$$

   However, the new point cannot be outside the trust region

   If $\left(\alpha_{(j+1)} < \alpha_L\right)$ or $\left(\alpha_{(j+1)} > \alpha_R\right)$ then $\alpha_{(j+1)} = \alpha_{\text{new}}$

   Go to point 10.

8. Reaching this point, the golden-section method must be applied:
   If $\alpha_R - \alpha_M > \alpha_M - \alpha_L$ then $\alpha_{\text{new}} = g\alpha_M + (1-g)\alpha_R$, $g = \frac{\sqrt{5}-1}{2}$
   Else $\alpha_{\text{new}} = g\alpha_M + (1-g)\alpha_L$
9. The golden-section estimate is applied also as the next value to be evaluated.
   $\alpha_{(j+1)} = \alpha_{\text{new}}$

10. Check for convergence. The relative tolerance , ϵ, and the absolute tolerance, ϵ_a, are preset.
    If $|\alpha_{\text{new}} - \alpha_{\text{old}}| < \epsilon\, |\alpha_{\text{new}}| + \epsilon_a$ then
    the algorithm returns $\alpha_{\min} = \alpha_{\text{new}}$ as the converges solution.
11. Calculate the function value of the latest guess. $\psi_{(j+1)} = \psi\left(\alpha_{(j+1)}\right)$.
12. Continue from Point 1 with the new guess treated as guess number j.

Appendix D. Algorithm for generating approximately evenly distributed points on a hypersphere

A simple, iterative algorithm generates approximately uniformly distributed points on a hypersphere by improving successively the latest guess. The starting point is a random distribution of the points (directions). The following steps are performed:

(1) Pick N points, $\vec{r}_i$, $i = 1, \ldots, N$, of random directions, located on the N_d dimensional hypersphere:

a. To generate each direction, $N_s = 100$ points with random directions are firstly drawn, by that each vector component is assigned a random value between -0.5 and 0.5. It follows from the central limit theorem that a large number of N_s points of such random directions, will obey a normal distribution. The normal distribution in higher dimensions is rotational invariant. Hence the mean value of the N_s points will be a random direction vector in the N_d dimensional space.
b. N such random direction points are drawn and projected onto the N_d dimensional hypersphere by a normalization so that they all have unity length in terms of the L_2 norm.

(2) Search through the N directions, calculate the angle between any two direction vectors

$$\theta_{ij} = \arccos(\vec{r}_i \cdot \vec{r}_j)$$

For each direction, one closest neighbor defines

$$\theta_i^{\text{closest}} = \min_j \theta_{ij}$$

Calculate also the mean angle between any point and its closest neighbor $\theta_{\text{mean}} = \overline{\theta_i^{\text{closest}}}$ and find the two directions with the smallest angle between $\theta_{\min} = \min(\theta_i^{\text{closest}})$. If $\theta_{\min} < 0.95\theta_{\text{mean}}$, try, if possible, to replace one of these two directions by a randomly assigned direction. Keep on trying new random directions until the new directions does not have any neighbor closer than the average closest neighbor. A maximum number of $N \cdot 10^{N_d}$ attempts are made.

(3) If the one of the two directions corresponding to $\theta_{\min}$ cannot be successfully replaced by another direction, try for all points, to move them into a randomly assigned direction that are closer than 0.5 θ_{mean} away from their original direction. The proposed new direction is approved only when the minimum angle to the closest neighbor of the new direction is larger than the current $\theta_{\min}$. Ten attempts are made for each of the N directions in the list. The idea here is that by this, is to "shake" the distribution, enabling replacement of the succeeding $\theta_{\min}$ direction.

(4) The algorithm is repeated from Point 2 by a desired number of iterations, which will each time improve the solution.

The quality of the solution can be measured in several ways. The average or the smallest angle between neighbor directions θ_{mean}, should be as large as possible, while $\theta_{\min}$ and $\theta_{\max} = \max(\theta_i^{\text{closest}})$ should be as close to θ_{mean} as possible. The number of close neighbors should be similar for all points. For point i, this can be measured as the number of directions $\vec{r}_j$ that makes an angle $\theta_{ij} < (1 + \alpha)\,\theta_{\text{mean}}$ with $\vec{r}_i$, where the number of neighbors will increase with increasing α. For a specified α, the number of neighbors should be similar for all points $i = 1, \ldots, N$. The minimum and max number of neighbors for any direction and the average number of neighbors for all directions can be plotted as a function of α. Then these three curves should be as close as possible.

References

[1] P. Franciosi, A. Zaoui, Multislip in f.c.c. crystals a theoretical approach compared with experimental data, Acta Metall. Mater. 30 (1982) 1627–1637.

[2] S.C. Soare, Plasticity and non-schmid effects, Proc. R. Soc. A Math. Phys. 470 (2014) 20130440.

[3] E. Schmid, W. Boas, KristallplastizitätT, Mit Besonderer Berücksichtigung Der Metalle, Springer Verlag, Berlin Heidelberg, 1935.

[4] C. Tome, U.F. Kocks, The yield surface of Hcp crystals, Acta Metall. Mater. 33 (1985) 603–621.

[5] H.R. Wenk, T. Takeshita, R. Jeanloz, G.C. Johnson, Development of texture and elastic-anisotropy during deformation of Hcp metals, Geophys. Res. Lett. 15 (1988) 76–79.

[6] K. Kowalczyk-Gajewska, Modelling of texture evolution in metals accounting for lattice reorientation due to twinning, Eur. J. Mech. A Solids 29 (2010) 28–41.

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

[8] T. Mánik, B. Holmedal, Additional relaxations in the Alamel texture model, Mater. Sci. Eng. A-Struct 580 (2013) 349–354.

[9] R.J. Asaro, A. Needleman, Overview.42. Texture development and strain-hardening in rate dependent polycrystals, Acta Metall. Mater. 33 (1985) 923–953.

[10] D. Peirce, R.J. Asaro, A. Needleman, Material rate dependence and localized deformation in crystalline solids, Acta Metall. Mater. 31 (1983) 1951–1976.

[11] R. Bandyopadhyay, S.E. Gustafson, K. Kapoor, D. Naragani, D.C. Pagan, M.D. Sangid, Comparative assessment of backstress models using high-energy X-ray diffraction microscopy experiments and crystal plasticity finite element simulations, Int. J. Plast. 136 (2021) 102887.

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

[13] F. Zhang, A.F. Bower, R.K. Mishra, K.P. Boyle, Numerical simulations of necking during tensile deformation of aluminum single crystals, Int. J. Plast. 25 (2009) 49–69.

[14] K. Zhang, B. Holmedal, S. Dumoulin, O.S. Hopperstad, An explicit integration scheme for hypo-elastic viscoplastic crystal plasticity, Trans. Nonferr. Met. Soc. China (English Edition) 24 (2014) 2401–2407.

[15] K. Zhang, B. Holmedal, T. Manik, A. Saai, Assessment of advanced Taylor models, the taylor factor and yield-surface exponent for FCC metals, Int. J. Plast. 114 (2019) 144–160.

[16] A.M. Cuitino, M. Ortiz, Computational modelling of single crystals, Modelling Simul. Mater. Sci. Eng. 1 (1993) 225–263.

[17] M. Grujicic, S. Batchu, Crystal plasticity analysis of earing in deep-drawn OFHC copper cups, J. Mater. Sci. 37 (2002) 753–764.

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

[19] A.M. Maniatty, P.R. Dawson, Y.S. Lee, A time integration algorithm for elasto-viscoplastic cubic crystals applied to modelling polycrystalline deformation, Internat. J. Numer. Methods Engrg. 35 (1992) 1565–1588.

[20] J. Rossiter, A. Brahme, M.H. Simha, K. Inal, R. Mishra, A new crystal plasticity scheme for explicit time integration codes to simulate deformation in 3D microstructures: Effects of strain path, strain rate and thermal softening on localized deformation in the aluminum alloy 5754 during simple shear, Int. J. Plast. 26 (2010) 1702–1725.

[21] G. Sarma, T. Zacharia, Integration algorithm for modeling the elasto-viscoplastic response of polycrystalline materials, J. Mech. Phys. Solids 47 (1999) 1219–1238.

[22] K. Zhang, O.S. Hopperstad, B. Holmedal, S. Dumoulin, A robust and efficient substepping scheme for the explicit numerical integration of a rate-dependent crystal plasticity model, Internat. J. Numer. Methods Engrg. 99 (2014) 239–262.

[23] S. Dumoulin, O.S. Hopperstad, T. Berstad, Investigation of integration algorithms for rate-dependent crystal plasticity using explicit finite element codes, Comput. Mater. Sci. 46 (2009) 785–799.

[24] F.J. Harewood, P.E. McHugh, Comparison of the implicit and explicit finite element methods using crystal plasticity, Comput. Mater. Sci. 39 (2007) 481–494.

[25] S. Pruger, B. Kiefer, A comparative study of integration algorithms for finite single crystal (visco-)plasticity, Int. J. Mech. Sci. 180 (2020).

[26] 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.

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

[28] A. Prakash, R.A. Lebensohn, Simulation of micromechanical behavior of polycrystals: finite elements versus fast Fourier transforms, Modelling Simul. Mater. Sci. 17 (2009) 064010.

[29] B. Liu, D. Raabe, F. Roters, P. Eisenlohr, R.A. Lebensohn, Comparison of finite element and fast Fourier transform crystal plasticity solvers for texture prediction, Modelling Simul. Mater. Sci. 18 (2010) 085005.

[30] P. Eisenlohr, M. Diehl, R.A. Lebensohn, F. Roters, A spectral method solution to crystal elasto-viscoplasticity at finite strains, Int. J. Plast. 46 (2013) 37–53.

[31] P. Shanthraj, P. Eisenlohr, M. Diehl, F. Roters, Numerically robust spectral methods for crystal plasticity simulations of heterogeneous materials, Int. J. Plast. 66 (2015) 31–45.

[32] F. Roters, M. Diehl, P. Shanthraj, P. Eisenlohr, C. Reuber, S.L. Wong, T. Maiti, A. Ebrahimi, T. Hochrainer, H.O. Fabritius, S. Nikolov, M. Friák, N. Fujita, N. Grilli, K.G.F. Janssens, N. Jia, P.J.J. Kok, D. Ma, F. Meier, E. Werner, M. Stricker, D. Weygand, D. Raabe, DAMASK – the Düsseldorf advanced material simulation kit for modeling multi-physics crystal plasticity, thermal, and damage phenomena from the single crystal up to the component scale, Comput. Mater. Sci. 158 (2019) 420–478.

[33] A. Eghtesad, K. Germaschewski, R.A. Lebensohn, M. Knezevic, A multi-GPU implementation of a full-field crystal plasticity solver for efficient modeling of high-resolution microstructures, Comput. Phys. Comm. 254 (2020) 107231.

[34] M. Knezevic, H.F. Al-Harbi, S.R. Kalidindi, Crystal plasticity simulations using discrete Fourier transforms, Acta Mater. 57 (2009) 1777–1784.

[35] M. Zecevic, R.J. McCabe, M. Knezevic, A new implementation of the spectral crystal plasticity framework in implicit finite elements, Mech. Mater. 84 (2015) 114–126.

[36] M. Knezevic, M. Zecevic, I.J. Beyerlein, R.A. Lebensohn, A numerical procedure enabling accurate descriptions of strain rate-sensitive flow of polycrystals within crystal visco-plasticity theory, Comput. Methods Appl. Mech. 308 (2016) 468–482.

[37] T. Mánik, B. Holmedal, Review of the Taylor ambiguity and the relationship between rate-independent and rate-dependent full-constraints Taylor models, Int. J. Plast. 55 (2014) 152–181.

[38] B. Holmedal, Regularized yield surfaces for crystal plasticity of metals, Crystals 10 (2020) 1076.

[39] J.W. Hutchinson, Bounds and self-consistent estimates for creep of polycrystalline materials, Proc. R. Soc. Lond. Ser. A 348 (1976) 101–127.

[40] M. Arminjon, A regular form of the schmid law - application to the ambiguity problem, Textures Microstruct. 14 (1991) 1121–1128.

[41] W. Gambin, Plasticity of crystals with interacting slip systems, Eng. Trans. 39 (1991) 303–334.

[42] W. Gambin, Crystal plasticity based on yield surfaces with rounded-off corners, Z. Angew. Math. Mech. 71 (1991) T265–T268.

[43] W. Gambin, F. Barlat, Modeling of deformation texture development based on rate independent crystal plasticity, Int. J. Plast. 13 (1997) 75–85.

[44] K. Kowalczyk, W. Gambin, Model of plastic anisotropy evolution with texture-dependent yield surface, Int. J. Plast. 20 (2004) 19–54.

[45] Y.B. Guan, F. Pourboghrat, F. Barlat, Finite element modeling of tube hydroforming of polycrystalline aluminum alloy extrusions, Int. J. Plast. 22 (2006) 2366–2393.

[46] S.K. Ravi, M. Seefeldt, A. Van Bael, J. Gawad, D. Roose, Multi-scale material modelling to predict the material anisotropy of multi-phase steels, Comput. Mater. Sci. 160 (2019) 382–396.

[47] A. Zamiri, F. Pourboghrat, F. Barlat, An effective computational algorithm for rate-independent crystal plasticity based on a single crystal yield surface with an application to tube hydroforming, Int. J. Plast. 23 (2007) 1126–1147.

[48] G. Kreisselmeier, R. Steinhauser, Systematic control design by optimizing a vector performance index, IFAC Proc. Vol. 12 (1979) 113–117.

[49] R. Esmaeilpour, H. Kim, T. Park, F. Pourboghrat, Z.R. Xu, B. Mohammed, F. Abu-Farha, Calibration of barlat yld2004-18p yield function using CPFEM and 3D RVE for the simulation of single point incremental forming (SPIF) of 7075-O aluminum sheet, Int. J. Mech. Sci. 145 (2018) 24–41.

[50] M. Khadyko, B.H. Frodal, O.S. Hopperstad, Finite element simulation of ductile fracture in polycrystalline materials using a regularized porous crystal plasticity model, Int. J. Fract. 228 (2021) 15–31.

[51] A. Mapar, H. Ghassemi-Armaki, F. Pourboghrat, K.S. Kumar, A differential-exponential hardening law for non-schmid crystal plasticity finite element modeling of ferrite single crystals, Int. J. Plast. 91 (2017) 268–299.

[52] B. Mohammed, T. Park, H. Kim, F. Pourboghrat, R. Esmaeilpour, The forming limit curve for multiphase advanced high strength steels based on crystal plasticity finite element modeling, Mater. Sci. Eng. A-Struct. 725 (2018) 250–266.

[53] B. Pan, H. Tanaka, C. Ling, Y. Shibutani, S.F. Li, Size-dependent yield function for single crystals with a consideration of defect effects, Acta Mech. 230 (2019) 4259–4271.

[54] A.R. Zamiri, F. Pourboghrat, A novel yield function for single crystals based on combined constraints optimization, Int. J. Plast. 26 (2010) 731–746.

[55] M. Schmidt-Baldassari, Numerical concepts for rate-independent single crystal plasticity, Comput. Methods Appl. Mech. 192 (2003) 1261–1280.

[56] S. Pruger, B. Kiefer, A comparative study of integration algorithms for finite single crystal (visco-)plasticity, Int. J. Mech. Sci. 180 (2020) 105740.

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

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

[59] J. Schroder, C. Miehe, Aspects of computational rate-independent crystal plasticity, Comput. Mater. Sci. 9 (1997) 168–176.

[60] L. Scheunemann, P.S.B. Nigro, J. Schroder, P.M. Pimenta, A novel algorithm for rate independent small strain crystal plasticity based on the infeasible primal–dual interior point method, Int. J. Plast. 124 (2020) 1–19.

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

[62] P. Van Houtte, S.Y. Li, M. Seefeldt, L. Delannay, Deformation texture prediction: from the Taylor model to the advanced lamel model, Int. J. Plast. 21 (2005) 589–624.

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

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

[65] H.K. Akpama, M. Ben Bettaieb, F. Abed-Meraim, Localized necking predictions based on rate-independent self-consistent polycrystal plasticity: Bifurcation analysis versus imperfection approach, Int. J. Plast. 91 (2017) 205–237.

[66] P. Lipinski, M. Berveiller, Elastoplasticity of micro-inhomogeneous metals at large strains, Int. J. Plast. 5 (1989) 149–172.

[67] J. Paux, M. Ben Bettaieb, H. Badreddine, F. Abed-Meraim, C. Labergere, K. Saanouni, An elasto-plastic self-consistent model for damaged polycrystalline materials: Theoretical formulation and numerical implementation, Comput. Methods Appl. Mech. 368 (2020) 113138.

[68] M. Zecevic, M. Knezevic, An implicit formulation of the elasto-plastic self-consistent polycrystal plasticity model and its implementation in implicit finite elements, Mech. Mater. 136 (2019) 103065.

[69] T. Mánik, A natural vector/matrix notation applied in an efficient and robust return-mapping algorithm for advanced yield functions, Eur. J. Mech. A Solids 90 (2021) 104357.

[70] T. Seifert, I. Schmidt, Line-search methods in general return mapping algorithms with application to porous plasticity, Internat. J. Numer. Methods Engrg. 73 (2008) 1468–1495.

[71] W.M. Scherzinger, A return mapping algorithm for isotropic and anisotropic plasticity models using a line search method, Comput. Methods Appl. Mech. 317 (2017) 526–553.

[72] A. Perez-Foguet, F. Armero, On the formulation of closest-point projection algorithms in elastoplasticity - part II: Globally convergent schemes, Internat. J. Numer. Methods Engrg. 53 (2002) 331–374.

[73] M. Dutko, D. Peric, D.R.J. Owen, Universal anisotropic yield criterion based on superquadric functional representation.1. Algorithmic issues and accuracy analysis, Comput. Methods Appl. Mech. 109 (1993) 73–93.

[74] M. Khadyko, S. Dumoulin, T. Børvik, O.S. Hopperstad, Simulation of large-strain behaviour of aluminium alloy under tensile loading using anisotropic plasticity models, Comput. Struct. 157 (2015) 60–75.

[75] B. Holmedal, P. Van Houtte, Y.G. An, A crystal plasticity model for strain-path changes in metals, Int. J. Plast. 24 (2008) 1360–1379.

[76] H. Kim, F. Barlat, Y. Lee, S. Bin Zaman, C.S. Lee, Y. Jeong, A crystal plasticity model for describing the anisotropic hardening behavior of steel sheets during strain-path changes, Int. J. Plast. 111 (2018) 85–106.

[77] B. Peeters, B. Bacroix, C. Teodosiu, P. Van Houtte, E. Aernoudt, Work-hardening/softening behaviour of BCC polycrystals during changing strain paths: Ii. TEM observations of dislocation sheets in an if steel during two-stage strain paths and their representation in terms of dislocation densities, Acta Mater. 49 (2001) 1621–1632.

[78] R.M. Brannon, Rotation, reflection, and frame changes, in: Orthogonal Tensors in Computational Engineering Mechanics, IOP Publishing, 2018.

[79] A. Cayley, A.R. Forsyth, The Collected Mathematical Papers of Arthur Cayley, At the University Press, Cambridge, 1889.

[80] U.F. Kocks, C.N. Tomé, H.-R. Wenk, Texture and Anisotropy : Preferred Orientations in Polycrystals and their Effect on Materials Properties, Cambridge University Press, New York ; Cambridge, 1998.

[81] P. Lequeu, P. Gilormini, F. Montheillet, B. Bacroix, J.J. Jonas, Yield surfaces for textured polycrystals—I. Crystallographic approach, Acta Metall. Mater. 35 (1987) 439–451.

[82] A.A. Goldstein, J.F. Price, An effective algorithm for minimization, Numer. Math. 10 (1967) 184–189.

[83] R.P. Brent, Algorithms for Minimization Without Derivatives, Prentice-Hall, Englewood Cliffs ; Hemel Hempstead, 1973.

[84] O. Cazacu, N. Chandola, B. Revil-Baudard, B.H. Frodal, T. Borvik, O.S. Hopperstad, Modeling the effect of notch geometry on the deformation of a strongly anisotropic aluminum alloy, Eur. J. Mech. A Solids 82 (2020) 104004.

[85] B.H. Frodal, L.E.B. Daehli, T. Borvik, O.S. Hopperstad, Modelling and simulation of ductile failure in textured aluminium alloys subjected to compression-tension loading, Int. J. Plast. 118 (2019) 36–69.

[86] M.A. Groeber, M.A. Jackson, DREAM.3D: A digital representation environment for the analysis of microstructure in 3D, Integr. Mater. Manuf. Innov. 3 (2014) 56–72.

[87] A. Saai, S. Dumoulin, O.S. Hopperstad, O.G. Lademo, Simulation of yield surfaces for aluminium sheets with rolling and recrystallization textures, Comput. Mater. Sci. 67 (2013) 424–433.

[88] P. Larour, A. Baumer, K. Dahmen, W. Bleck, Influence of strain rate, temperature, plastic strain, and microstructure on the strain rate sensitivity of automotive sheet steels, Steel Res. Int. 84 (2013) 426–442.

[89] M. Zecevic, I.J. Beyerlein, R.J. McCabe, B.A. McWilliams, M. Knezevic, Transitioning rate sensitivities across multiple length scales: Microstructure–property relationships in the taylor cylinder impact test on zirconium, Int. J. Plast. 84 (2016) 138–159.

[90] W.G. Feather, D.J. Savage, M. Knezevic, A crystal plasticity finite element model embedding strain-rate sensitivities inherent to deformation mechanisms: Application to alloy AZ31, Int. J. Plast. 143 (2021) 103031.

[91] T.J.R. Hughes, J. Winget, Finite rotation effects in numerical-integration of rate constitutive-equations arising in large-deformation analysis, Internat. J. Numer. Methods Engrg. 15 (1980) 1862–1867.