ORIGINAL PAPER

Cyprian Suchocki

Finite element implementation of a certain class of elasto-viscoplastic constitutive models

Abstract In this paper a selected type of elasto-viscoplastic constitutive equations is considered. The viscoplastic model which was proposed by Marquis is generalized by allowing multiple terms describing the isotropic and the kinematic hardening. Furthermore, the presented model formulation enables one to use an arbitrary equivalent plastic strain function to describe the isotropic hardening behavior. What is more, the backstress evolution equation which was proposed by Marquis was modified so that any equivalent plastic strain function can be used in the recovery term now. The general form of the generalized constitutive model obtained this way was subsequently implemented into the finite element method (FEM). For that purpose, the radial-return mapping algorithm was utilized. The consistent tangent operator was derived for the considered class of models and is presented in the paper. A developed user material subroutine (UMAT) which allows one to use the viscoplastic models under consideration in the FEM program CalculiX is attached in the appendix section. A number of numerical simulations were conducted in order to verify the performance of the developed UMAT code.

1 Introduction

The classical small strain plasticity theory finds a wide range of applications in the engineering analysis. This theory is based on the Huber–von Mises–Hencky (HMH) yield criterion and is most commonly utilized to capture the mechanical properties of metals, e.g., [22]. One of the simplest versions of this theory assumes the so-called isotropic hardening rule, i.e., the expansion of the yield surface in the stress space which occurs during the plastic straining. Linear, piecewise linear and nonlinear relations are used to describe the evolution of the yield stress. A slightly more sophisticated approach assumes the usage of kinematic hardening rule, i.e., the translation of the yield surface which occurs during straining. The so-called backstress variable which defines the translation of the yield surface center is, in the simplest case, taken to evolve according to the linear equation proposed by Prager [17]. Furthermore, a group of mixed-hardening models exists which combine the two aforementioned behaviors. The constitutive models listed above are available in practically every software which is utilized for engineering analysis, e.g., [6,9,11,15]. In the case of cyclic loadings, the nonlinear kinematic hardening rules, such as the Armstrong–Frederick (A-F) rule [1], allow to achieve a substantially better agreement between the model predictions and the experimental results.

In order to capture such effects as strain rate sensitivity, creep or stress relaxation, a flow plasticity model must be generalized by formulating the so-called viscoplastic model. One of the classical formulations in this field is the constitutive model by Perzyna [16] and presents itself an extension of the flow plasticity model with isotropic hardening. A viscoplastic model assuming the mixed-hardening rule was proposed by Chaboche [5]. This model is a generalization of the previously proposed Chaboche–Rousselier elastoplastic model [3,4] and

C. Suchocki (✉)
Institute of Fundamental Technological Research, Polish Academy of Sciences, Pawińskiego 5B, 02-106 Warsaw, Poland
E-mail: csuchoc@ippt.pan.pl

is more suitable for predicting material response when subjected to cyclic loadings. One possible extension of the Chaboche viscoplastic model was introduced in the work by Benallal and Marquis [2]. The proposed model assumes utilizing a nonlinear kinematic hardening rule which was obtained by modifying the A-F equation. A function of the equivalent plastic strain was added to the so-called recovery term in the backstress evolution equation. In the original formulation by Marquis, this multiplier was taken in the form of an exponential law. Furthermore, it was assumed that the kinematic hardening behavior is captured by a single backstress variable, whereas the isotropic hardening is governed by the nonlinear rule developed by Voce [21]. The viscous effects follow a nonlinear power law.

In this study a generalization of the viscoplastic model by Marquis is proposed. It is assumed that the kinematic hardening can be described by any defined number of backstress variables, whereas the backstress evolution law proposed by Marquis is extended by allowing the usage of any selected function of the equivalent plastic strain as the additional multiplier in the evolution law's recovery term. Furthermore, the model is generalized by allowing the isotropic hardening behavior to be described by arbitrary function of the equivalent plastic strain.

In the case of more advanced constitutive equations, the analytical solutions to boundary value problems are available for a limited number of cases only. Thus, in order to obtain solutions it is necessary to utilize numerical methods such as the finite element method (FEM), for instance. For that purpose, a numerical calculations algorithm has to be developed for the considered constitutive model.

In the work by Zienkiewicz and Cormeau [24], an algorithm for the finite element (FE) implementation of elasto-viscoplastic constitutive equations was proposed. Different variants of the associated and non-associated flow rule were considered. The cases of HMH and Tresca yield conditions with the assumption of ideal plasticity were analyzed. Furthermore, in one of the proposed model formulations the HMH yield condition with isotropic hardening was used. The aforementioned constitutive equations were utilized to solve several test problems using the introduced model integration algorithm. Moreover, an extended version of the constitutive model was proposed which included additional terms accounting for the viscoelastic effects. These studies were continued as reported in [25] by analyzing other viscoplastic model formulations which would utilize the Drucker–Prager or the Coulomb–Mohr yield conditions.

A review of the elastoplastic constitutive models along with their integration using the radial-return mapping algorithm was presented in [7] with the main focus on the models utilizing the isotropic hardening. The Newton–Raphson (N-R) method was used for solving the nonlinear equations which result from the algorithm. The case of nonlinear stress–strain relation in the elastic range was considered as well. In [8] a modified version of the radial-return algorithm was proposed which is correct for any defined number of internals variables which simulate the loading history effects. The nonlinear equation system which follows from the applied model discretization algorithm was solved using the N-R method. The proposed methodology was utilized to implement into FEM various elastoplastic and elasto-viscoplastic constitutive models. In [10] the FE implementation of two alternative viscoplastic models was discussed, i.e., the Perzyna and the so-called consistency model. A comparative study was conducted.

In the paper by Yang and Feng [23], a viscoplastic model was considered in which additive decomposition of the plastic strain was assumed, i.e., into the steady state and the viscoplastic parts. A numerical integration algorithm was developed for the constitutive model under consideration which was subsequently used to implement the model into the FE analysis program ABAQUS.

Simo and Taylor [19] used the radial-return mapping algorithm to derive the consistent tangent operators for elastoplastic models with mixed hardening behavior. In particular, the nonlinear isotropic hardening was assumed, whereas the kinematic hardening was simulated with a modified linear Prager rule. In the previous work by Suchocki [20], the FE implementation of the cyclic elastoplasticity was discussed. In particular, some generalizations of the Chaboche–Rousselier elastoplastic model were analyzed. Kullig and Wippler [14] developed a numerical integration algorithm for the viscoplastic Chaboche model including a static recovery term in the backstress evolution equation.

In this work the FE implementation of the proposed Marquis viscoplastic model generalization is discussed. The considered constitutive equation makes use of the modified A-F rule to capture the kinematic hardening behavior. Implementing this particular model formulation into FEM has not been thoroughly discussed in the literature yet. The radial-return mapping algorithm was used to integrate the considered constitutive model. A general formula for the fourth-order consistent tangent operator was derived and is presented in the paper. To the best of author's knowledge, the specific forms of the tangent operator and the stress update procedure obtained for the considered class of viscoplastic constitutive models have not been reported in the literature yet. The derived operator along with the relations for the stress were utilized to implement the model into

the non-commercial, open-source FE program CalculiX. This was achieved by taking advantage of UMAT (UserMATerial) subroutine option which is provided by CalculiX, c.f. [9]. In order to verify the performance of the developed UMAT subroutine, numerous FE simulations were performed. Selected results are included in this work. In the case of processes with the homogeneous stress and strain fields, such as uniaxial tension/compression (UT/UC), equibiaxial tension/compression (BT/BC) and simple shear (SS), the developed discrete version of the model was used to derive equation systems which describe the aforementioned deformations. The obtained equations were subsequently utilized to write proper programs in Scilab [18]. These programs allow one to calculate the material's stress response for a given deformation process and strain history. Thus, the homogeneous deformations previously analyzed in CalculiX using the developed UMAT code were simulated again in Scilab with the prepared scripts. The results produced using these two different methods were compared to verify the performance of the UMAT subroutine. A perfect agreement was found between the results generated by CalculiX and those obtained in Scilab. Additionally, some results achieved when analyzing stress relaxation, creep and nonhomogeneous deformations are included. A notched flat bar subjected to cyclic biaxial tension/compression and a thick-walled tube loaded by internal pressure were analyzed. The developed UMAT subroutine's performance has been tested for different types of finite elements. The code is attached in the appendix section.

2 Basic notions

The total stress tensor $\boldsymbol{\sigma}$ is taken to be the sum of the volumetric stress p and the stress deviator $\mathbf{s}$, i.e.,

$$\boldsymbol{\sigma} = p\mathbf{1} + \mathbf{s}, \qquad p = B\epsilon^e, \qquad \mathbf{s} = 2\mu\mathbf{e}^e, \tag{1}$$

with $\epsilon^e = \text{tr}(\boldsymbol{\varepsilon}^e)$ and $\mathbf{e}^e = \boldsymbol{\varepsilon}^e - \frac{1}{3}\text{tr}(\boldsymbol{\varepsilon}^e)\mathbf{1}$, where $\boldsymbol{\varepsilon}^e$ is the small elastic strain tensor, $\mathbf{1}$ is the identity tensor, whereas $\text{tr}(\bullet)$ denotes the trace operation. B and μ are the bulk and the shear elastic moduli, respectively. It is assumed that the total small strain tensor can be additively decomposed into the elastic and the plastic components, i.e.,

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p. \tag{2}$$

Thus, in order to calculate the stress for a given strain it is, according to Eqs. (1) and (2) necessary to know the amount of plastic strain. The plastic strain tensor is determined utilizing the associated flow rule with the assumption of plastic incompressibility, i.e., $\text{tr}(\boldsymbol{\varepsilon}^p) = 0$. The following form of the yield surface is assumed which takes into account both the isotropic and the kinematic hardening behaviors:

$$F(\boldsymbol{\sigma}, \mathbf{X}, \bar{e}^p) = J_2(\mathbf{s} - \mathbf{X}) - k - R(\bar{e}^p) = 0, \tag{3}$$

where $J_2(\mathbf{s} - \mathbf{X}) = \sqrt{\frac{3}{2}(\mathbf{s} - \mathbf{X}) \cdot (\mathbf{s} - \mathbf{X})}$ with $\mathbf{X}$ being the so-called backstress tensor. The initial yield stress is denoted as k, whereas $R(\bar{e}^p)$ is the isotropic hardening function with $\bar{e}^p$ being the equivalent plastic strain defined as:

$$\bar{e}^p(t) = \int_0^t \dot{\bar{e}}^p(t')\mathrm{d}t', \qquad \dot{\bar{e}}^p = \sqrt{\frac{2}{3}\dot{\mathbf{e}}^p \cdot \dot{\mathbf{e}}^p}. \tag{4}$$

The flow rule associated with the yield surface given by Eq. (3) takes the form:

$$\dot{\mathbf{e}}^p = \frac{3}{2}\dot{\bar{e}}^p\bar{\mathbf{n}}, \qquad \bar{\mathbf{n}} = \frac{\mathbf{s} - \mathbf{X}}{J_2(\mathbf{s} - \mathbf{X})}, \tag{5}$$

with $\bar{\mathbf{n}}$ being the normalized effective stress. The effective plastic strain rate is assumed to be given by the following relation:

$$\dot{\bar{e}}^p = \zeta \left\langle \frac{J_2(\mathbf{s} - \mathbf{X}) - R(\bar{e}^p) - k}{K} \right\rangle^m, \tag{6}$$

where $\langle\bullet\rangle$ are Macaulay brackets while K and m are the viscoplasticity constants while $\zeta = 1\,[1/s]$ is the viscosity parameter. In general, the isotropic hardening (or softening) behavior can be governed by any chosen function $R(\bar{e}^p)$. However, this function is commonly assumed in the form of a series of terms proposed by Voce [21], i.e.,

$$R = \sum_{i=1}^N R_i, \qquad R_i = Q_i\left(1 - e^{-b_i\bar{e}^p}\right), \tag{7}$$

with Q_i and b_i $(i = 1, 2, \ldots, N)$ being the material parameters. In the original formulation presented in [2], a single backstress tensor $\mathbf{X}$ was used. However, the total backstress can be taken to be the sum of multiple components, i.e.,

$$\mathbf{X} = \sum_{i=1}^{M} \mathbf{X}^{(i)}. \tag{8}$$

In [3–5] the backstress variables were assumed to follow the nonlinear evolution law developed by Armstrong–Frederick, c.f. [1], i.e.,

$$\dot{\mathbf{X}}^{(i)} = \frac{2}{3} C_i \dot{\mathbf{e}}^p - \gamma_i \dot{\bar{e}}^p \mathbf{X}^{(i)}, \qquad (i = 1, 2, \ldots, M), \tag{9}$$

where C_i and γ_i $(i = 1, 2, \ldots, M)$ are material parameters. In the paper by Benallal and Marquis [2], a modified version of the Armstrong–Frederick (A-F) evolution law was used. The modification concerns introducing a plastic strain function $\varphi(\bar{e}^p)$ in the equation's "recovery" term, i.e.,

$$\dot{\mathbf{X}} = \frac{2}{3} C \dot{\mathbf{e}}^p - \gamma \varphi(\bar{e}^p) \dot{\bar{e}}^p \mathbf{X}, \tag{10}$$

with the following specific form of $\varphi(\bar{e}^p)$ used in [2]:

$$\varphi(\bar{e}^p) = \varphi_\infty + (1 - \varphi_\infty) e^{-\bar{\omega} \bar{e}^p}, \tag{11}$$

where $\varphi_\infty \in \langle 0; 1 \rangle$ and $\bar{\omega}$ are material constants. This concept can be generalized for the total backstress being a sum of M components as given in Eqs. (8) and (9), i.e.,

$$\dot{\mathbf{X}}^{(i)} = \frac{2}{3} C_i \dot{\mathbf{e}}^p - \gamma_i \varphi_i(\bar{e}^p) \dot{\bar{e}}^p \mathbf{X}^{(i)}, \qquad (i = 1, 2, \ldots, M), \tag{12}$$

with

$$\varphi_i(\bar{e}^p) = \varphi_{\infty i} + (1 - \varphi_{\infty i}) e^{-\bar{\omega}_i \bar{e}^p}, \tag{13}$$

and $\varphi_{\infty i} \in \langle 0; 1 \rangle$. In the following derivations the general case is considered, i.e., the forms of the functions $R(\bar{e}^p)$ and $\varphi(\bar{e}^p)$ remain unspecified.

3 Model discretization

For the purpose of discretizing the constitutive model described in the previous section, the radial-return mapping algorithm is utilized. For each computational step $n + 1$, the strain increment is initially assumed to be purely elastic. Thus, the deviatoric stress is calculated as:

$$\mathbf{s}_{n+1} = \mathbf{s}_{n+1}^{pr} - 2\mu \Delta \mathbf{e}^p, \tag{14}$$

with the stress predictor $\mathbf{s}_{n+1}^{pr} = \mathbf{s}_n + 2\mu \Delta \mathbf{e}$. If the condition $J_2(\mathbf{s}_{n+1}^{pr} - \mathbf{X}_n) < k + R(\bar{e}_n^p)$ is satisfied, then no yielding takes place and so $\mathbf{s}_{n+1} = \mathbf{s}_{n+1}^{pr}$ and $\bar{e}_{n+1}^p = \bar{e}_n^p$. If the condition is not satisfied the plastic flow occurs. The equivalent plastic strain is given as $\bar{e}_{n+1}^p = \bar{e}_n^p + \Delta \bar{e}^p$ where the plastic strain increment $\Delta \bar{e}^p$ has to be determined so that the other variables, i.e., $\mathbf{s}_{n+1}$, $\mathbf{X}_{n+1}$ and $\mathbf{e}_{n+1}^p$, can be updated. To this end, Eq. (12) is discretized using the backward Euler method, c.f. [14,20]:

$$\mathbf{X}_{n+1}^{(i)} = w_i \left(\mathbf{X}_n^{(i)} + \frac{2}{3} C_i \Delta \mathbf{e}^p \right), \qquad \mathbf{X}_{n+1} = \sum_{i=1}^{M} \mathbf{X}_{n+1}^{(i)}, \tag{15}$$

with

$$w_i = \frac{1}{1 + \gamma_i \varphi_i(\bar{e}_n^p + \Delta \bar{e}^p) \Delta \bar{e}^p}, \qquad i = 1, 2, \ldots, M. \tag{16}$$

Subtracting $\mathbf{X}_{n+1}$ from Eq. (14) and substituting $\Delta \mathbf{e}^p = \frac{3}{2} \Delta \bar{e}^p \bar{\mathbf{n}}$ after some manipulations leads to:

$$\mathbf{s}_{n+1} - \mathbf{X}_{n+1} = \mathbf{Z} - \left(3\mu + \sum_{i=1}^{M} w_i C_i \right) \Delta \bar{e}^p \bar{\mathbf{n}}, \qquad \mathbf{Z} = \mathbf{s}_{n+1}^{pr} - \sum_{i=1}^{M} w_i \mathbf{X}_n^{(i)}, \tag{17}$$

or, after using Eq. (5)$_2$:

$$(\mathbf{s}_{n+1} - \mathbf{X}_{n+1})\left[1 + \left(3\mu + \sum_{i=1}^{M} w_i C_i\right)\frac{\Delta \bar{e}^p}{J_2(\mathbf{s}_{n+1} - \mathbf{X}_{n+1})}\right] = \mathbf{Z}. \tag{18}$$

During the plastic flow, the discretized form of Eq. (6) is given as:

$$\frac{\Delta \bar{e}^p}{\Delta t} = \left(\frac{J_2(\mathbf{s}_{n+1} - \mathbf{X}_{n+1}) - R(\bar{e}_{n+1}^p) - k}{K}\right)^m. \tag{19}$$

Equation (19) can be solved for $J_2(\mathbf{s}_{n+1} - \mathbf{X}_{n+1})$, i.e.,

$$J_2(\mathbf{s}_{n+1} - \mathbf{X}_{n+1}) = h(\Delta \bar{e}^p) = K\left(\frac{\Delta \bar{e}^p}{\Delta t}\right)^{\frac{1}{m}} + R(\bar{e}_n^p + \Delta \bar{e}^p) + k. \tag{20}$$

Thus,

$$(\mathbf{s}_{n+1} - \mathbf{X}_{n+1})\,\alpha(\Delta \bar{e}^p) = \mathbf{Z}, \qquad \alpha(\Delta \bar{e}^p) = 1 + \left(3\mu + \sum_{i=1}^{M} w_i C_i\right)\frac{\Delta \bar{e}^p}{h(\Delta \bar{e}^p)}. \tag{21}$$

It follows from Eq. (21)$_1$ that:

$$J_2\,(\mathbf{s}_{n+1} - \mathbf{X}_{n+1})\,\alpha(\Delta \bar{e}^p) = J_2(\mathbf{Z}). \tag{22}$$

Thus, the following scalar, nonlinear equation is obtained which can be utilized to compute the effective plastic strain increment $\Delta \bar{e}^p$:

$$r(\Delta \bar{e}^p) = h(\Delta \bar{e}^p)\alpha(\Delta \bar{e}^p) - J_2(\mathbf{Z}) = 0. \tag{23}$$

Equation (23) can be solved iteratively using the Newton–Raphson method. Thus, during every step of the iterative process a correction term is calculated:

$$c_{i+1} = -\frac{r(\Delta \bar{e}_i^p)}{\left.\frac{\partial r(\Delta \bar{e}^p)}{\partial \Delta \bar{e}^p}\right|_{\Delta \bar{e}^p = \Delta \bar{e}_i^p}}, \tag{24}$$

which is subsequently used to update the equivalent plastic strain increment, i.e., $\Delta \bar{e}_{i+1}^p = \Delta \bar{e}_i^p + c_{i+1}$. The derivative of $r(\Delta \bar{e}^p)$ with respect to $\Delta \bar{e}^p$ takes the form:

$$\frac{\partial r(\Delta \bar{e}^p)}{\partial \Delta \bar{e}^p} = \frac{\partial h(\Delta \bar{e}^p)}{\partial \Delta \bar{e}^p}\alpha(\Delta \bar{e}^p) + h(\Delta \bar{e}^p)\frac{\partial \alpha(\Delta \bar{e}^p)}{\partial \Delta \bar{e}^p} - \frac{\partial J_2(\mathbf{Z})}{\partial \Delta \bar{e}^p}, \tag{25}$$

where

$$\frac{\partial h(\Delta \bar{e}^p)}{\partial \Delta \bar{e}^p} = \frac{K}{m \Delta t}\left(\frac{\Delta \bar{e}^p}{\Delta t}\right)^{\frac{1-m}{m}} + \frac{dR(\bar{e}_{n+1}^p)}{d\bar{e}^p} = \kappa, \tag{26}$$

$$\frac{\partial J_2(\mathbf{Z})}{\partial \Delta \bar{e}^p} = \frac{3}{2}\sum_{i=1}^{M} \gamma_i w_i^2 \chi_i \left(\bar{\mathbf{n}}_{n+1} \cdot \mathbf{X}_n^{(i)}\right) \tag{27}$$

and

$$\frac{\partial \alpha(\Delta \bar{e}^p)}{\partial \Delta \bar{e}^p} = \frac{1}{h(\Delta \bar{e}^p)}\left[3\mu + \sum_{i=1}^{M} w_i C_i \left(1 - \gamma_i w_i \chi_i \Delta \bar{e}^p\right) + (1 - \alpha(\Delta \bar{e}^p))\kappa\right]. \tag{28}$$

After computing $\Delta \bar{e}^p$, the backstress components can be updated according to Eq. (15) and subsequently the total deviatoric stress can be calculated, i.e.,

$$\mathbf{s}_{n+1} = \mathbf{X}_{n+1} + \frac{\mathbf{Z}}{\alpha(\Delta \bar{e}^p)}. \tag{29}$$

The presented integration algorithm is used to determine the consistent tangent operator which is required in order to implement the considered model into FEM.

4 Consistent tangent operator

Taking a variation of the deviatoric stress with respect to all quantities leads to:

$$\delta\mathbf{s}_{n+1} = \delta\mathbf{s}_{n+1}^{pr} - 3\mu\delta\bar{e}^p\bar{\mathbf{n}}_{n+1} - 3\mu\Delta\bar{e}^p\delta\bar{\mathbf{n}}_{n+1}. \tag{30}$$

It is seen that the variations $\delta\bar{e}^p$ and $\delta\bar{\mathbf{n}}_{n+1}$ are necessary in order to determine the formula for the consistent tangent operator. The term $\delta\bar{e}^p$ can be found by taking the variation on Eq. (23), i.e.,

$$\delta h(\Delta\bar{e}^p)\alpha(\Delta\bar{e}^p) + h(\Delta\bar{e}^p)\delta\alpha(\Delta\bar{e}^p) - \delta J_2(\mathbf{Z}) = 0. \tag{31}$$

The variations $\delta h(\Delta\bar{e}^p)$, $\delta J_2(\mathbf{Z})$ and $\delta\alpha(\Delta\bar{e}^p)$ need to be found so that Eq. (31) could be solved for $\delta\bar{e}^p$. The variation $\delta h(\Delta\bar{e}^p)$ is determined by taking a variation on Eq. (20), thus,

$$\delta h(\Delta\bar{e}^p) = \kappa(\Delta\bar{e}^p)\delta\bar{e}^p, \qquad \kappa(\Delta\bar{e}^p) = \frac{K}{m\Delta t}\left(\frac{\Delta\bar{e}^p}{\Delta t}\right)^{\frac{1-m}{m}} + \frac{dR(\bar{e}^p)}{d\bar{e}^p}. \tag{32}$$

In order to find $\delta J_2(\mathbf{Z})$, a variation must be taken on Eq. (17)$_2$ which results in:

$$\delta\mathbf{Z} = \delta\mathbf{s}_{n+1}^{pr} - \sum_{i=1}^{M}\delta w_i\mathbf{X}_n^{(i)}, \tag{33}$$

with

$$\delta w_i = -\gamma_i w_i^2\left(\frac{d\varphi_i}{d\bar{e}^p}\Delta\bar{e}^p + \varphi_i\right)\delta\bar{e}^p. \tag{34}$$

Thus,

$$\delta J_2(\mathbf{Z}) = \frac{3}{2}\bar{\mathbf{n}}_{n+1}\cdot\delta\mathbf{s}_{n+1}^{pr} + \frac{3}{2}\sum_{i=1}^{M}\gamma_i w_i^2\left(\bar{\mathbf{n}}_{n+1}\cdot\mathbf{X}_n^{(i)}\right)\chi_i\delta\bar{e}^p, \tag{35}$$

where the following notation is introduced:

$$\chi_i = \frac{d\varphi_i(\bar{e}_{n+1}^p)}{d\bar{e}^p}\Delta\bar{e}^p + \varphi_i(\bar{e}_{n+1}^p). \tag{36}$$

Taking the variation on Eq. (21)$_2$ leads to the formula for $\delta\alpha(\Delta\bar{e}^p)$:

$$\delta\alpha(\Delta\bar{e}^p) = \frac{\delta\bar{e}^p}{h(\Delta\bar{e}^p)}\left[3\mu + \sum_{i=1}^{M}w_i C_i\left(1 - \gamma_i w_i\chi_i\Delta\bar{e}^p\right) + (1-\alpha(\Delta\bar{e}^p))\kappa\right]. \tag{37}$$

After inserting Eqs. (32), (35) and (37) into Eq. (31), it can be solved for $\delta\bar{e}^p$; thus,

$$\delta\bar{e}^p = \frac{\frac{3}{2}\bar{\mathbf{n}}_{n+1}\cdot\delta\mathbf{s}_{n+1}^{pr}}{3\mu + h^*}, \tag{38}$$

where the so-called effective hardening modulus has been introduced:

$$h^* = \sum_{i=1}^{M}w_i C_i\left(1 - \gamma_i w_i\chi_i\Delta\bar{e}^p\right) + \kappa - \frac{3}{2}\sum_{i=1}^{M}\gamma_i w_i^2\left(\bar{\mathbf{n}}_{n+1}\cdot\mathbf{X}_n^{(i)}\right)\chi_i. \tag{39}$$

It can be shown that $\bar{\mathbf{n}}_{n+1} = \frac{\mathbf{Z}}{J_2(\mathbf{Z})}$, e.g., [20]. Taking a variation on this relation $\delta\bar{\mathbf{n}}_{n+1}$ can be found, i.e.,

$$\delta\bar{\mathbf{n}}_{n+1} = \frac{1}{J_2(\mathbf{Z})}\left(\mathbf{I} - \frac{3}{2}\bar{\mathbf{n}}_{n+1}\otimes\bar{\mathbf{n}}_{n+1}\right)\cdot\left(\delta\mathbf{s}_{n+1}^{pr} + \sum_{i=1}^{M}\gamma_i w_i^2\mathbf{X}_n^{(i)}\chi_i\delta\bar{e}^p\right). \tag{40}$$

Substituting Eqs. (38) and (40) into Eq. (30), after some rearrangements and taking into account the volumetric stress component, leads to a linear relation between the variations of stress and strain, i.e.,

$$\delta\boldsymbol{\sigma}_{n+1} = \mathcal{C}_{n+1}^{vp} \cdot \delta\boldsymbol{\varepsilon}_{n+1}, \tag{41}$$

with the consistent viscoplastic tangent operator given as:

$$\mathcal{C}_{n+1}^{vp} = \lambda^*\mathbf{1} \otimes \mathbf{1} + 2\mu^*\mathbf{I} + \left(\frac{h^*}{1 + \frac{h^*}{3\mu}} - 3\mu^*\right)\bar{\mathbf{n}}_{n+1} \otimes \bar{\mathbf{n}}_{n+1}$$

$$- \frac{1 - \frac{\mu^*}{\mu}}{1 + \frac{h^*}{3\mu}} \sum_{i=1}^{M} \gamma_i w_i^2 \chi_i \left[\mathbf{X}_n^{(i)} \otimes \bar{\mathbf{n}}_{n+1} - \frac{3}{2}\left(\bar{\mathbf{n}}_{n+1} \cdot \mathbf{X}_n^{(i)}\right)\bar{\mathbf{n}}_{n+1} \otimes \bar{\mathbf{n}}_{n+1}\right], \tag{42}$$

where $\mathbf{I}$ is the fourth order identity tensor, whereas the introduced effective Lamé constants are given by the following relations:

$$\mu^* = \mu\left(1 - \frac{3\mu\Delta\bar{e}^p}{J_2(\mathbf{Z})}\right), \qquad \lambda^* = B - \frac{2}{3}\mu^*. \tag{43}$$

If $J_2(\mathbf{s}_{n+1}^{pr} - \mathbf{X}_n) < k + R(\bar{e}_n^p)$, the elastic stiffness is used in the computations, i.e.,

$$\mathcal{C}^e = \lambda\mathbf{1} \otimes \mathbf{1} + 2\mu\mathbf{I}, \tag{44}$$

where the Lamé parameter $\lambda = B - \frac{2}{3}\mu$.

5 User material (UMAT) subroutine for cyclic elasto-viscoplasticity

Many commercial and non-commercial FE packages provide some interfaces enabling the users to code their own constitutive laws, e.g., [6,9,11,15]. The open-source FE analysis program CalculiX offers two options for writing a user subroutine UMAT (UserMATerial), i.e., the ABAQUS UMAT or the CalculiX native UMAT interface, c.f. [9]. Usually, the native interface proves to be more efficient than ABAQUS UMAT used under CalculiX. Thus, it was decided to code the constitutive model under study using CalculiX UMAT. Some details regarding the interaction of UMAT subroutine with CalculiX can be found in [9] or [20]. The FE solver provided by CalculiX assumes that the tangent operator is symmetric with 21 independent components only. Thus, the determined operator given by Eq. (42) was symmetrized for the purpose of implementing the considered viscoplastic model via CalculiX UMAT. It has been demonstrated that performing symmetrization does not affect the accuracy of computations and has very little influence on their robustness, e.g., [12,13,20].

6 Exemplary problems

In order to verify the performance of the constitutive model under consideration, a number of FE simulations have been conducted. These included simulations of processes with homogeneous stress and strain fields, i.e., uniaxial tension/compression (UT/UC), equibiaxial tension/compression (BT/BC) and simple shear (SS). The discretized version of the model which is discussed in Sect. 3 was used do derive the process equations for the aforementioned homogeneous deformations. These equations were subsequently used to write Scilab programs [18]. Each program can be utilized to calculate the stress response for a given strain history. The results generated in Scilab were used to verify the computations performed in CalculiX with the use of the written UMAT code. In addition, some more complicated problems were analyzed, i.e., stress relaxation test, creep test, notched flat bar in equibiaxial tension/compression and thick-walled tube loaded by internal pressure.

The material parameters gathered in Table 1 were utilized for the conducted simulations. Thus, a single backstress variable ($M = 1$) and the function $\varphi(\bar{e}^p)$ form as given by Eq. (11) were used, whereas the isotropic hardening behavior was simulated using a single Voce term, i.e., $N = 1$ in Eq. (7). The Lamé parameters are calculated as: $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$, $\mu = \frac{E}{2(1+\nu)}$, whereas the bulk modulus $B = \frac{E}{3(1-2\nu)}$. The results presented in the following paragraphs were obtained for C3D8 elements.[1] The developed UMAT procedure has been tested for other types of finite elements as well.

---

[1] Cubic, three-dimensional, eight nodes.

Table 1 Utilized material parameters of the viscoplastic model based on [2]

No.	Parameter	Name	Value	Unit
1	E	Young's modulus	180,000	MPa
2	ν	Poisson's ratio	0.33	–
3	k	Initial yield stress	114	MPa
4	K	Viscoplasticity parameter	92	MPa
5	m	Viscoplasticity parameter	8	–
6	Q	Isotropic hardening parameter	100	MPa
7	b	Isotropic hardening parameter	32	–
8	C	Kinematic hardening parameter	60,632	MPa
9	γ	Kinematic hardening parameter	572	–
10	φ_∞	Kinematic hardening parameter	0.66	–
11	$\bar{\omega}$	Kinematic hardening parameter	10	–

6.1 Uniaxial tension/compression (UT/UC)

A one-dimensional stress state is considered, i.e.,

$$\boldsymbol{\sigma}_{3\times3} = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{s}_{3\times3} = \begin{bmatrix} \frac{2}{3}\sigma & 0 & 0 \\ 0 & -\frac{1}{3}\sigma & 0 \\ 0 & 0 & -\frac{1}{3}\sigma \end{bmatrix}, \quad \mathbf{X}_{3\times3} = \begin{bmatrix} \frac{2}{3}X & 0 & 0 \\ 0 & -\frac{1}{3}X & 0 \\ 0 & 0 & -\frac{1}{3}X \end{bmatrix}. \tag{45}$$

In each increment the axial stress predictor σ_{n+1}^{pr} is calculated for the given axial strain increment $\Delta\varepsilon_a$:

$$\sigma_{n+1}^{pr} = \sigma_n + E\Delta\varepsilon_a. \tag{46}$$

Subsequently, the yield criterion is checked. If $\left|\sigma_{n+1}^{pr} - X_n\right| < k + R(\bar{e}_n^p)$, the material response is elastic and $\sigma_{n+1} = \sigma_{n+1}^{pr}$ is assumed. Otherwise, the plastic flow occurs and the plastic strain increment $\Delta\bar{e}^p$ must be determined. This is done by solving Eq. (23) which in this particular case takes the form:

$$h(\Delta\bar{e}^p)\alpha(\Delta\bar{e}^p) - |Z| = 0, \tag{47}$$

with $h(\Delta\bar{e}^p)$ being specified by Eq. (20), whereas:

$$\alpha(\Delta\bar{e}^p) = 1 + \left(E + \sum_{i=1}^{M} w_i C_i\right)\frac{\Delta\bar{e}^p}{h(\Delta\bar{e}^p)}, \qquad Z = \sigma_{n+1}^{pr} - \sum_{i=1}^{M} w_i X_n^{(i)}. \tag{48}$$

The `fsolve` function offered by Scilab was used for solving Eq. (47). After obtaining $\Delta\bar{e}^p$, the backstress values are updated, i.e.,

$$X_{n+1}^{(i)} = w_i\left(X_n^{(i)} + C_i\Delta\bar{e}^p\,\text{sgn}(Z)\right), \quad X_{n+1} = \sum_{i=1}^{M} X_{n+1}^{(i)}, \tag{49}$$

where sgn($\bullet$) denotes the signum function, while w_i ($i = 1, 2, \ldots, M$) is given by Eq. (16). Finally, the total axial stress is updated:

$$\sigma_{n+1} = X_{n+1} + \frac{Z}{\alpha(\Delta\bar{e}^p)}. \tag{50}$$

The computations according to Eqs. (46–50) were performed in Scilab for every single increment of the performed simulations.

In CalculiX the UT/UC FE simulations were conducted on a $1\,\text{mm}\times1\,\text{mm}\times1\,\text{mm}$ cube meshed with a single C3D8 element. The considered deformation along with the applied boundary conditions can be seen in Fig. 1a. A kinematic loading in the form of a displacement δ in the direction "1" (Fig. 1a,e) was applied to the cube's $ABCD$ face. Furthermore, a zero displacement in direction "1" was set on the face $EFGH$, a zero displacement in direction "2" was set on the face $AEHD$, and zero displacement in direction "3" was set on face $ABFE$.
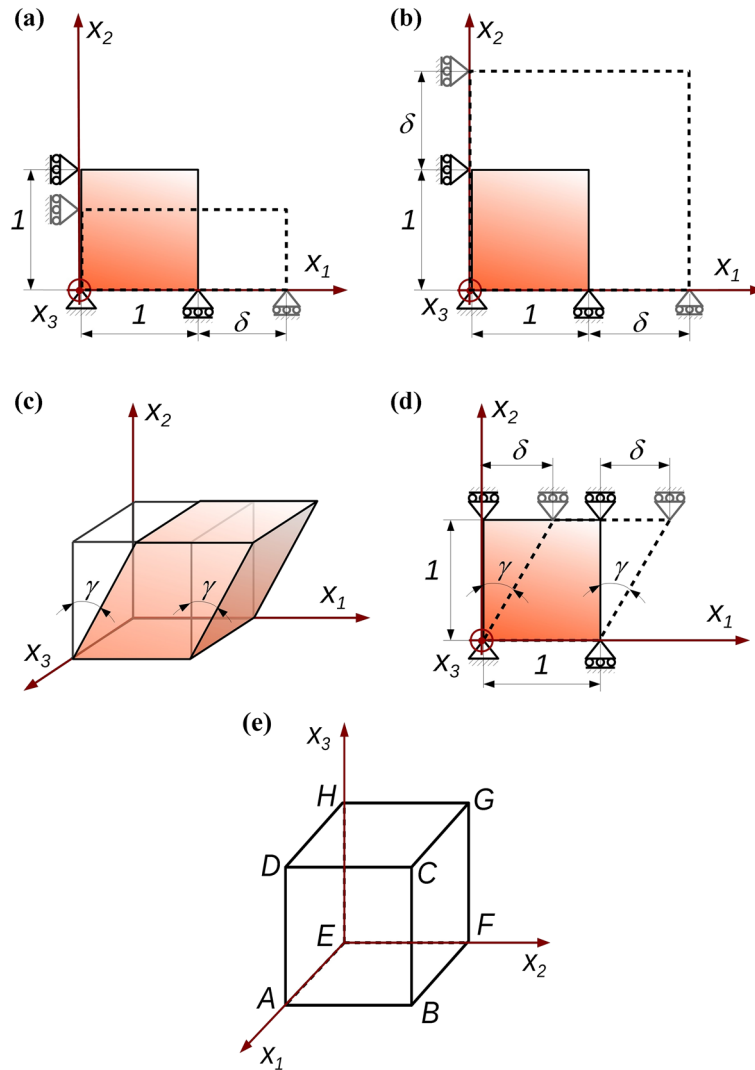
Fig. 1 Deformation of a single FE: **a** uniaxial tension/compression, **b** equibiaxial tension/compression, **c** and **d** simple shear, **e** characteristic points of the geometry

In the first approach a ramp tension test was considered with the maximum displacement value $\delta = 0.005$ mm. The kinematic excitation was applied with different deformation rates, i.e., $1 \times 10^{-6} \ s^{-1}$, $1 \times 10^{-2} \ s^{-1}$ and $1 \ s^{-1}$. The simulations were performed in CalculiX utilizing the developed UMAT code and later repeated in Scilab using the program described above. An excellent agreement between the results obtained in CalculiX and those produced by Scilab was observed (Fig. 2).

In the second approach a cyclic deformation of the cube was analyzed. The kinematic excitation was defined using a saw function with the amplitude of $\delta_a = 0.005$ mm and the period $T = 0.2$ s. A preloading to the maximum value of δ_a was defined which was followed by 10 full deformation cycles. Again, for the verification purposes, the considered deformation process was analyzed both in CalculiX and Scilab. As previously, the obtained results were found to be in an excellent agreement (Fig. 3). Both simulations were later repeated for the cube meshed with 125 C3D8 elements with the same results as previously.
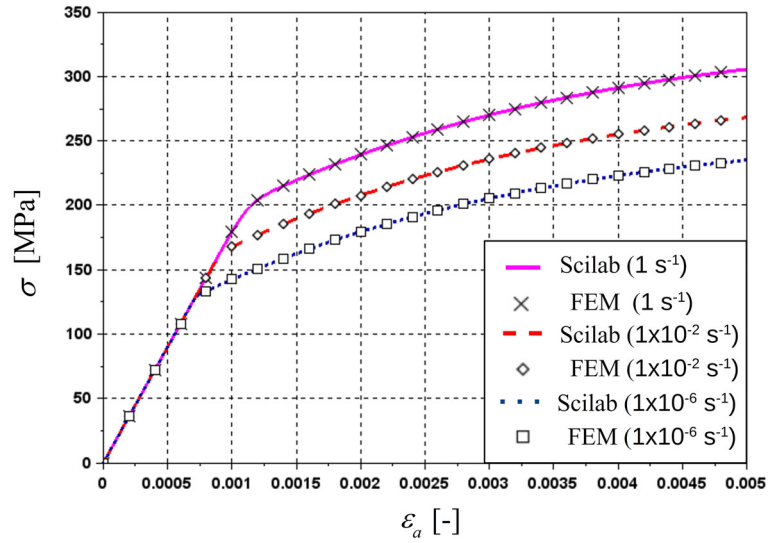
Fig. 2 Normal stress versus axial strain for cube undergoing uniaxial tension/compression with different strain rates
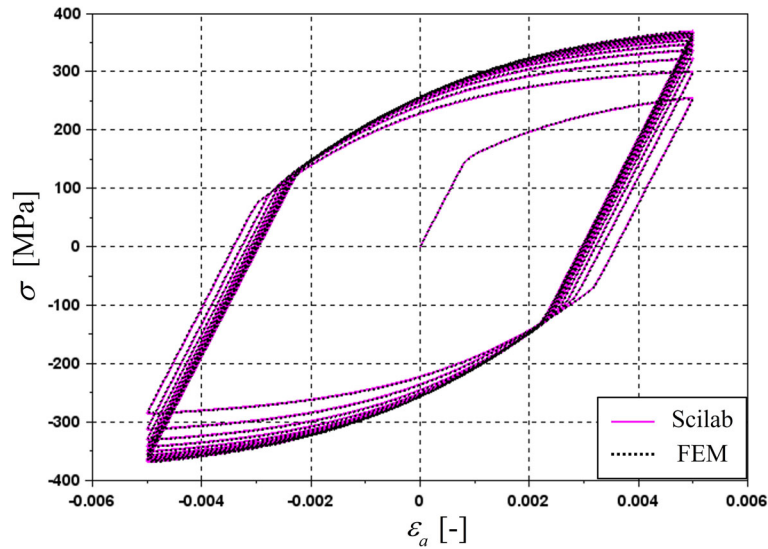
Fig. 3 Normal stress versus axial strain for cube undergoing cyclic uniaxial tension/compression with the strain rate $\dot{\varepsilon} = 1 \times 10^{-3} \, s^{-1}$
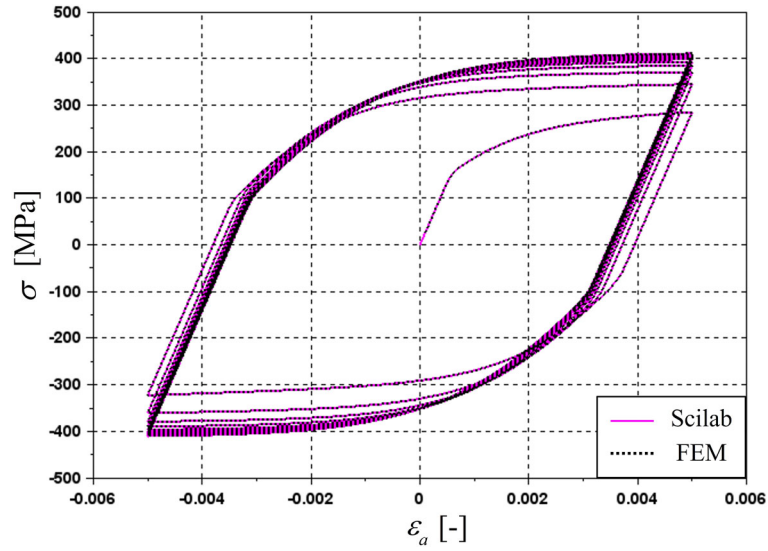
6.2 Equibiaxial tension/compression (BT/BC)

A two-dimensional stress state is considered, i.e.,

$$\boldsymbol{\sigma}_{3\times3} = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad p = \frac{\text{tr}(\boldsymbol{\sigma})}{3} = \frac{2}{3}\sigma, \tag{51}$$

consequently:

$$\mathbf{s}_{3\times3} = \begin{bmatrix} \frac{1}{3}\sigma & 0 & 0 \\ 0 & \frac{1}{3}\sigma & 0 \\ 0 & 0 & -\frac{2}{3}\sigma \end{bmatrix}, \qquad \mathbf{X}_{3\times3} = \begin{bmatrix} \frac{1}{3}X & 0 & 0 \\ 0 & \frac{1}{3}X & 0 \\ 0 & 0 & -\frac{2}{3}X \end{bmatrix}. \tag{52}$$

Fig. 4 Stress versus axial strain for cube undergoing cyclic equibiaxial tension/compression with the strain rate $\dot{\varepsilon} = 1 \times 10^{-3} \, s^{-1}$

The total and plastic strain tensors have the following components:

$$\boldsymbol{\varepsilon}_{3\times3} = \begin{bmatrix} \varepsilon_a & 0 & 0 \\ 0 & \varepsilon_a & 0 \\ 0 & 0 & \varepsilon_l \end{bmatrix}, \qquad \boldsymbol{\varepsilon}^p_{3\times3} = \begin{bmatrix} \varepsilon^p & 0 & 0 \\ 0 & \varepsilon^p & 0 \\ 0 & 0 & -2\varepsilon^p \end{bmatrix}, \tag{53}$$

with ε_a and ε_l being the axial and lateral strains, respectively. The axial stress in the increment $n+1$ can be expressed as follows:

$$\sigma_{n+1} = \sigma^{pr}_{n+1} - \frac{E}{1-\nu}\Delta\varepsilon^p, \qquad \sigma^{pr}_{n+1} = \sigma_n + \frac{E}{1-\nu}\Delta\varepsilon_a. \tag{54}$$

If the condition $\left|\sigma^{pr}_{n+1} - X_n\right| < k + R(\bar{e}^p_n)$ is satisfied, the predictor stress value is taken to be the total updated axial stress, i.e., $\sigma_{n+1} = \sigma^{pr}_{n+1}$. Otherwise, the equivalent plastic strain increment $\Delta\bar{e}^p$ needs to be determined by solving the following equation:

$$h(\Delta\bar{e}^p)\alpha(\Delta\bar{e}^p) - |Z| = 0, \tag{55}$$

where

$$\alpha(\Delta\bar{e}^p) = 1 + \left(\frac{E}{2(1-\nu)} + \sum_{i=1}^{M} w_i C_i\right)\frac{\Delta\bar{e}^p}{h(\Delta\bar{e}^p)}, \tag{56}$$

and

$$Z = \sigma^{pr}_{n+1} - \sum_{i=1}^{M} w_i X^{(i)}_n. \tag{57}$$

where w_i $(i = 1, 2, \ldots, M)$ is given by Eq. (16). After solving Eq. (55), the backstress variables can be updated, i.e.,

$$X_{n+1} = \sum_{i=1}^{M} X^{(i)}_{n+1}, \qquad X^{(i)}_{n+1} = w_i\left(C_i\Delta\bar{e}^p \, \mathrm{sgn}(Z) + X^{(i)}_n\right), \tag{58}$$

and subsequently the total axial stress value can be computed:

$$\sigma_{n+1} = X_{n+1} + \frac{Z}{\alpha(\Delta\bar{e}^p)}. \tag{59}$$

The calculations according to Eqs. (54–59) are performed in every increment of the analysis.

The FE simulation in CalculiX was performed on a 1 mm×1 mm×1 mm cube meshed with a single C3D8 element. The boundary conditions which were applied to the cube are depicted in Fig. 1b. A kinematic excitation was assumed in the form of a displacement δ set on two faces, i.e., the displacement in direction "1" was applied to the cube's face $ABCD$, whereas the displacement in direction "2" was applied to the face $BFGC$ (Fig. 1b,e). A zero displacement in direction "1" was defined on the face $EFGH$, while a zero displacement in direction "2" was set on the face $AEHD$, and finally, a zero displacement in direction "3" was defined on the face $ABFE$. The time history of the displacement δ was assumed in the form of a saw function with the amplitude $\delta_a = 0.005$ mm and the period $T = 20$ s. Consequently, the strain rate in each of the straining axes was equal to 1×10^{-3} s^{-1}. The initial prestraining and 10 subsequent cycles were analyzed. Again, the process equations derived for the discretized version of the model were used to write a Scilab program which calculates the material stress response for a given strain history. This program was utilized to verify the results obtained with CalculiX. An excellent agreement was found (Fig. 4). The considered FE simulations were subsequently repeated for the cube meshed with 125 C3D8 elements with the same results.

6.3 Simple shear (SS)

In the case of simple shear (SS) process, the stress tensors are assumed to have the following components:

$$\boldsymbol{\sigma}_{3\times3} = \mathbf{s}_{3\times3} = \begin{bmatrix} 0 & \tau & 0 \\ \tau & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{X}_{3\times3} = \begin{bmatrix} 0 & X_\tau & 0 \\ X_\tau & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{60}$$

with τ and X_τ being the shear stress and the shear backstress component, respectively. For the analyzed process, the following scalar equations can be obtained:

$$\tau_{n+1} = \tau_{n+1}^{pr} - 2\mu\Delta\varepsilon_s^p, \qquad \tau_{n+1}^{pr} = \tau_n + 2\mu\Delta\varepsilon_s, \tag{61}$$

with τ_{n+1}^{pr} being the shear predictor stress in the increment $n+1$ while ε_s^p is the plastic shear strain increment, whereas $\Delta\varepsilon_s$ is the total shear strain increment. If $\sqrt{3}\left|\tau_{n+1}^{pr} - X_{\tau n}\right| < k + R(\bar{e}_n^p)$ it is assumed that $\tau_{n+1} = \tau_{n+1}^{pr}$. Otherwise, the plastic flow occurs and the equivalent plastic strain increment $\Delta\bar{e}^p$ must be determined. This is done by solving the nonlinear equation:

$$h(\Delta\bar{e}^p)\alpha(\Delta\bar{e}^p) - \sqrt{3}\,|Z| = 0, \tag{62}$$

where

$$\alpha(\Delta\bar{e}^p) = 1 + \left(3\mu + \sum_{i=1}^{M} w_i C_i\right)\frac{\Delta\bar{e}^p}{h(\Delta\bar{e}^p)}, \qquad Z = \tau_{n+1}^{pr} - \sum_{i=1}^{M} w_i X_{\tau n}^{(i)}, \tag{63}$$

while w_i ($i = 1, 2, \ldots, M$) is given by Eq. (16). After calculating the plastic strain increment, the stresses can be updated, i.e.,

$$\tau_{n+1} = X_{\tau n+1} + \frac{Z}{\alpha(\Delta\bar{e}^p)}, \tag{64}$$

and

$$X_{\tau n+1}^{(i)} = w_i \left(C_i \Delta\bar{e}^p \frac{\mathrm{sgn}(Z)}{\sqrt{3}} + X_n^{(i)}\right). \tag{65}$$

The computations listed in Eqs. (61–65) are performed for every analysis step.

As before, the analyzed deformation process was simulated in CalculiX for a cube meshed with a single C3D8 element. The deformed shape and the applied boundary conditions can be seen in Fig. 1c, d with $\gamma = 2\varepsilon_s$ being the shear angle. A kinematic excitation was assumed in the form of the displacement δ in the direction "1" applied to the cube's face $BFGC$ where the displacements in the directions "2" and "3" were set to zero. Due to the small deformations, it is assumed that $\delta/1 = \mathrm{tg}(\gamma) \approx \gamma$. The displacements in all directions were defined as zero on the face $AEHD$. A cyclic deformation was considered with the amplitude $\delta_a = 0.01$ mm and the period $T = 20$ s which resulted in the shear strain rate 1×10^{-3} s^{-1}. A preloading and 10 subsequent cycles were analyzed. Again, the set of equations derived for the discretized version of the model was utilized to write a Scilab program which was used to verify the results obtained in CalculiX. It can be seen in Fig. 5 that once again excellent agreement was found.
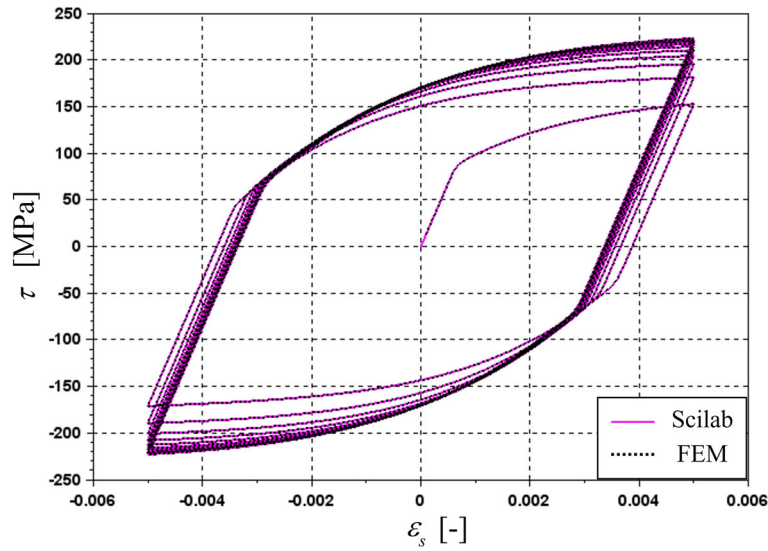
Fig. 5 Shear stress versus shear strain for cube undergoing cyclic simple shear with the strain rate $\dot{\varepsilon} = 1 \times 10^{-3} \ s^{-1}$
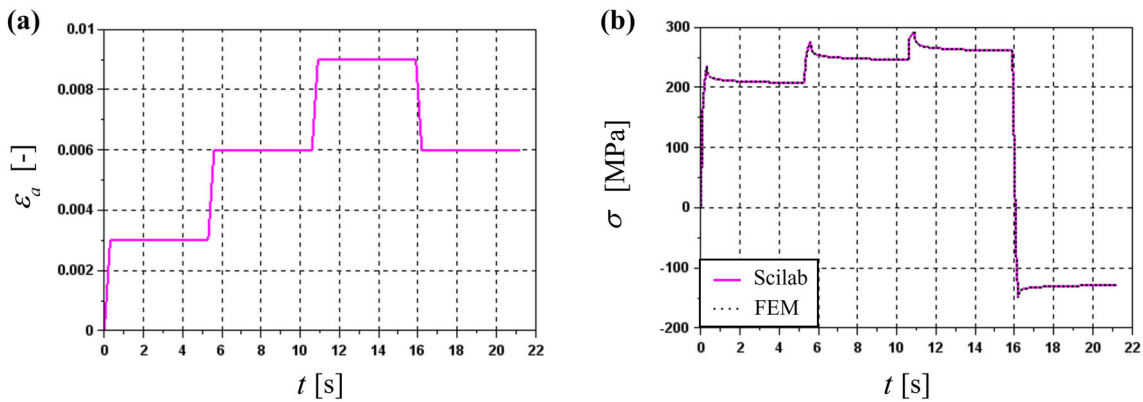

Fig. 6 Cube undergoing stress relaxation in tension/compression: **a** axial strain history, **b** axial stress response

6.4 Stress relaxation

A one-dimensional stress relaxation process was simulated. Again, the FE simulation was performed in CalculiX using a cube meshed with a single C3D8 element. The boundary conditions as described in Sect. 6.1 were used with a different strain history which acted as an excitation (Fig. 6a). The material stress response can be seen in Fig. 6b. In order to validate the obtained FE results, a Scilab program utilizing Eqs. (46–50) was used to simulate the considered deformation process again. It can be seen in Fig. 6b that the results obtained in CalculiX and Scilab are in a perfect agreement.

6.5 Creep

A creep test simulation was performed. In CalculiX, as before, a cube meshed with a C3D8 element was used. The defined boundary conditions are the same as described in Sect. 6.1 with the only difference being a stress excitation applied to the face ABCD instead of a displacement. The assumed stress history and the strain response can be seen in Fig. 7a and b, respectively.

For the purpose of validating the obtained FE results, the axial strain history computed in CalculiX was used as an input for two different Scilab programs in order to check if the correct stress history will be recreated by each one of them. The first program utilizes the radial-return mapping algorithm which for the one-dimensional case leads to Eqs. (46–50). The second Scilab program partially takes advantage of the analytical formulas.
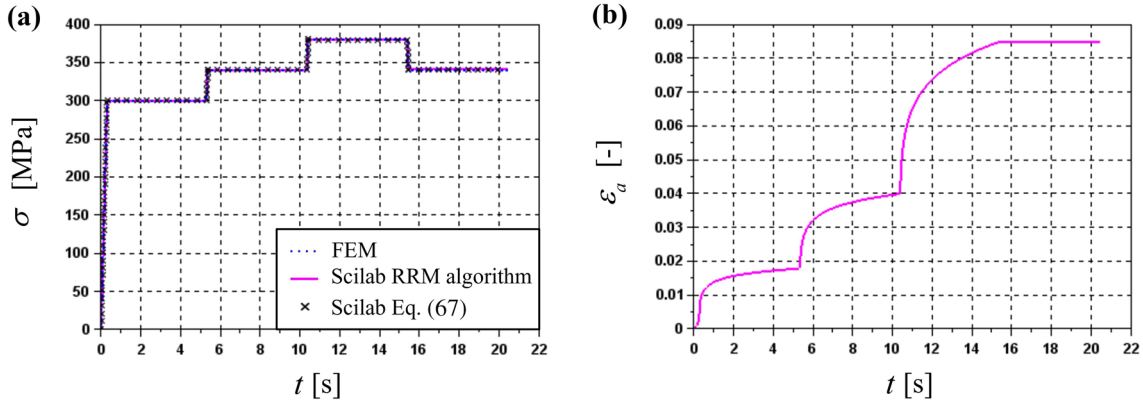
Fig. 7 Cube undergoing creep in tension: **a** axial stress history, **b** axial strain response

According to Eq. (6), for the stress state components defined by Eq. (45), the effective plastic strain is given as:

$$\dot{e}^p = \left(\frac{|\sigma - X| - k - R(\bar{e}^p)}{K}\right)^m,$$ (66)

with $\bar{e}^p = |\varepsilon^p|$ and, $\dot{e}^p = |\dot{\varepsilon}^p|$. After some rearrangements, a formula for the axial stress in the increment $n + 1$ is obtained, i.e.,

$$\sigma_{n+1} = X_{n+1} + \left[k + R(|\varepsilon^p_{n+1}|) + K(|\dot{\varepsilon}^p_{n+1}|)^{1/m}\right] \mathrm{sgn}(\sigma_{n+1} - X_{n+1}),$$ (67)

where $\mathrm{sgn}(\sigma_{n+1} - X_{n+1}) = \mathrm{sgn}(\dot{\varepsilon}^p_{n+1})$ with $\dot{\varepsilon}^p_{n+1} \approx \frac{\Delta \varepsilon^p}{\Delta t}$. The previously presented recurrence-update formula given by Eq. (49)$_1$ was utilized to calculate the backstress component:

$$X_{n+1} = w\left(X_n + C\Delta\varepsilon^p\right),$$ (68)

with

$$w = \frac{1}{1 + \gamma\varphi(|\varepsilon^p_{n+1}|)\,|\Delta\varepsilon^p|}, \qquad \varphi(|\varepsilon^p_{n+1}|) = \varphi_\infty + (1 - \varphi_\infty)e^{-\bar{\omega}|\varepsilon^p_{n+1}|},$$ (69)

where $|\varepsilon^p_{n+1}| = |\varepsilon^p_n| + |\Delta\varepsilon^p_n|$. In the elastic initial range $\sigma_{n+1} = E\varepsilon_{a,n+1}$.

The stress histories recreated by both Scilab programs based on the strain history given in Fig. 7b are presented in Fig. 7a. It is seen that they are in a very good agreement with the original stress history which was used as excitation in the previously performed FE analysis.

6.6 Notched flat bar in biaxial tension/compression

In order to check the performance of the developed UMAT code, a notched flat bar was considered which was subjected to cyclic biaxial tension/compression. The used geometry along with the dimensions, the applied boundary conditions and the coordinate system can be seen in Fig. 8a. The bar was meshed with C3D8 elements with three elements defined along the bar's thickness which equaled to 0.1 mm (Fig. 8b). The kinematic excitation was defined on bar's outer faces in the form of displacement histories δ_1 and δ_2 (Fig. 8a). The displacement δ_1 was set as a ramp elongation until its maximum value of 0.0025 mm which is reached for the time instant $t = 10$ s. The displacement δ_2 was defined with a saw function with the amplitude $\delta_{2a} = 0.005$ mm and the period $T = 40$ s. The total of 10 cycles were performed. The final contour maps obtained for the displacement magnitude, the Huber–von Mises–Hencky (HMH) equivalent stress and the equivalent total strain are shown in Fig. 9.
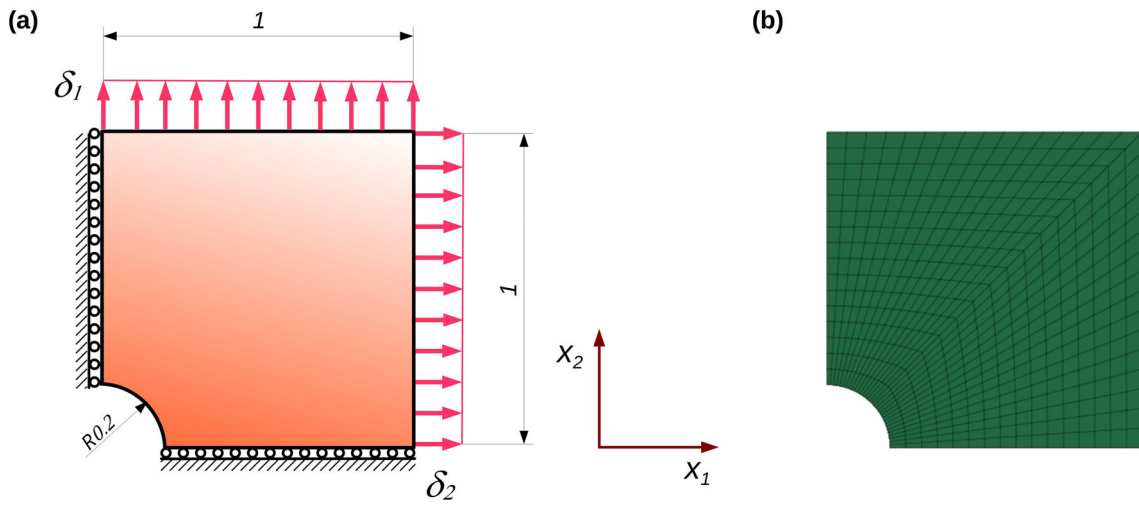
Fig. 8 Notched flat bar undergoing cyclic deformation: **a** boundary conditions and dimensions, **b** FE mesh
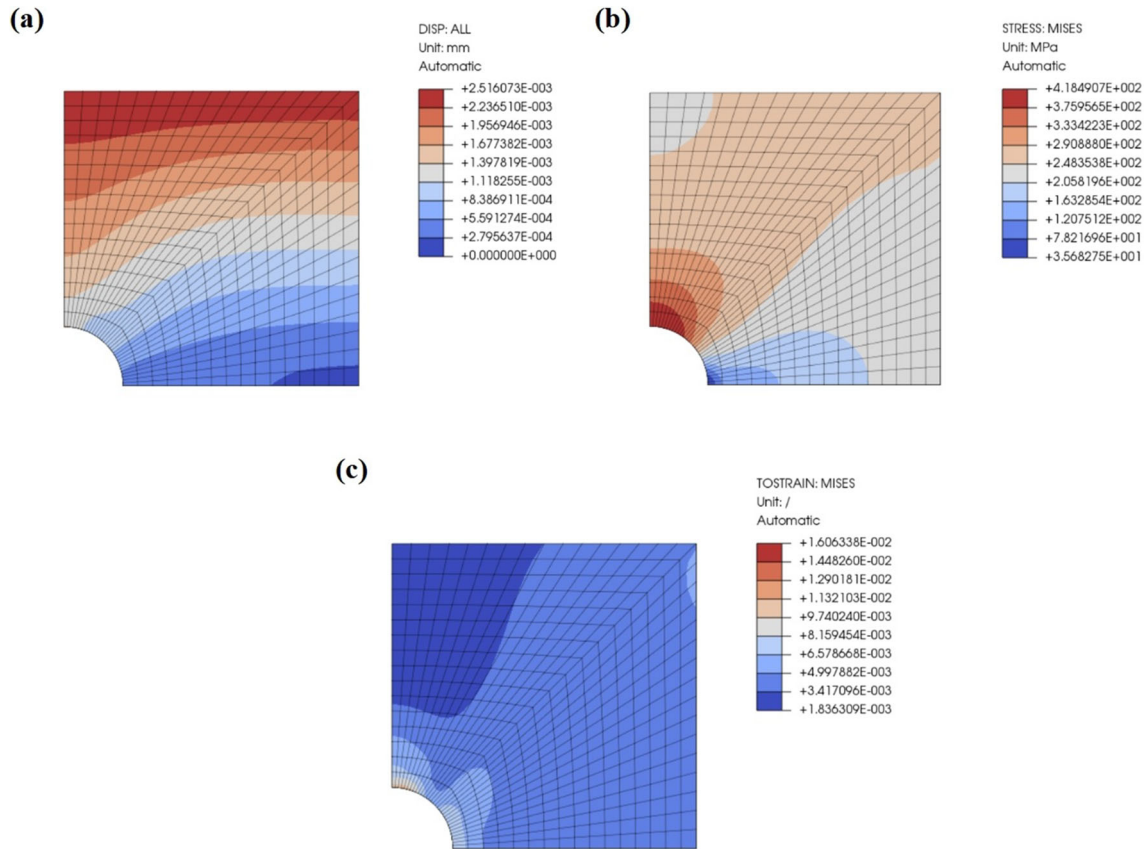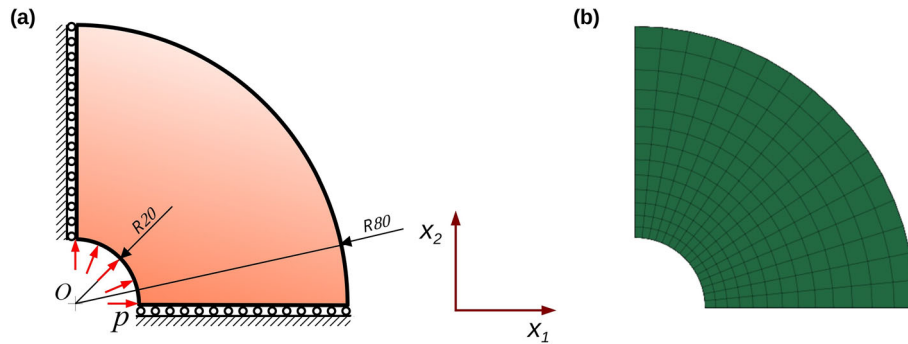

Fig. 9 Notched flat bar undergoing cyclic deformation: **a** displacement magnitude, **b** HMH equivalent stress. **c** equivalent total strain

Fig. 10 Thick-walled tube loaded with internal cyclic pressure: **a** boundary conditions and dimensions, **b** FE mesh
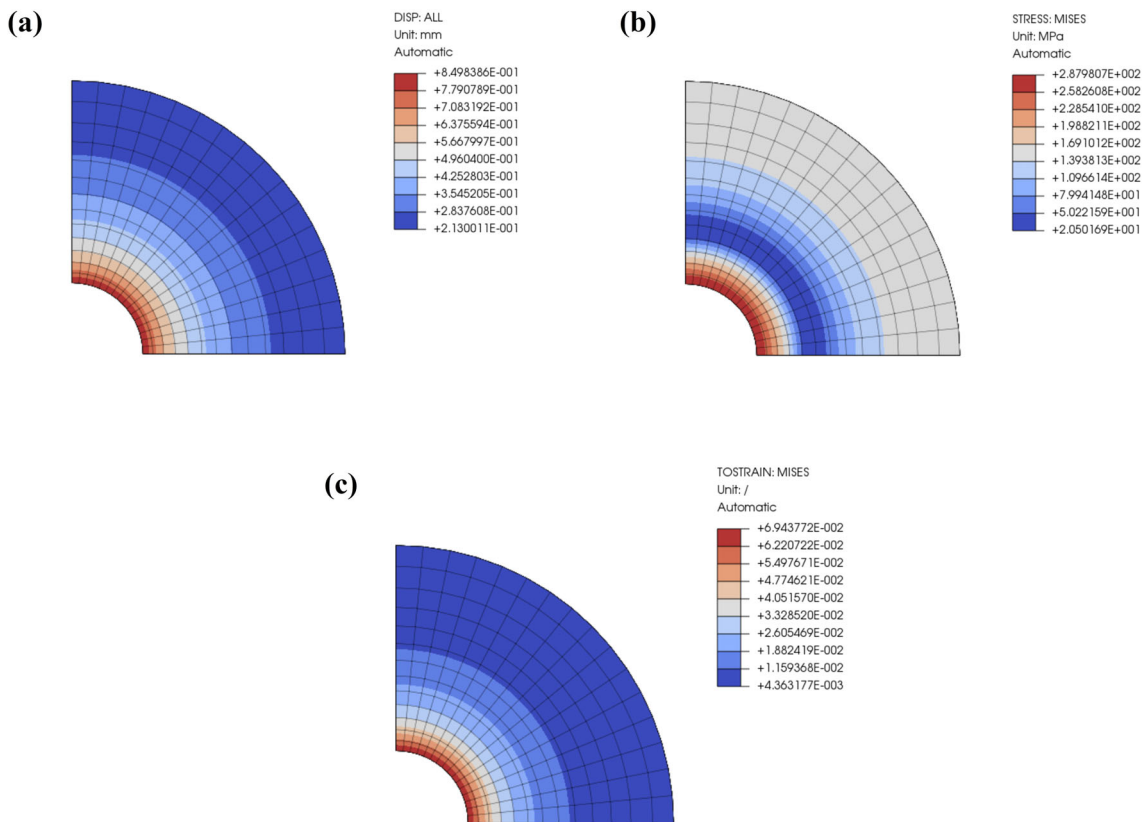
Fig. 11 Thick-walled tube loaded with internal cyclic pressure: **a** displacement magnitude, **b** HMH equivalent stress. **c** equivalent total strain

6.7 Thick-walled tube loaded with internal pressure

As another problem a thick-walled tube loaded with internal pressure was considered. The assumed geometry along with the boundary conditions and the coordinate system are depicted in Fig. 10a. The pressure loading was defined using a saw function with the pressure limits $p_{max} = 470$ MPa, $p_{min} = 0$ MPa and the period $T = 20$ s. The total of 10 cycles were analyzed. The tube was assumed to be 100 mm thick and meshed with C3D8 elements (Fig. 10b). The contour plots showing the final distributions of the displacement magnitude, the HMH stress and the equivalent total strain can be seen in Fig. 11.

7 Conclusions

The viscoplastic model proposed in the work by Benallal and Marquis [2] was analyzed. This model presents itself a certain generalization of the constitutive model which was introduced by Chaboche [3–5]. In this study the model by Benallal and Marquis was further generalized by assuming the isotropic hardening behavior to be governed by any selected function of the equivalent plastic strain $R(\bar{e}^p)$, c.f. Eq. (3). Moreover, the strain function in the "recovery" term of the backstress evolution law was taken to be any chosen function of the plastic strain $\varphi(\bar{e}^p)$, see Eq. (12). The model was also extended by allowing multiple backstress variables to be utilized.

The obtained generalized viscoplastic constitutive equation was subsequently discretized by using the radial-return mapping algorithm. A general formula for the consistent tangent operator was derived. According to the author's best knowledge, this formula has not been presented in the literature before. The discrete model was implemented into the non-commercial, open-source FE program CalculiX by writing a UMAT subroutine which is included in the appendix section.

In order to verify the performance of the written UMAT code, proper equation sets were derived from the discretized version of the model which describe selected homogeneous deformation processes. The obtained equations were in the next step used to write Scilab scripts which computed the stress responses for the deformation histories in question. The simulation results obtained in Scilab were found to be in an excellent agreement to those obtained by performing the FE simulations in CalculiX using the developed UMAT subroutine. What is more, the UMAT code was used to solve several other boundary value problems such as notched flat bar in biaxial tension/compression and thick-walled tube loaded with internal pressure.

Appendix: Coding in Fortran 77

The following UMAT code is provided under the terms of the GNU General Public License (GPL). If you use the UMAT code, please cite this article in your work (book, article, report, etc.).

```fortran
      subroutine umat(amat,iel,iint,kode,elconloc,emec,emec0,
     &        beta,xokl,voj,xkl,vj,ithermal,t1l,dtime,time,ttime,
     &        icmd,ielas,mi,nstate_,xstateini,xstate,stre,stiff,
     &        iorien,pgauss,orab,pnewdt,ipkon)

      implicit none
!
      character*80 amat
!
      integer ithermal,icmd,kode,ielas,iel,iint,nstate_,mi(*),iorien,
     &   ipkon(*)
!
      real*8 elconloc(21),stiff(21),emec(6),emec0(6),beta(6),stre(6),
     &   vj,t1l,dtime,xkl(3,3),xokl(3,3),voj,pgauss(3),orab(7,*),
     &   time,ttime,pnewdt
!
      real*8 xstate(nstate_,mi(1),*),xstateini(nstate_,mi(1),*)

      real*8 one, two, three, six
```

```fortran
      integer k1, knewton

      parameter(one=1.d0, two=2.d0, three=3.d0, six=6.d0)
!
      real*8 eelas(6),eplas(6),
     1  backstre(6),backstre1(6),
     2  flow(6),Z(6)
!
      real*8 e,nu,G,G2,G3,B,lam,kyield,Q1,b1,C1,gamma1,
     1 sHMH,syield,shydro,
     2 eqplas,deqplas,
     3 effhrd,effG,effG2,effG3,efflam,effh,
     4 term1,term2,term3,term4,term5,term7,treelas,
     5 eqZ,w1,R1,alfa,
     6 dRdep,dJ2Zdep,dalfadep,drhsddep,
     7 rhs,FlowdotBstre1,threetwo,
     8 K,n,kappa,chi1,phi1,phiinf1,omega1,dphi1dep,h

      integer newton

      real*8 toler

      newton=1000
      toler=1.D-6
! ----------------------------------------------------------------------
!              CALCULIX UMAT FOR MARQUIS VISCOPLASTIC MODEL
!                      (SMALL STRAIN FORMULATION)
!
!                      POLISH ACADEMY OF SCIENCES
!          INSTITUTE OF FUNDAMENTAL TECHNOLOGICAL RESEARCH
!
!                   CYPRIAN SUCHOCKI, DECEMBER 2022
! ----------------------------------------------------------------------
!      elconloc(1) - e  Young's modulus
!      elconcoc(2) - nu Poisson's ratio
!      elconloc(3) - kyield initial yield stress
!      elconcoc(4) - Q1 isotropic hardening parameter
!      elconloc(5) - b1 isotropic hardening parameter
!      elconcoc(6) - C1 kinematic hardening parameter
!      elconloc(7) - gamma1 kinematic hardening parameter
!      elconcoc(8) - phiinf1 kinematic hardening parameter
!      elconloc(9) - omega1 kinematic hardening parameter
!      elconcoc(10) - K viscosity parameter
!      elconloc(11) - n viscosity parameter
! ----------------------------------------------------------------------
!      Local arrays:
!      backstre(6) - back stress
!      backstre1(6) - back stress component no. 1
!      eelas(6) - elastic strain
!      eplas(6) - plastic strain
!      flow(6) - flow direction
!      Z(6) - auxiliary stress
!
!      Local variables:
!      G - shear modulus
!      B - bulk modulus
```

```
!        lam - lambda (Lame constant)
!        treelas - trace of the lastic strain tensor
!        eqplas - equivalent plastic strain
!        deqplas - equivalent plastic strain increment
!        sHMH - equivalent stress
!        syield - yield stress
!        shydro - hydrostatic stress
!        effh - effective hardening parameter
!        effG - effective shear modulus
!        effG2 - effective shear modulus times two
!        effG3 - effective shear modulus times three
!        efflam - effective lambda constant
! -----------------------------------------------------------------
!
!        Material parameters
!
        e=elconloc(1)  ! Young's modulus
        nu=elconloc(2)  ! Poisson's ratio
        kyield=elconloc(3)  ! initial yield stress
        Q1=elconloc(4)  ! isotropic hardening parameter
        b1=elconloc(5)  ! isotropic hardening parameter
        C1=elconloc(6)  ! kinematic hardening parameter
        gamma1=elconloc(7)  ! kinematic hardening parameter
        phiinf1=elconloc(8)  ! kinematic hardening parameter
        omega1=elconloc(9)  ! kinematic hardening parameter
        K=elconloc(10)  ! viscosity parameter
        n=elconloc(11)  ! viscosity parameter

        G=e/(two*(one+nu))  ! shear modulus
        G2=two*G
        G3=three*G
        B=e/(three*(one-two*nu))  ! bulk modulus
        lam=B-two/three*G  ! lambda

        threetwo=three/two
!
!        Extract state variables
!
        do k1=1,6
            backstre(k1)=xstateini(k1,iint,iel)
            backstre1(k1)=xstateini(k1+6,iint,iel)
            eplas(k1)=xstateini(k1+12,iint,iel)
        end do
        eqplas=xstateini(19,iint,iel)
!
!        Calculate elastic strain
!
        do k1=1,6
            eelas(k1)=emec(k1)-eplas(k1)
        end do
!
!        Calculate predictor stress
!
        treelas=eelas(1)+eelas(2)+eelas(3)

        stre(1)=lam*treelas+G2*eelas(1)
```

```fortran
        stre(2)=lam*treelas+G2*eelas(2)
        stre(3)=lam*treelas+G2*eelas(3)
        stre(4)=G2*eelas(4)
        stre(5)=G2*eelas(5)
        stre(6)=G2*eelas(6)
!
!      HMH equivalent stress
!
        sHMH=(stre(1)-backstre(1)-stre(2)+backstre(2))**2
     1  +(stre(2)-backstre(2)-stre(3)+backstre(3))**2
     2  +(stre(3)-backstre(3)-stre(1)+backstre(1))**2
        do k1=4,6
             sHMH=sHMH+six*(stre(k1)-backstre(k1))**2
        end do
        sHMH=sqrt(sHMH/two)
!
!      Yield stress
!
        R1=Q1*(one-exp(-b1*eqplas))
        syield=kyield+R1
!
!      Determine if actively yielding
!
        if(sHMH.gt.(one+toler)*syield) then
!
!         Separate the deviatoric from the hydrostatic stress
!         Calculate the flow direction
!
           shydro=(stre(1)+stre(2)+stre(3))/three
!
!         Solve for equivalent plastic strain, Newton iteration
!
           deqplas=1.d-60
           do knewton=1,newton
              call calcterm(Q1,b1,kyield,R1,gamma1,C1,G3,stre,
     1         backstre1,eqplas+deqplas,deqplas,shydro,syield,w1,alfa,Z,
     2         eqZ,dRdep,dJ2Zdep,dalfadep,FlowdotBstre1,term2,threetwo,
     3         kappa,chi1,K,n,dtime,phi1,phiinf1,omega1,dphi1dep,h)

              rhs=h*alfa-eqZ
              drhsddep=kappa*alfa+h*dalfadep-dJ2Zdep
              deqplas=deqplas-rhs/drhsddep
              if(abs(rhs).lt.toler*syield) goto 10
           end do
10         continue

           eqplas=eqplas+deqplas
           call calcterm(Q1,b1,kyield,R1,gamma1,C1,G3,stre,
     1 backstre1,eqplas,deqplas,shydro,syield,w1,alfa,Z,
     2 eqZ,dRdep,dJ2Zdep,dalfadep,FlowdotBstre1,term2,threetwo,
     3 kappa,chi1,K,n,dtime,phi1,phiinf1,omega1,dphi1dep,h)
!
!         Calculate the flow direction
!
           do k1=1,6
                flow(k1)=Z(k1)/eqZ
```

```fortran
          end do
!
!         Calculate symmetrized Material Jacobian for active yielding
!
          effG=G*(one-G3*deqplas/eqZ)
          effG2=two*effG
          effG3=three*effG
          efflam=B-effG2/three

          effh=term2+kappa-dJ2Zdep
          effhrd=effh/(one+effh/G3)-effG3

          term3=(one-effG/G)/(one+effh/G3)*gamma1*chi1*w1**two
          term4=threetwo*FlowdotBstre1

          if(icmd.ne.3) then

            stiff(1)=effG2+efflam+flow(1)*flow(1)*effhrd
     1              -term3*(backstre1(1)*flow(1)
     2              -term4*flow(1)*flow(1))
            stiff(2)=efflam+flow(1)*flow(2)*effhrd
     1              -term3*(0.5d0*(backstre1(1)*flow(2)
     2              +backstre1(2)*flow(1))-term4*flow(1)*flow(2))
            stiff(3)=effG2+efflam+flow(2)*flow(2)*effhrd
     1              -term3*(backstre1(2)*flow(2)
     2              -term4*flow(2)*flow(2))
            stiff(4)=efflam+flow(1)*flow(3)*effhrd
     1              -term3*(0.5d0*(backstre1(1)*flow(3)
     2              +backstre1(3)*flow(1))-term4*flow(1)*flow(3))
            stiff(5)=efflam+flow(2)*flow(3)*effhrd
     1              -term3*(0.5d0*(backstre1(2)*flow(3)
     2              +backstre1(3)*flow(2))-term4*flow(2)*flow(3))
            stiff(6)=effG2+efflam+flow(3)*flow(3)*effhrd
     1              -term3*(backstre1(3)*flow(3)
     2              -term4*flow(3)*flow(3))
            stiff(7)=flow(1)*flow(4)*effhrd
     1              -term3*(0.5d0*(backstre1(1)*flow(4)
     2              +backstre1(4)*flow(1))-term4*flow(1)*flow(4))
            stiff(8)=flow(2)*flow(4)*effhrd
     1              -term3*(0.5d0*(backstre1(2)*flow(4)
     2              +backstre1(4)*flow(2))-term4*flow(2)*flow(4))
            stiff(9)=flow(3)*flow(4)*effhrd
     1              -term3*(0.5d0*(backstre1(3)*flow(4)
     2              +backstre1(4)*flow(3))-term4*flow(3)*flow(4))
            stiff(10)=effG+flow(4)*flow(4)*effhrd
     1              -term3*(backstre1(4)*flow(4)
     2              -term4*flow(4)*flow(4))
            stiff(11)=flow(1)*flow(5)*effhrd
     1              -term3*(0.5d0*(backstre1(1)*flow(5)
     2              +backstre1(5)*flow(1))-term4*flow(1)*flow(5))
            stiff(12)=flow(2)*flow(5)*effhrd
     1              -term3*(0.5d0*(backstre1(2)*flow(5)
     2              +backstre1(5)*flow(2))-term4*flow(2)*flow(5))
            stiff(13)=flow(3)*flow(5)*effhrd
     1              -term3*(0.5d0*(backstre1(3)*flow(5)
     2              +backstre1(5)*flow(3))-term4*flow(3)*flow(5))
```

```
          stiff(14)=flow(4)*flow(5)*effhrd
     1             -term3*(0.5d0*(backstre1(4)*flow(5)
     2             +backstre1(5)*flow(4))-term4*flow(4)*flow(5))
          stiff(15)=effG+flow(5)*flow(5)*effhrd
     1             -term3*(backstre1(5)*flow(5)
     2             -term4*flow(5)*flow(5))
          stiff(16)=flow(1)*flow(6)*effhrd
     1             -term3*(0.5d0*(backstre1(1)*flow(6)
     2             +backstre1(6)*flow(1))-term4*flow(1)*flow(6))
          stiff(17)=flow(2)*flow(6)*effhrd
     1             -term3*(0.5d0*(backstre1(2)*flow(6)
     2             +backstre1(6)*flow(2))-term4*flow(2)*flow(6))
          stiff(18)=flow(3)*flow(6)*effhrd
     1             -term3*(0.5d0*(backstre1(3)*flow(6)
     2             +backstre1(6)*flow(3))-term4*flow(3)*flow(6))
          stiff(19)=flow(4)*flow(6)*effhrd
     1             -term3*(0.5d0*(backstre1(4)*flow(6)
     2             +backstre1(6)*flow(4))-term4*flow(4)*flow(6))
          stiff(20)=flow(5)*flow(6)*effhrd
     1             -term3*(0.5d0*(backstre1(5)*flow(6)
     2             +backstre1(6)*flow(5))-term4*flow(5)*flow(6))
          stiff(21)=effG+flow(6)*flow(6)*effhrd
     1             -term3*(backstre1(6)*flow(6)
     2             -term4*flow(6)*flow(6))

        end if
!
!       Calculate stress and update strains
!
        term1=threetwo*deqplas

        term5=C1*deqplas

        do k1=1,3
           backstre1(k1)=w1*(backstre1(k1)+term5*flow(k1))
           backstre(k1)=backstre1(k1)
           stre(k1)=backstre(k1)+Z(k1)/alfa+shydro
           eplas(k1)=eplas(k1)+term1*flow(k1)
        end do
        do k1=4,6
           backstre1(k1)=w1*(backstre1(k1)+term5*flow(k1))
           backstre(k1)=backstre1(k1)
           stre(k1)=backstre(k1)+Z(k1)/alfa
           eplas(k1)=eplas(k1)+term1*flow(k1)
        end do

      else
!
!     Calculate Material Jacobian for elastic case
!
        if(icmd.ne.3) then

           term7=lam+G2

           stiff(1)=term7
           stiff(2)=lam
```

```fortran
            stiff(3)=term7
            stiff(4)=lam
            stiff(5)=lam
            stiff(6)=term7
            stiff(7)=0.d0
            stiff(8)=0.d0
            stiff(9)=0.d0
            stiff(10)=G
            stiff(11)=0.d0
            stiff(12)=0.d0
            stiff(13)=0.d0
            stiff(14)=0.d0
            stiff(15)=G
            stiff(16)=0.d0
            stiff(17)=0.d0
            stiff(18)=0.d0
            stiff(19)=0.d0
            stiff(20)=0.d0
            stiff(21)=G

         endif

       endif
!
!      Store state variables
!
       do k1=1,6
          xstate(k1,iint,iel)=backstre(k1)
          xstate(k1+6,iint,iel)=backstre1(k1)
          xstate(k1+12,iint,iel)=eplas(k1)
       end do
       xstate(19,iint,iel)=eqplas
!
       return
       end

! --------------------------------------------------------------------
       subroutine calcterm(Q1,b1,kyield,R1,gamma1,C1,G3,stre,
     1 backstre1,eqplas,deqplas,shydro,syield,w1,alfa,Z,
     2 eqZ,dRdep,dJ2Zdep,dalfadep,FlowdotBstre1,term2,threetwo,
     3 kappa,chi1,K,n,dtime,phi1,phiinf1,omega1,dphi1dep,h)

       real*8 Q1,b1,kyield,R1,gamma1,C1,G3,stre(6),
     1   backstre1(6),eqplas,deqplas,shydro

       real*8 syield,w1,alfa,Z(6),eqZ,dRdep,dJ2Zdep,dalfadep,
     1   FlowdotBstre1,term2,threetwo,
     2   K,n,kappa,chi1,phi1,phiinf1,omega1,dphi1dep,dtime,h

       real*8 one, two, six

       parameter(one=1.d0, two=2.d0, six=6.d0)

       R1=Q1*(one-exp(-b1*eqplas))
       dRdep=Q1*b1*exp(-b1*eqplas)
       syield=kyield+R1
```

```fortran
      h=K*(deqplas/dtime)**(one/n)+syield

      phi1=phiinf1+(one-phiinf1)*exp(-omega1*eqplas)
      dphi1dep=-omega1*(one-phiinf1)*exp(-omega1*eqplas)
      chi1=phi1+dphi1dep*deqplas

      w1=(one+gamma1*phi1*deqplas)**(-one)
      alfa=one+(G3+w1*C1)*deqplas/h
!
!     Z stress
!
      do k1=1,3
          Z(k1)=stre(k1)-w1*backstre1(k1)-shydro
      end do
      do k1=4,6
          Z(k1)=stre(k1)-w1*backstre1(k1)
      end do
!
!     Z equivalent stress
!
      eqZ=(Z(1)-Z(2))**2+(Z(2)-Z(3))**2+(Z(3)-Z(1))**2
      do k1=4,6
          eqZ=eqZ+six*Z(k1)**2
      end do
      eqZ=sqrt(eqZ/two)
      !
      !   Derivatives
      !
      kappa=K/(n*dtime)*(deqplas/dtime)**((one-n)/n)+dRdep

      FlowdotBstre1=(Z(1)*backstre1(1)+Z(2)*backstre1(2)
     1          +Z(3)*backstre1(3)+two*(Z(4)*backstre1(4)
     2          +Z(5)*backstre1(5)+Z(6)*backstre1(6)))/eqZ
       dJ2Zdep=threetwo*FlowdotBstre1*gamma1*chi1*w1**two
       term2=w1*C1*(one-gamma1*w1*chi1*deqplas)
       dalfadep=(G3+term2+(one-alfa)*kappa)/h

      return
      end
```

References

1. Armstrong, P.J., Frederick, C.O.: A mathematical representation of the multiaxial Bauschinger effect. GEGB report RD/B/N731, Berkley Nuclear Laboratories (1966)
2. Benallal, A., Marquis, D.: Constitutive equations for nonproportional cyclic elasto-viscoplasticity. J. Eng. Mater. Technol. **109**, 159–164 (1987)
3. Chaboche, J.L., Rousselier, G.: On the plastic and viscoplastic constitutive equations - part I: rules developed with internal variable concept. J. Press. Vessel Technol. **105**, 153–158 (1983)
4. Chaboche, J.L., Rousselier, G.: On the plastic and viscoplastic constitutive equations - part II: application of internal variable concepts to the 316 stainless steel. J. Press. Vessel Technol. **105**, 159–164 (1983)
5. Chaboche, J.L.: Constitutive equations for cyclic plasticity and cyclic viscoplasticity. Int. J. Plast. **5**, 247–302 (1989)
6. Code Aster: Documentation 15, https://www.code-aster.org/V2/doc/default/en/index.php?man=R (2020). Accessed 26 April 2021
7. de Borst, R., Heeres, O.: Computational plasticity. In: Schrefler, B.A. (ed.) Environmental Geomechanics, Chapter 4, pp. 171–202. Springer, Vienna (2001)
8. de Borst, R., Heeres, O.: A unified approach to the implicit integration of standard, non-standard and viscous plasticity models. Int. J. Numer. Anal. Methods Geomech. **26**, 1059–1070 (2002)

9. Dhondt, G.: CalculiX CrunchiX USER'S MANUAL version 2.17. (2020)
10. Heeres, O.M., Suiker, A.S.J., de Borst, R.: A comparison between the Perzyna viscoplastic model and the Consistency viscoplastic model. Eur. J. Mech. A Solids **21**, 1–12 (2002)
11. Hibbit, B., Karlsson, B., Sorensen, P.: ABAQUS Theory Manual. Hibbit, Karlsson & Sorensen Inc., Pawtucket (2008)
12. Hopperstad, O.S., Remseth, S.: A return mapping algorithm for a class of cyclic plasticity models. Int. J. Numer. Methods Eng. **38**, 549–564 (1995)
13. Kobayashi, M., Ohno, N.: Implementation of cyclic plasticity models based on a general form of kinematic hardening. Int. J. Numer. Methods Eng. **53**, 2217–2238 (2002)
14. Kullig, E., Wippler, S.: Numerical integration and FEM-implementation of a viscoplastic Chaboche-model with static recovery. Comput. Mech. **38**, 491–503 (2006)
15. Marc 2008 r1, volume A: theory and user information. Santa Ana (2008)
16. Perzyna, P.: The study of the dynamic behavior of rate sensitive plastic materials. Arch. Mech. **1**, 113–129 (1963)
17. Prager, W.: A new method of analyzing stresses and strains in work hardening plastic solids. J. Appl. Mech. **23**, 795–810 (1956)
18. Scilab: Documentation, https://wiki.scilab.org/Documentation (2020). Accessed 11 May 2021
19. Simo, J.C., Taylor, R.L.: Consistent tangent operators for rate-independent elastoplasticity. Comput. Methods Appl. Mech. Eng. **48**, 101–118 (1985)
20. Suchocki, C.: On finite element implementation of cyclic elastoplasticity: theory, coding, and exemplary problems. Acta Mech. **233**, 83–120 (2022)
21. Voce, E.: The relationship between stress and strain for homogeneous deformation. J. Inst. Metals **74**, 537–562 (1948)
22. Wu, H.-Ch.: Continuum Mechanics and Plasticity. Chapman & Hall/CRC Press, New York (2005)
23. Yang, M., Feng, M.: Finite element implementation of non-unified visco-plasticity model considering static recovery. Mech. Time-Depend. Mater. **24**, 59–72 (2020)
24. Zienkiewicz, O.C., Cormeau, I.C.: Visco-plasticity solution by finite element process. Arch. Mech. **24**, 873–889 (1972)
25. Zienkiewicz, O.C., Cormeau, I.C.: Visco-plasticity - plasticity and creep in elastic solids—a unified numerical solution approach. Int. J. Numer. Methods Eng. **8**, 821–845 (1974)