CZECH TECHNICAL UNIVERSITY IN PRAGUE

DOCTORAL THESIS

Numerical Implementation of Distortional Hardening Models

Author:
René MAREK

Supervisor:
Jiří PLEŠEK

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Mechanics of Solids, Deformable Bodies and Continua

January 30, 2018

Declaration of Authorship

I, René MAREK, declare that this thesis titled, "Numerical Implementation of Distortional Hardening Models" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Czech Technical University in Prague

Abstract

Faculty of Mechanical Engineering
Department of Mechanics, Biomechanics and Mechatronics

Doctor of Philosophy

Numerical Implementation of Distortional Hardening Models

by René MAREK

The presented thesis describes the formulation of two new constitutive models of metal plasticity within the realm of small strains, both featuring directional distortional hardening. New yield functions have been proposed with distortion and cross-effect control. The models feature new kinematic hardening rule that builds on the premise of multisurface plasticity while keeping the amount of internal variables low. The models' intended use is the study of multiaxial ratcheting–the cumulation of plastic deformation in combined loading, and ways of its calibration. Their possible real application lies in cyclic multiaxial operational loading present in pressurized piping systems and in some very specific applications. The models have been implemented into a UMAT subroutine for FE-system Abaqus.

Acknowledgements

First of all, I wish to express my sincere gratitude towards my supervisor Jiří Plešek for his continual motivation and endless support throughout my research. I will always remember his many demonstrations of rigorous attitude to scientific work as well as extraordinary sense of humor.

My sincere thanks go to my co-supervisor prof. Yannis Dafalias, who guided me during my research into constitutive modeling. I also thank his ex-student, now Professor Heidi Feigenbaum, for all the discussions we had and for her welcoming reception at NAU, Flagstaff.

I thank my closest colleagues: Slávek, Zbyněk, Honza, Honza, Honza and most of all, Dušan, for creating an absolutely wonderful work environment.

This work was supported by the European Regional Development Fund under Grant No. CZ.02.1.01/0.0/0.0/15_003/0000493 Centre of Excellence for Nonlinear Dynamic Behaviour of Advanced Materials in Engineering and by the Ministry of Education, Youth and Sports KONTAKT II LH14018.

Contents

List of Figures

List of Tables

List of Symbols

Symbol	Unit	Description
a_1	$[\mathrm{MPa}]$	model parameter: speed of kin. hardening
a_2	$[\mathrm{MPa}^{-1}]$	model parameter: limit of kin. hardening
A_1	$[\mathrm{MPa}^{-3}]$	model parameter: speed of distortion evolution
A_2	$[\mathrm{MPa}^{2}]$	model parameter: limit of distortion
b_1	$[\mathrm{MPa}]$	model parameter: speed of B. Surface movement
b_2	$[\mathrm{MPa}^{-1}]$	model parameter: limit of B. Surface movement
c_0	$[\mathrm{MPa}^{-2}]\,,[-]$	model parameter: initial directional distortion
c_1	$[\mathrm{MPa}^{-2}]\,,[-]$	model parameter: rate of distortion evolution
c_2	$[\mathrm{MPa}^{-2}]\,,[-]$	model parameter: limit of directional distortion
c	$[\mathrm{MPa}^{-2}]\,,[-]$	model parameter or variable: directional distortion
$\mathbf{C}$	$[\mathrm{MPa}]$	elastic stiffness matrix
E	$[\mathrm{MPa}]$	elastic modulus
f	$[\mathrm{MPa}]$	yield function
F_i	$[\mathrm{MPa}]$	internal variable: flow of forces between backstress components
G_4	$[\mathrm{MPa}]$	internal variable: growth of new slip systems
h_1	$[\mathrm{MPa}]$	model parameter: initial speed of kin. hardening
h_2	$[\mathrm{MPa}]$	model parameter: nominal speed of kin. hardening
h_3	$[\mathrm{MPa}]$	model parameter: component depletion speed
h_4	$[-]$	model parameter: multiaxial flow distribution
h_5	$[-]$	model parameter: multiaxial flow distribution
$h_\mathrm{a}, h_\mathrm{b}, h_\mathrm{c}$	$[-]$	dependent parameters: multiaxial flow distribution
k_0	$[\mathrm{MPa}]$	model parameter: initial yield limit
k	$[\mathrm{MPa}]$	model variable: nominal yield surface size
K	$[\mathrm{MPa}]$	model variable: Bounding Surface size
K_1	$[\mathrm{MPa}]$	model parameter: B. Surface size evolution speed
K_2	$[\mathrm{MPa}^{-1}]$	model parameter: B. Surface size evolution limit
K_p	$[\mathrm{MPa}]$	plastic modulus
$\mathbf{n}$	$[-]$	Yield Surface outer normal
$\mathbf{n}_\mathrm{r}$	$[-]$	normalized radial tensor
$\mathbf{n}_\mathrm{z}$	$[-]$	normalized tensor $\mathbf{z}$

$\mathbf{s}$	$[\mathrm{MPa}]$	deviatoric (traceless) stress tensor
S_i	$[\mathrm{MPa}]$	Size of the leading edge of i-th band of multisurface equivalent
t	$[-]$	unitless time, integration step identification
$\mathbf{z}$	$[\mathrm{MPa}]\,,[-]$	Yield Surface axis, traceless second-order tensor
$\boldsymbol{\alpha}$	$[\mathrm{MPa}]$	backstress tensor
$\boldsymbol{\alpha}_i$	$[\mathrm{MPa}]$	i-th backstress component
$\boldsymbol{\alpha}^{\mathrm{B}}$	$[\mathrm{MPa}]$	projection of backstress tensor on the B. Surface
$\boldsymbol{\beta}$	$[\mathrm{MPa}]$	model variable: Bounding Surface center
δ	$[\mathrm{MPa}]$	internal measure: distance to projection on Bounding S.
ε	$[\mathrm{MPa}]\,,[-]$	model parameter: smoothness of elastopl. trans.
ϵ_{cum}	$[-]$	cumulative plastic strain
ϵ_{plat}	$[-]$	model parameter: plateau depletion strain
$\boldsymbol{\epsilon}$	$[-]$	strain tensor
κ_1	$[\mathrm{MPa}]$	model parameter: speed of isotropic hardening
κ_2	$[\mathrm{MPa}]$	model parameter: limit of isotropic hardening
κ_3	$[\mathrm{MPa}]$	model parameter: speed of isotropic hardening
κ_4	$[\mathrm{MPa}]$	model parameter: limit of isotropic hardening
λ	$[-]$	plastic multiplier
$\boldsymbol{\sigma}$	$[\mathrm{MPa}]$	stress tensor
ν	$[-]$	Poisson's ratio

Chapter 1

Introduction

Refinement of constitutive plasticity models is a continuing trend. Its aim is to increase accuracy and credibility of their predictions. This brings a higher chance of success when optimizing the distribution of operational stresses for the purpose of lowering fatigue, increasing safety or guaranteeing proper function by predicting deformation during operation.

After a successful attempt to simulate the hardening curve of material by the use of kinematic and isotropic hardening, applicable even for complex uniaxial cyclic loadings, the scientific community started to focus on multiaxial loading sequences and a possible influence of directional distortional hardening, which addresses distortion of the elastic region in the stress space. The ongoing research into multiaxial ratcheting simulation only recently explored the use of such models. A great number of authors utilized a large set of experiments assembled in [13],[12] and [2]. However, these and other published experiments do not explore the shape of the distortion. The use of distorted hardening models on these data, such as [8] or [35], lacks an important piece of the information needed to adapt the kinematic rules and follow the real position of the yield surface. Better results are achieved in [10], having both types of experiments available. Such research is an iterative process of simulation, to proposition of experiments, to improvement of simulations. The experimental campaign needs to be interconnected with the modeling team and can hardly be assigned to a third party.

Fortunately, experimental facilities are becoming available to us on several institutions in Czech Republic, USA and Taiwan. Experimental measurement to such detail necessary for these models is very complex and often employs dynamic control of the machine displacement according to real-time data analysis. To successfully prepare a trajectory for the experiment, a reliable simulation is consulted. Increased precision of measurement is reachable in such trajectories, where the material's response is most pronounced, e.g. attacking the yield threshold perpendicularly. Therefore, a powerful model is needed as a basis for the design of experiments and for validation of new features and evolution rules. Recently, new ideas came to

mind, how to speedup calibration using just one or two specimens. A complex loading pattern could target specific parameters of the model to get the best initial calibration before numerical refinement is performed.

This thesis focuses on modeling in small deformations, where the application of such models is more sensible. Although the distortion of the elastic region is clearly measurable, its influence on the behavior of structures is generally minimal. In proportional loading in stress space, distortion of the elastic region is rotationally symmetrical about the loading direction and is manifested by an increase in the curvature of the frontal apex and flattening of the rear. To expose this phenomenon, multiaxial cyclic behavior has to be studied. In real application, that includes the operation of pressure systems and pipelines, where pressure and temperature loads combine. Other uses may include circularity violation, also called ovalization, of large diameter pipes from various loadings at seabed, mounting of heavy machinery during earthquake and other limit loadings, coiled springs, etc.

In this thesis, two new constitutive models are suggested, the first being the stepping stone for the second one. Sufficient convexity conditions of the yield surface have been derived for both models. Both models have been implemented in Abaqus and subjected to a multitude of tests concerning energy dissipation, numerical error and others. With a well-controlled distortion and appropriate kinematic hardening rules, the effect of distortion is quantified on selected multiaxial ratcheting experiments.

Although these models may find their practical application in their presented form, it is not intended. The theory of multiaxial ratcheting is still young. A very large set of data has to be obtained experimentally for such a model to be finalized and to give reliable predictions. To initiate an experimental campaign consisting of up to a hundred specimens is a strategic decision for the very near future.

Chapter 2

State of the Art

This work focuses on design and implementation of an advanced model of metal plasticity. It will become clear that accurate phenomenological modeling of recognized behavioral traits of metals collides with demands for hardware capacity and calibration complexity. The following section discusses some known characteristics of metals relevant to this work as well as its use in future applications. A deeper look into the inner workings of the material is necessary as a solid foundation for model development.

2.1 Selected observed behavior in metal plasticity

Due to intuition and a bit of luck, a very convenient experiment has been conducted for an in-depth analysis and a model proposition in [3] and [4], introducing what has become known as Bounding Surface (BS) plasticity theory. A Grade 60 steel specimen underwent a short series of random cycles in strain-driven uniaxial stress test. The plotted results in Fig. 2.1 show a number of recognizable phenomena. A section of constant stress response called "plateau" is generated during the first stage of loading, when the density of dislocations rises. During this stage, uneven plastic straining may occur in certain geometries, where the unchanging stress does not force the material to behave uniformly. Basic models of perfect plasticity are effectively used on simple cases with construction and low-carbon steels. At the end of the plateau region, intergranular cavities are depleted and the density of dislocations is so high, that stacking of their individual effects occurs. This causes hardening. During plateau, despite the constant stress response in uniaxial loading, the material undergoes observable changes. As the density of dislocations rises, the material is less resilient and shows a rapid decrease of yield strength in opposite direction to preloading - the so-called Bauschinger effect. The yield surface is shrinking and its center is moving towards the direction of loading. This phenomenon can be so pronounced, that the elastic region no longer encompasses the origin of the stress space. In that case, a visible yielding occurs when unloading, while the

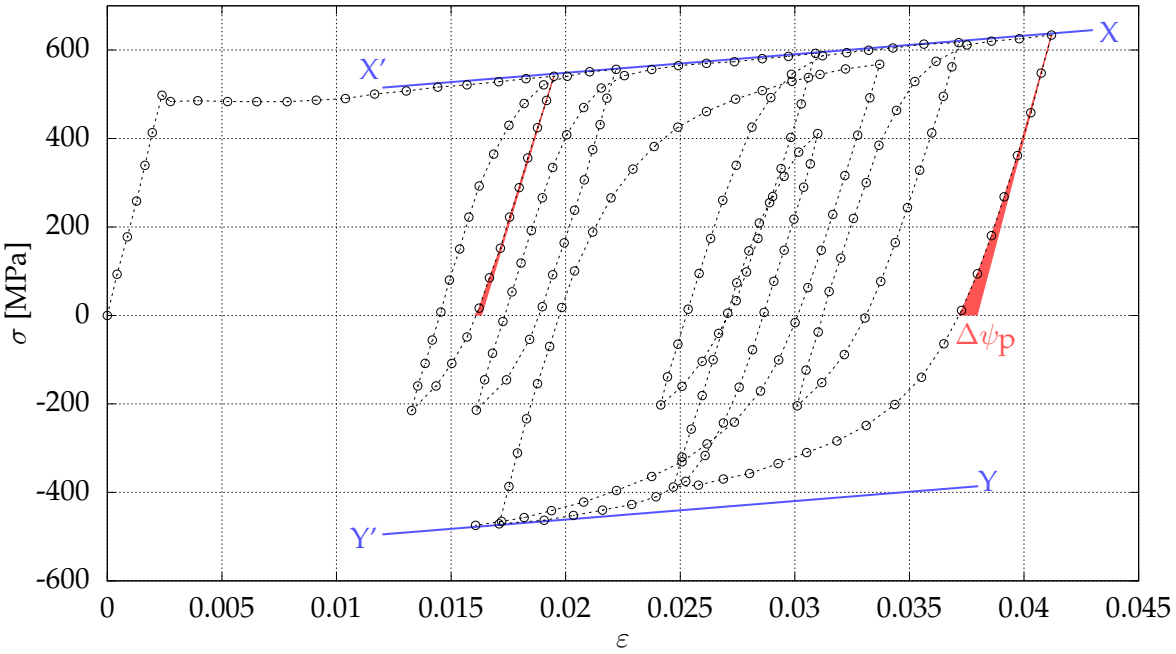

FIGURE 2.1: Random uniaxial cyclic loading after [4]

stress is still aiming in the original direction, before unloaded state is even reached. Because the direction of plastic flow is opposite to the now decreasing stress, the dissipative plastic flow has to be propelled by stored internal energy. Supplementary to the elastic free energy, this part represents a change in the plastic part of the Helmholtz free energy ψ_{p}, see Figure 2.1. Its value is loosely connected to the backstress. However, as the definition of backstress is the position of a referential point (center) inside the yield surface, its size depends on the definition of yield. The translation of the elastic region in stress space is called kinematic hardening and has been studied and simulated for many years.

To help estimate the amount of free energy stored in the material, several assumptions may be used. Distribution of plastic free energy among internal variables is fictitious and arbitrary, although influenced by the material behavior. We can choose a very fine definition of yield, rendering the yield surface small following a rapid isotropic softening. In that case, as the yield surface shrinks, some energy is even transferred from isotropic to kinematic hardening, which is now dominant form of energy deposition. For a course definition of yield, the backstress evolution is much less intensive. Although it is always possible to measure the backstress by reverse loading, the value it reaches, taking to account our definition of yield, does not tell us the amount of energy stored within it. For that, either the kinematic hardening rule or the recent loading history has to be evaluated. An initial part of a fast kinematic hardening with smooth elastic-plastic transition does not store

much energy, as the plastic strain is still low. Later, when backstress draws near its asymptotic value, the storing of energy approaches zero as well.

Slow effect of cyclic hardening is visible as a growing saturation limit of first four individual loops in Figure 2.1. This occurs at the cost of reducing tensile strength in the opposite direction, which is ultimately regained at the end of the loading sequence. Upper and lower bounds denoted as X'X and Y'Y respectively, are linear within the limits of this experiment. For larger deformations, the amount of plastic energy stored in the material rises and reaches a material limit. The position of bounds influences the vertical position of loops. In stress-driven cyclic loading this position influences the amount of plastic strain on the forward and returning trajectory. Their differences represent the strain shift within one loop, or in other words, the speed of ratcheting. Within the realm of small deformations and for a given cyclic loading, this speed is changing into a stabilized value.

2.1.1 Plastic flow and its effects under non-proportional loading

In metal plasticity for non-porous materials, the associative flow rule is used. This means the plastic strain rate (plastic flow) does not have a volumetric part and its direction is normal to the yield surface as

$$\dot{\boldsymbol{\varepsilon}}_{\mathrm{p}} = \dot{\lambda}\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}. \tag{2.1}$$

During plastic loading, the direction of plastic flow may greatly differ from that of stress. This is a transient state, while all tensor variables are aligning to the direction of loading. The rate of hardening is defined from plastic modulus K_{p} according to

$$\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} = K_{\mathrm{p}}\dot{\lambda}, \tag{2.2}$$

where λ is the plastic multiplier. The plastic modulus will be defined later from the evolution rules of the internal variables.

Projecting the process of plastic loading into a 1D representation, the superposition of strains is as follows:

$$\mathrm{d}\varepsilon^{\mathrm{t}} = \mathrm{d}\varepsilon^{\mathrm{e}} + \mathrm{d}\varepsilon^{\mathrm{p}} \tag{2.3}$$

or expanded as

$$\frac{\mathrm{d}\sigma}{E^{\mathrm{t}}} = \frac{\mathrm{d}\sigma}{E^{\mathrm{e}}} + \frac{\mathrm{d}\sigma}{E^{\mathrm{p}}} \tag{2.4}$$

for total, elastic and plastic moduli. In this representation, a smooth transition from elastic to plastic behavior, seen on many points in Fig. 2.1, is achieved by gradual reduction of initially infinite E^{p}. Similarly, the plastic modulus K_{p} undergoes the same change.

The most complete picture of the material behavior linked to plastic flow and distortion of the elastic region is the equi-strain map. A very detailed work in [39] is inspirational, however the distortion measured is not very pronounced. A very sharp yield surface has been detected by [42], however, the work by [27] features more data points, see Fig. 2.2. It was generated by measuring 12 identical specimens made of mild steel. Emphasis was put to reduce any differences between specimens–they were manufactured from a single ingot in the same direction, and the size of grains was monitored.

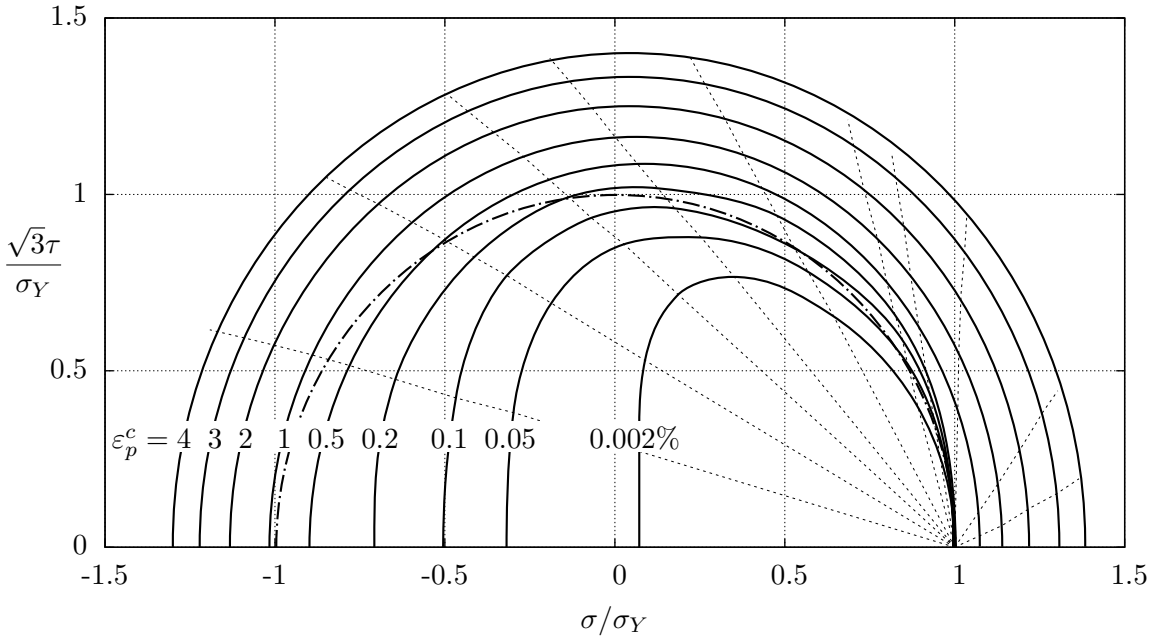

FIGURE 2.2: Equi-strain map after [27]

Looking at this particular case, the plastic modulus K_p reaches at its first measured position at the cummulative strain $\varepsilon_\mathrm{p}^\mathrm{c} = 0.002\%$ (20 microstrains) a value of about 300 GPa.

2.1.2 Distortion of the elastic region

If we assume the plastic process to be negligible inside the yield surface, a different method of examining the shape of the elastic region may be used. A dynamic control of the testing machine enables tracing the yield surface within a single specimen by touching the yield threshold without significant disruption of its state. Many works have been published applying this method on tubular specimens, either loaded by tension and torsion, or by tension and inner pressure. Notably Wu and Yeh used 5 microstrain deviation from elastic response as the yield definition for 304 Stainless steel [43]. Tracing with such a small, fine definition of yield is highly demanding

on methodology and prone to systematic error. A comprehensive study in [15], [16] and [17] uses a 10 microstrain threshold on Aluminium with loading trajectories reaching up to 16% strain. Apart from the effects of such a large deformation, the distortion is fully formed. Following combined loading trajectories, the main axis of distortion does not go through the origin of the stress space anymore. Great insight into the evolution of distortion came from Phillips and Tang in [30] and [31]. The examined loading sequences were all within the realm of small strains, so the distortion features distinct footprint of its preloading history, see Fig. 2.3.

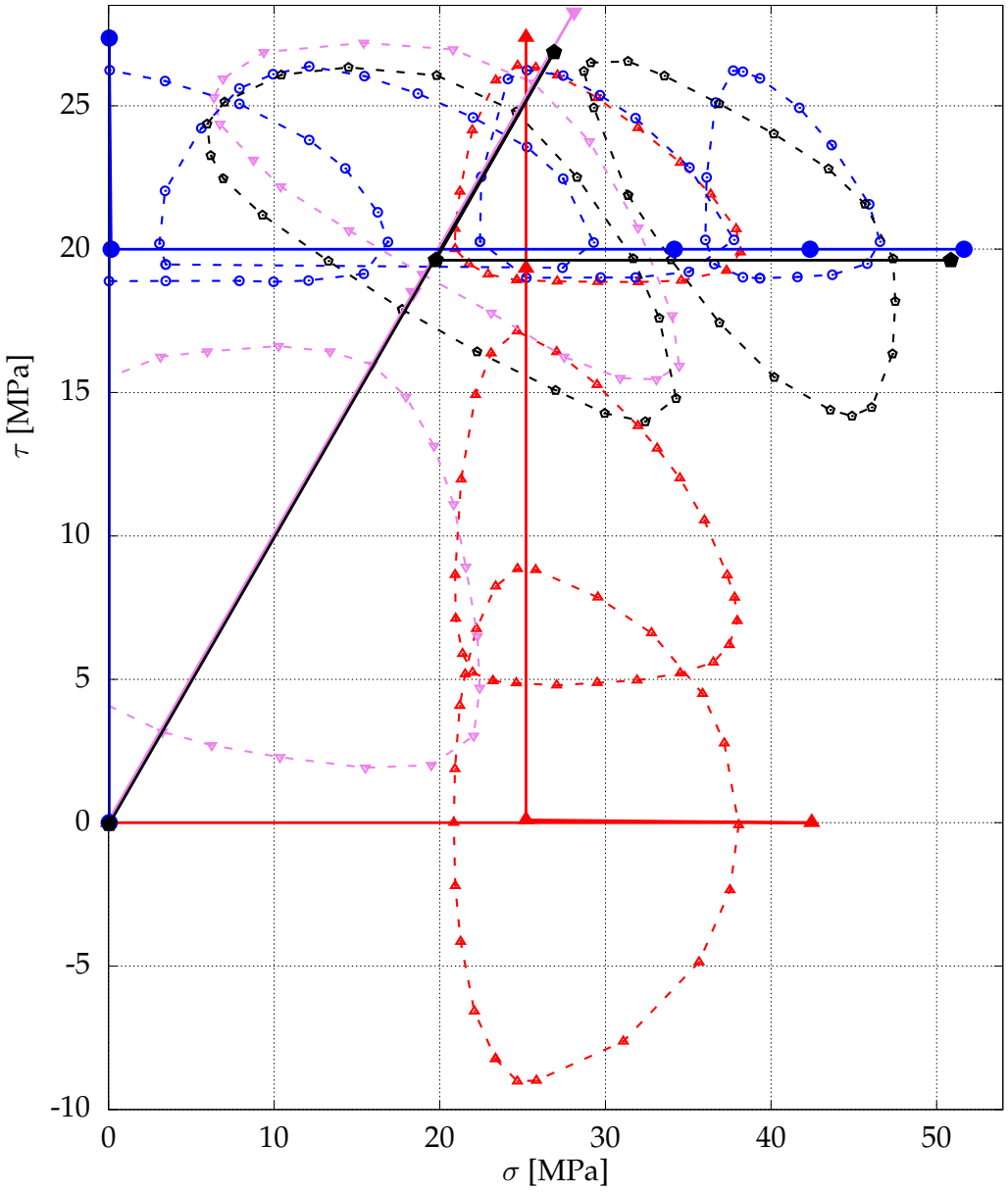

FIGURE 2.3: Selected datapoints with loading trajectories after [30] for $T = 294\text{K}$

Four loading histories in the stress space are plotted by full lines. Note, that the initial prestress point for each of the traced yield surface lies outside of the measured elastic region. This is due to very high plastic modulus on the returning trajectory. The first registered yield point was in the unloading trajectory, which changed the position of the yield surface.

Although numerically inconvenient, the yield surface does not need to be convex. That rule originally arose from Prager's surpassed condition of irreversibility [34], stating that plastic work done should always be positive. As the definition of yield represents our acceptance of residual straining within the elastic region, possible convexity violations can be measured due to the same assumption, see [24].

2.1.3 Dissipation inequality

A convenient way to satisfy the second law of thermodynamics of a general constitutive model is described in [6]. It is assumed, that the Helmholtz free energy in the material can be divided into elastic and plastic part as

$$\psi = \psi_{\mathrm{e}} + \psi_{\mathrm{p}}. \tag{2.5}$$

The authors proposed a series of positive and negative functions for each individual internal variable to represent its contribution to the plastic part of free energy. With the assumption of isothermal process, the simplified Clausius-Duhem inequality yields

$$\boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} - \rho \dot{\psi}_{\mathrm{p}} \geq 0. \tag{2.6}$$

Expressing the distributed plastic part of free energy led to a series of conditions for the isotropic, kinematic and distortional rules and their limits.

For a very complex model however, this approach may not be feasible, as there may not be an analytical solution for each of the individual conditions. Additionally, even the assumptions for energy storage and distribution may be difficult to propose. Therefore it remains just as a point of interest to search for instances of general loading where free energy is released.

2.2 Notable models with advanced kinematic hardening

A thorough analysis of uniaxial and multiaxial ratcheting on internally pressurized specimens made of mild steel has been presented by Corona, Hassan and Kyriakides in [13, 12] and [2]. Various kinematic hardening rules were employed focusing on

simpler schemes to see whether any of them could be used for large-scale simulations. Greatest differences in predictions were met when combining different types of tests on single calibration. Varying the amplitudes and internal pressure for the same type of test had a much lesser impact.

As a basis for a reliable universal model, excellent uniaxial ratcheting predictions need to be achieved. A strong contender is the SANISTEEL model in [25] using the Bounding Surface framework. An initially fixed non-hardening BS induces a perfectly plastic response (plateau), while the kinematic and isotropic parts evolve. This can be demonstrated by unloading–the material will yield before nominal yield strength in opposite direction is reached. The yield surface moves within the Bounding Surface. After depleting the plateau threshold, the Bounding Surface is unlocked and its movement now allows slow reach of higher levels of stress. The authors studied three simple rules for Bounding Surface kinematic hardening, differing in the way the BS moves when reloading and whether it has any asymptotic limits of position. These rules have different effects on ratcheting speed. Ultimately, the decision for a particular rule, or a proposal of a new one, will be made after observing a greater amount of experimental data. The choice of such kinematic rule for the intended model is completely free, having no effect on the yield function and its derivatives. Limits have to be kept for the position of the Bounding Surface, as it rapidly increases the amount of storable plastic free energy.

A multisurface kinematic hardening rule offers even better predictions for complex loading trajectories, as it imitates our understanding of the inner workings of the material. Each surface represents a subset of slip systems, whose activation require a particular state of stress. An interesting observation were made by Zhang, Benítez and Montáns in [44]. A reproduction of yield surface tracing procedure on a model featuring multisurface kinematic rule with smooth elastic-plastic transition showed a noticeable conformity with experimental data.

A deep study of equi-strain maps on distorted and non-distorted models will ultimately answer, whether the concept of distorted yield surface is a right way to proceed, or whether just the distortion of kinematic rule, planned or induced, is more logical. A yield surface with very precisely defined yield threshold loses the notion of shape, as the plastic modulus approaches infinity.

2.3 Models of directional distortional hardening

A number of different ways have been described throughout the literature on how to distort the shape of the yield function. The straight-forward approach presented in [28] is to vary the radius of the von Mises hypersphere. In this case, it is done by

a series of harmonic functions according to the position angle $\arccos(\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z})$, where $\mathbf{n}_\mathrm{z}$ is the unit-norm director tensor defining the orientation of the distortion and $\mathbf{n}_\mathrm{r}$ is the unit-norm tensor along the "radial" tensor $(\mathbf{s} - \boldsymbol{\alpha})$. Its definition will therefore be

$$\mathbf{n}_\mathrm{r} \equiv \frac{\mathbf{s} - \boldsymbol{\alpha}}{\|\mathbf{s} - \boldsymbol{\alpha}\|}, \tag{2.7}$$

where the tensor norm is defined as

$$\|\mathbf{x}\| \equiv \sqrt{\mathbf{x} : \mathbf{x}} = \sqrt{x_{ij} x_{ij}}. \tag{2.8}$$

In the original work, $\mathbf{n}_\mathrm{z}$ is the unit-norm tensor simply in the direction of backstress.

Two or more harmonic components are needed, as the desired function for fully formed distortion is quite complex. Adjustment for cross effect is non-trivial and a high price for trigonometric computations does not make this a good choice.

Another straight forward idea is to use a polynomial function. A 3rd degree polynomial is presented in [22]. It offers a multitude of affine transformations, however the shape of distortion is not pronounced enough and may ultimately need a higher degree of polynomials.

Extensive study by Kurtyka and Życzkowski in [18],[19] demonstrated remarkable agreement of their model with experimental data. Distortion was modeled by geometrical description in Ilyushin's auxiliary stress space. Intact proportional characteristics, explicit independent control of the transverse dimension (the cross effect), independent orientation of the elastic region, and a possibility of non-symmetric distortion are its most valuable features. This yield surface could be used as a basis for a new model. However, the handling of the yield function is difficult and the possibility of non-symmetric distortion does not really offer the capabilities needed.

A very promising approach by Shutov and Ihleman in [36] and [37] is represented by a rotationally symmetrical yield surface generated by a meridian assembled from arc sections. For intermediate distortion, a weighted average between initial von Mises hypersphere and a fully distorted shape was chosen. This model has a very simple gradient equation. If the intermediate shape were also described by arc sections, the yield function would have had a single step solution for radial corrector method, as well as for the plastic part detection of the trial stress. In other models, both processes require multiple calculations of the yield function and its gradient. However, calculation of the arc section positions for different intensity of distortion as well as cross effect would be quite difficult. By its definition, there are no possible expansions towards non-symmetrical distortion.

The most promising approach uses affine transformations of the radial stress $\mathbf{s} - \boldsymbol{\alpha}$. The work by François [9] proposes a simple distortion function which is

quadratic in all transverse directions to the main axis of the yield surface. This function leaves the proportional characteristics unaltered, making it easily applicable to mixed hardening models that have already been in use. The magnitude of distortion may be controlled for a particular class of materials, simplifying the identification of a specific batch of production materials to solely proportional calibration. The distortion is oriented via the backstress tensor, but any other convenient director can be employed. It should be noted that the plastic function becomes non-convex outside the elastic region, which has to be properly treated by numerical implementation. This form of yield function lacks in curvature of the frontal apex and does not feature cross-effect control.

2.3.1 Recent models by Feigenbaum and Dafalias

The original model that caught the interest of our team was proposed in [6]. Their model, motivated by the work of Voyiadjis and Foroozesh [40], employs a fourth-order tensor as an evolving internal variable. Its yield function

$$\mathrm{f}(\boldsymbol{\sigma}) = (\mathbf{s} - \boldsymbol{\alpha}) : \left[\tfrac{3}{2}\mathbf{I}^{\mathrm{D}} + (\mathbf{n}_{\mathrm{r}} : \boldsymbol{\alpha})\,\mathbf{A}\right](\mathbf{s} - \boldsymbol{\alpha}) - k^2 \tag{2.9}$$

forms distortion by applying a transformation via an oriented elongation on its radial tensor, see Fig. 2.4. In this thesis, the original denomination $\mathbf{n}_{\mathrm{r}}$ used in [6] and [7] will be used for new models as well.

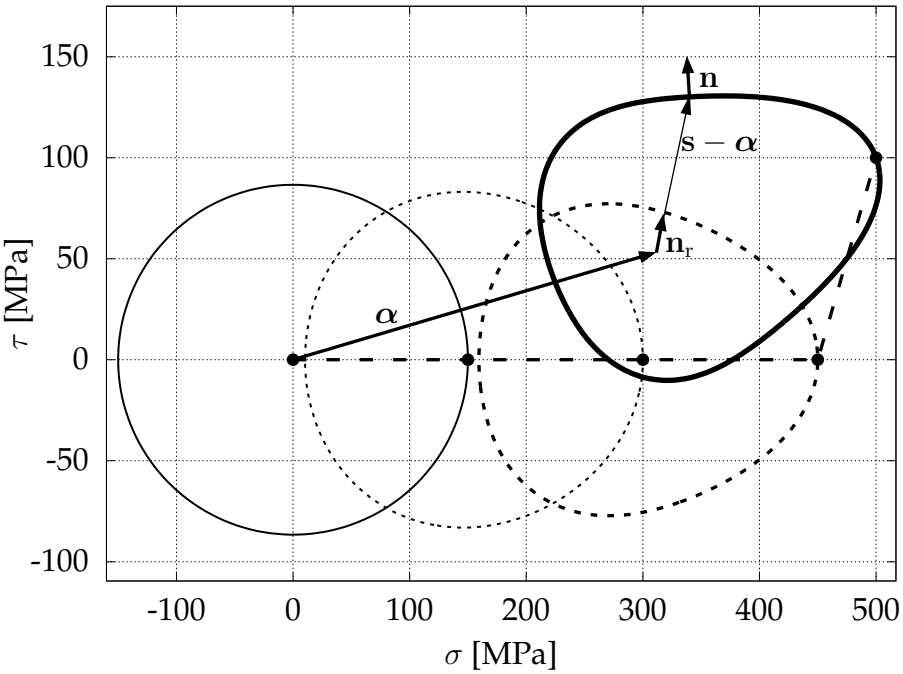

FIGURE 2.4: **A**-model - distortion and its orientation in stress space

The evolution law of the 4th-order tensor

$$\dot{\mathbf{A}} = -\lambda c_1 \left\|\mathbf{s}-\boldsymbol{\alpha}\right\|^2 \left[\left(\mathbf{n}_\mathrm{r} : \boldsymbol{\alpha}\right) \mathbf{n}_\mathrm{r} \otimes \mathbf{n}_\mathrm{r} + \tfrac{3}{2} c_2 \mathbf{A}\right] \tag{2.10}$$

provides an inside look to the principle. The distortion grows in various intensity in the direction of the radial stress, while a recurrent term $\frac{3}{2}c_2\mathbf{A}$ bounds its evolution. Tensor $\mathbf{A}$ becomes a continuous summation of increments in the direction of $\mathbf{n}_\mathrm{r}^t \otimes \mathbf{n}_\mathrm{r}^t$ as well as recurrent terms, representing various stretching of the applied tensor. A function of angular position $(\mathbf{n}_\mathrm{r} : \boldsymbol{\alpha})$ is then able to control this effect of stretching with respect to backstress. In effect, this causes the prolonged stretching or sharpening of the frontal apex and contraction or flattening of the rear. Generally, the yield surface may become non-symmetric with respect to the direction of backstress, as no eigentensors of $\mathbf{A}$ are collinear with backstress. The fact that the distortion is proportional to backstress means the evolution of $\mathbf{A}$ may be very rapid without much effect on the uniaxial behavior of the model. That may be the cause of high demands on computational power that were detected when implemented in [14]. No further investigation has been conducted. As the evolution rule of $\mathbf{A}$ is very close to that of backstress, the asymmetry is negligent and the model could be effectively simplified as $\mathbf{A} = \mathbf{z} \otimes \mathbf{z}$, having a much simpler evolution rule for the now fully independent main axis of yield surface defined by $\mathbf{z}$ while the yield function changed into

$$\mathrm{f}(\boldsymbol{\sigma}) = (\mathbf{s}-\boldsymbol{\alpha}) : \left[\tfrac{3}{2}\mathbf{I}^\mathrm{D} + (\mathbf{n}_\mathrm{r} : \mathbf{z})\, \mathbf{z} \otimes \mathbf{z}\right] (\mathbf{s}-\boldsymbol{\alpha}) - k^2 \tag{2.11}$$

The originally published α-model in [7] represents an even simpler modification of this model. It features an elongating distortion, as depicted in Figure 2.5. During the development of calibration procedures and implementation in [A2], the simplest model's shortcomings, a price to pay for simplicity, appeared in full light. Its yield function

$$\mathrm{f}(\boldsymbol{\sigma}) = \tfrac{3}{2}\left[1 - c\left(\mathbf{n}_\mathrm{r} : \boldsymbol{\alpha}\right)\right] (\mathbf{s}-\boldsymbol{\alpha}) : (\mathbf{s}-\boldsymbol{\alpha}) - k^2 \tag{2.12}$$

forms the distortion by altering the radial tensor rather than controlling the size of k. That is understandable, as the observed shapes of distortion require a complex function to do so. The multiplying nature of the distortion in Eq. (2.12) creates a highly nonlinear behavior during hardening, ultimately requiring limitation based on thermodynamic and convexity conditions. The shape of the distortion is limited to a form of elongation in the direction of backstress. Unlike experimental evidence, any visible differences in curvature on the frontal apex and back are present in high distortion/elongation only. There have been only very few experiments in metal

plasticity that would indicate such a state of material.

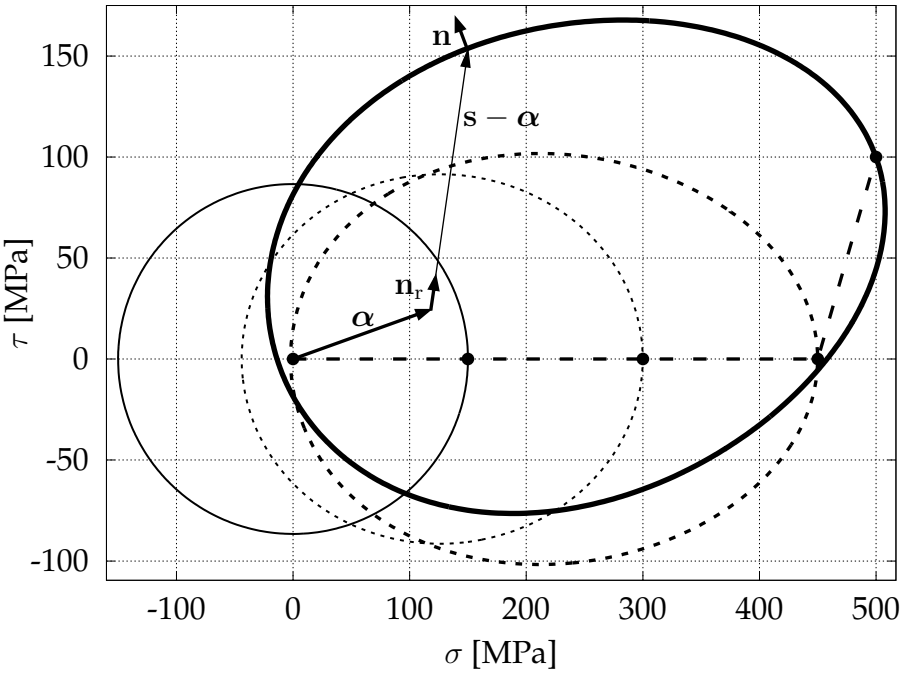

FIGURE 2.5: α-model - shape of distortion and its orientation in stress space

The crosswise dimension of the yield surface cannot be changed and it is predetermined by the isotropic yield limit with a small alteration during distortion. Affine transformation of the radial stress (axial contraction) before distortion is applied showed a way to cure these problems and laid the foundations for the new model.

2.4 Consistency condition

During plastic loading, the stress point remains on the yield surface ensuring

$$\mathrm{f}(\boldsymbol{\sigma}, \mathbf{x}_i, y_j) = 0. \tag{2.13}$$

By differentiating in respect to all internal evolving variables, a consistency equation emerges as

$$\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} + \sum_{i=1}^{N} \frac{\partial \mathrm{f}}{\partial \mathbf{x}_i} : \dot{\mathbf{x}}_i + \sum_{j=1}^{M} \frac{\partial \mathrm{f}}{\partial y_j} : \dot{y}_j = 0 \tag{2.14}$$

with the internal variables, tensors $\mathbf{x}_i$ and scalars y_j, obeying the evolution rules

$$\mathbf{x}_i = \dot{\lambda} \bar{\mathbf{x}}_i \tag{2.15}$$

$$y_j = \dot{\lambda} \bar{y}_j, \tag{2.16}$$

and where the stress response in differential form yields, accounting for Eq. (2.1):

$$\dot{\boldsymbol{\sigma}} = \mathbf{C}\dot{\boldsymbol{\epsilon}} - \dot{\lambda}\mathbf{C}\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}. \tag{2.17}$$

Recalling Eq. (2.2), the plastic modulus can be defined from Eq. (2.14) in combination with the aforementioned rate equations for the internal variables as

$$K_{\mathrm{p}} = -\sum_{i=1}^{N} \frac{\partial \mathrm{f}}{\partial \mathbf{x}_i} : \bar{\mathbf{x}}_i - \sum_{j=1}^{M} \frac{\partial \mathrm{f}}{\partial y_j} \bar{y}_j. \tag{2.18}$$

Combined with Eqs. (2.2) and (2.17), the desired equation for the plastic multiplier yields

$$\dot{\lambda} = \frac{\dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \mathbf{C}\dot{\boldsymbol{\epsilon}}}{K_{\mathrm{p}} + \dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \mathbf{C}\dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}}. \tag{2.19}$$

If needed, by differentiating Eq. (2.17) the continuum tangent stiffness can be arrived at as

$$\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}} = \mathbf{C} - \frac{\mathbf{C}\dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} \otimes \mathbf{C}\dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}}{K_{\mathrm{p}} + \dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \mathbf{C}\dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}}. \tag{2.20}$$

Chapter 3

Aims of the Thesis

The study of multiaxial ratcheting is the natural path of our research into constitutive modeling. A powerful model serves as a showcase of possibilities when tailoring a constitutive model for specific application and research. The initial strategy is to slowly open a new degree of freedom in the loading characteristics. The simulation of a slow transverse drift from the uniaxial cyclic behavior requires control of the curvature of front and back of the yield surface. A Bounding Surface framework is the ideal way to deal with plateau behavior of mild steels and with control of the asymptotic speed of ratcheting. Projection of the backstress flow tensor on the Bounding Surface opens up possibilities for tweaking the kinematic rule to fit experimental data. The interaction between the Bounding Surface and the yield surface is unclear. The Bounding Surface is assumed to be non-distorted. To help answer these questions and proceed with our research, a powerful model is needed:

1. Assembling a new model featuring directional distortional hardening.
    (a) Proposal of a new yield function with controlled transverse size.
    (b) Derivation of convexity condition.
    (c) Proposal of kinematic laws and bounding surface interaction rules.

To enable interlaboratory applications on simulations of whole specimens, a transferable model implementation into a FE-system is needed. Specimens subjected to internal pressure may require refining the calibration procedure due to the gradient of radial stress within its walls. A stable implementation opens our work to the public.

2. Implementation into a finite element code.
    (a) Euler forward scheme. Assessment of other methods.
    (b) Stability tests. Analysis of possibilities for optimization.

The calibration procedure consists of initial estimate of internal parameters followed by numerical calibration on available data. It is therefore important to prepare a tool for calibration on multiple trajectories on selected subset of parameters and weighted data points.

3. Numerical calibration on available ratcheting experiments.

Chapter 4

Applied Methods

4.1 Finite element method

Following [A1], the basic laws of computational plasticity within the Finite Element method will be reminded. The equilibrium equation discretized by the Finite Element method reads

$$\int_{\Omega} \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma} \mathrm{d}\Omega = \mathbf{R}, \tag{4.1}$$

where $\mathbf{B}$ is the strain-displacement matrix, $\boldsymbol{\sigma}$ is the stress column vector and $\mathbf{R}$ is the equivalent external force acting on the nodal points. As the material behaves elasto-plastically, the stress-strain relation is no longer linear but forms an algebro-differential system according to the hardening laws. In general, this system can only be solved numerically.

4.1.1 Incremental problem

The nonlinear equations of elastoplasticity on the level of governing relations between nodal displacements and forces are solved by a suitable iteration procedure. Starting at time t in an equilibrium state of stress ${}^{t}\boldsymbol{\sigma}$, plastic strain ${}^{t}\boldsymbol{\epsilon}_{\mathrm{p}}$, and with a set of tensor and scalar internal variables $\{{}^{t}\mathbf{x}_i, {}^{t}y_j\}$, one is to determine ${}^{t+1}\boldsymbol{\sigma}$, ${}^{t+1}\boldsymbol{\epsilon}_{\mathrm{p}}$ and $\{{}^{t+1}\mathbf{x}_i, {}^{t+1}y_j\}$ at time $t+1$. For this purpose, the iteration procedure proposes trial increment $\Delta\boldsymbol{\epsilon}^{\mathrm{trial}}$, which is imported into the constitutive model to calculate stress response, so that a new improved estimation of $\Delta\boldsymbol{\epsilon}^{\mathrm{trial}}$ can be obtained. This process is repeated until equilibrium is reached. The quasi-Newton Broyden-Fletcher-Goldfarb-Shano (BFGS) method may be employed to this end, as introduced in [26]. In the case a Newton-Raphson solver is used, a continual tangent can be acquired according to Eq. (2.20).

4.1.2 Integration scheme

There are a number of integration schemes suitable for general plastic constitutive model. Due to unconditional stability, the implicit backward Euler scheme

$$ {}^{t+1}\boldsymbol{\sigma} = {}^{t}\boldsymbol{\sigma} + \Delta\boldsymbol{\sigma}^{\mathrm{trial}} - \Delta\lambda \mathbf{C} \left.\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}\right|_{t+1}, \tag{4.2} $$

is preferred. The main downside is the computational demands, as it has to be iterated. Additionally, within Eq. (4.2) the value of plastic multiplier has to be solved, so for the updated stress remains $\mathrm{f}(\boldsymbol{\sigma}, \mathbf{x}_i, y_j)|_{t+1} = 0$.

In this thesis, many abrupt changes in the developed model happen during general loading, such as unlocking the plateau, or complex internal processes withing the new kinematic rule. Despite good experience with the substepping method by Sloan in [38], as it was successfully implemented to a distortional model in [A3], a slight modification of explicit scheme is used, which enables greater control over the integration step length and therefore interrupts the integration process in the moments of discrete changes of internal variables. The stress response is calculated by the tangent stiffness-radial corrector method, originally introduced in [41]. After proposing a trial stress value from Hooke's law

$$ \boldsymbol{\sigma}^{\mathrm{trial}} = {}^{t}\boldsymbol{\sigma} + \Delta\boldsymbol{\sigma}^{\mathrm{trial}} = {}^{t}\boldsymbol{\sigma} + \mathbf{C}\Delta\boldsymbol{\epsilon}^{\mathrm{trial}}, \tag{4.3} $$

the elastic check is performed. If $\mathrm{f}(\boldsymbol{\sigma}^{\mathrm{trial}}) < \mathtt{tol}$, the elastic law is considered valid and the stress response is set as ${}^{t+1}\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathrm{trial}}$. Otherwise, if the elastic check is not fulfilled, a search for the actual yield stress is initiated. The intersection point

$$ \boldsymbol{\sigma} = {}^{t}\boldsymbol{\sigma} + (1-g)\Delta\boldsymbol{\sigma}^{\mathrm{trial}} \quad \text{for } g \in [0,\, 1] \tag{4.4} $$

with the yield surface $\mathrm{f}(\boldsymbol{\sigma}, \mathbf{x}_i, y_j)|_t = 0$ is computed iteratively by Newton's method, i.e.

$$ g_{i+1} = g_i + \left.\frac{f}{\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \Delta\boldsymbol{\sigma}^{\mathrm{trial}}}\right|_{\boldsymbol{\sigma} = {}^{t}\boldsymbol{\sigma} + (1-g_i)\Delta\boldsymbol{\sigma}^{\mathrm{trial}}} \tag{4.5} $$

until the prescribed tolerance is met. The remaining plastic part $\Delta\boldsymbol{\sigma}_{\mathrm{p}}^{\mathrm{trial}} = g\Delta\boldsymbol{\sigma}^{\mathrm{trial}}$ of the trial stress enters the procedure described in the next section, from which a new equilibrium state ${}^{t+1}\boldsymbol{\sigma}$, ${}^{t+1}\boldsymbol{\epsilon}_{\mathrm{p}}$, $\{{}^{t+1}\mathbf{x}_i,\ {}^{t+1}y_j\}$, satisfying the plasticity condition, arises. More information is available in [33] together with the proof of convergence for an arbitrary convex yield function. This state returns to the core of the implicit finite element solver as a basis for a new iteration of the trial estimation.

A square root form of the proposed yield function does not represent a hyperconical function. Still, for the proportional loading only, the function on the main axis is linear and so the above iteration process converges to $\mathrm{f} = 0$ within one step. An osculation sphere on the frontal apex and on the rear may speedup these calculations on some specific loading conditions.

Handling all the rates as increments in dimensionless time, one may rewrite the original expression for the plastic multiplier (2.19) into the incremental form as

$$\Delta\lambda = \frac{\dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \Delta\boldsymbol{\sigma}_{\mathrm{p}}^{\mathrm{trial}}}{K_{\mathrm{p}} + \dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \mathbf{C} \dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}}. \tag{4.6}$$

Acquiring the plastic multiplier, the evolution rules can be followed to calculate $\{{}^{t+1}\mathbf{x}_i,\ {}^{t+1}y_j\}$, using Eqs. 2.15 and 2.16. Consequently, rewriting the differential form (2.17) into the incremental form, the stress tensor yields

$${}^{t+\Delta t}\boldsymbol{\sigma}^{*} = {}^{t}\boldsymbol{\sigma} + \Delta\boldsymbol{\sigma}^{\mathrm{trial}} - \Delta\lambda\mathbf{C}\left.\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}\right|_{t}. \tag{4.7}$$

This updated value, marked by asterisk, generally does not satisfy the yield condition. A corrector scheme described below needs to be applied to achieve unconditional stability, see [29]. From this, the new value of stress ${}^{t+\Delta t}\boldsymbol{\sigma}$ is obtained. To keep track of the accumulated plastic strain, the flow rule (2.1) in scalar form may be used as

$$\Delta\epsilon_{\mathrm{p}} = \sqrt{\frac{2}{3}}\Delta\lambda\left\|\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}\right\|. \tag{4.8}$$

4.1.3 Radial return corrector

Distortion of the yield surface offers various approaches to project the stress back onto the yield surface. A gradient method

$${}^{t+\Delta t}\boldsymbol{\sigma}^{*}_{i+1} = {}^{t+\Delta t}\boldsymbol{\sigma}^{*}_{i} - \frac{\partial \mathrm{f}_i}{\partial \boldsymbol{\sigma}} f_i \left\|\frac{\partial \mathrm{f}_i}{\partial \boldsymbol{\sigma}}\right\|^{-2} \tag{4.9}$$

may be the only choice when no convenient procedure exists. Again, for regions very close to the axis of the yield surface, analytical solution for projection on osculation spheres may be used. It depends on the complexity of the terms defining their position and radius.

The radial return procedure will on many occasions require correction of the stress response from within the elastic region of the updated yield function. A step size limit has to be determined, so as to prevent this correction to be initiated from

a position too close to the singularity at ${}^{t+1}\boldsymbol{\alpha}$, where it would lose clear definition of the appropriate projection point.

4.1.4 Accuracy and maximum step

Accuracy of the algorithm is controlled by the appropriate subdivision during integration and by adjusting the desired tolerance. Subdivision of the integration is of a very high importance during rapid evolution and non-proportional loadings. Therefore, the trial stress increment fed into the evolution equations gradually with a varying step length while satisfying all actual, precise or conservative, safety limits for each evolution rule. These limits come from possible overshooting of the asymptotic levels of individual internal variables as well as an empirical estimate of their effect on overall precision.

The choice of numerical tolerance affects the general accuracy mainly during rapid changes of loading directions. Its default limit, $\texttt{tol} = 10^{-4}k_0$, is sufficient for any practical use. Trial stress increments that are tangential to the yield surface may induce very small negative value of $\Delta\lambda$. This phenomenon is proportional to the value of `tol` and should be corrected by the use of Macaulay brackets.

4.1.5 Stability and consistency

The presented implementation employs single step Euler's forward integration scheme with return mapping corrector. This method is linear, possessing linear convergence for continuous function, see [29]. For additional examples on various plasticity models see [45] and [1]. The Euler forward method is consistent with a truncation error of order one, see [11], [20] and [21]. It will be confirmed by proportional as well as non-proportional tests.

It is well known, that the Euler forward method is only conditionally stable, provided

$$\Delta\lambda_{\text{crit}} = \frac{2}{|\mu_{\max}|}, \tag{4.10}$$

where $\mu_{\max}$ is the maximum eigenvalue of the Jacobian matrix of the ODE system, defined by Eqns. (2.17) and (2.19), taken over the whole solution domain. The fully implicit corrector employed in [33] features unconditional stability as the return correction only applies to stress variables, not the distortional parameters.

4.2 Numerical tests

Due to the developments in information technology, more complex testing can be performed in the form of high resolution error maps. These maps can provide a greater picture, showing particular kinds of loading trajectories that strain the capabilities of the model or its implementation. For instance, high error may be present when material is subjected to loading direction tangential to the yield surface at the actual state of prestress.

A necessary condition for convergence of the equilibrium solver is a continuous stress response. Many minute details can create narrow stripes in the image of the implementation, where stress solution doesn't exist. This may be caused by various phenomena during evolution of internal variables, or a discontinuous function of subdivision into integration steps, see [A1].

4.3 Abaqus implementation

For easy cooperation within the scientific community, the models have been implemented into the commercial ABAQUS FE-code. A UMAT subroutine for static analysis in ABAQUS/Standard has been chosen.

```
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3)
C
C     User implementation output:
C     Material stress response:                  STRESS
C     Updated internal variables:                STATEV
C     Jacobian matrix of the constitutive model: DDSDDE
      RETURN
      END
```

Chapter 5

Results and Discussion

I suggested two new models with partially differing yield definitions. Both arise straight from the original alpha model in [7]. The problem of elongation, that greatly limited the amount of materials suitable for the anticipated behavior, becomes truly visible when performing calibration procedure in [A2]. Answering this and several other problems that were encountered during development, these new yield functions are equipped with correcting affine transformations shortening their axial length and widening their crosswise diameter. Their kinematic rules have been completely rewritten. By adopting the Bounding Surface principle, the models are now capable of simulating an initial plateau, as well as a better elastic-plastic transition and ratcheting behavior. The first model features coupled distortion with kinematic law, maintaining its axis of the elastic region coincident with the origin of its Bounding Surface. The second model is an extension of the first one by means of uncoupling this bond. Serving as a means to study non-proportional ratcheting behavior, the model's kinematic laws try to address and utilize specific behavioral traits that have been observed throughout literature, concerning yield shape evolution and rotation.

The main focus of this thesis is on implementation. Founded on widely used Euler's forward rule with radial return corrector, the implementation of these models has been optimized via controlling the integration step length. Advances in computing power enabled me to move from traditional iso-error maps and focus on the stability of the model by checking various states and trajectories, searching for unexpected behavior and observing internal variables.

A designated experimental campaign for these models has not been initiated yet. Any scenario has to be consulted with the model to search for the effect of distortion. The general idea is to repeatedly load in the directions where the distorted shape of the yield surface differs from general von Mises hypersphere. With a large set of ratcheting experiments, the goal is to study the movement of the Bounding Surface, as it has a great impact on ratcheting speeds. For now, a numerical calibration tool has been developed and used on selected experimental data.

5.1 New coupled bounding surface model

Using a square root form of Eq. (2.12) and using correcting transformations on the radial tensor, the yield condition acquires the following form:

$$\mathrm{f}(\boldsymbol{\sigma}) = \sqrt{\tfrac{3}{2}\left[1 - c\,\|\mathbf{z}\|\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]}\,\|\mathbf{b}\| - k = 0, \tag{5.1}$$

where c remains as the distortional parameter, $\mathbf{z}$ serves as a definition of the main axis of the Yield Surface and the aforementioned transformations are covered in the transformed radial stress $\mathbf{b}$. The size of the Yield Surface is controlled by the evolving isotropic parameter k. In this particular model, tensor $\mathbf{z}$ acquires the form

$$\mathbf{z} = \boldsymbol{\alpha} - \boldsymbol{\beta}, \tag{5.2}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the traditional backstress and the center of the Bounding Surface, respectively. In effect, the main axis of the YS always aims from $\boldsymbol{\beta}$, which makes the interaction between YS and BS very simple. This is an arbitrary decision and it is expected that this model will not behave well in cases of significant changes in the direction of stress.

A necessary condition required by the square root prescribes the limit value of c according to the limit state of $\mathbf{z}$ as

$$c^0_\mathrm{lim} = 1/\left\|\mathbf{z}^l\right\|^2 = L^2. \tag{5.3}$$

The value L will later be deduced from the kinematic evolution rules.

To simplify calibration, the frontal apex always stays the same distance from backstress. When distortion occurs, the apex increases curvature, while the rear flattens and draws closer to the value of backstress. The actual intensity of the distortion is proportional to the norm of $\mathbf{z}$. This implies a series of problems. Due to the nonlinear nature of the forming of distortion, a pronounced shape is present only when $\|\mathbf{z}\|$ starts to saturate to its limit L. Additionally, when unloading occurs, the flattening of the rear disappears back to the von Mises sphere where it remains in a constant distance from backstress. This implies a first order discontinuity in the stress response. To combat this phenomenon, an additional term $\|\mathbf{z}\|$ has been added to the yield condition (5.1), smoothing out the onset of the generated distortion. This further delayed the appearance of a pronounced distortion. Therefore, the distortion had to be sped up. Using an internal function to do that was expected to unnecessarily complicate the yield condition and its gradients. Therefore, the distortion is rather kept constant once it reaches a chosen limit. The yield condition is

therefore reformulated as follows:

$$\mathrm{f}(\boldsymbol{\sigma}) = \sqrt{\tfrac{3}{2}\left[1 - c_{\mathrm{zz}}\left(\mathbf{n}_{\mathrm{r}} : \mathbf{n}_{\mathrm{z}}\right)\right]}\,\|\mathbf{b}\| - k = 0, \tag{5.4}$$

where

$$c_{\mathrm{zz}} = \min\left\{c\mathbf{z}:\mathbf{z}, c_{\mathrm{zz}}^{\max}\right\};\ c_{\mathrm{zz}}^{\max} \leq c_{\mathrm{lim}}^{1} \leq c_{\mathrm{lim}}^{0} \tag{5.5}$$

unifies the yield condition for semi- and fully distorted state. A necessary condition forming the limit c_{lim}^{1} stems from a convexity requirement derived later.

5.1.1 Contraction and cross effect

The modified radial stress $\mathbf{b}$ consists of the original radial stress followed by an axial contraction reached by inverting the Yield function, so that the elongation of the frontal apex is exactly counteracted. This brings a potential for a much simpler analytical solution of monotonic loading. At the very least, a calibration of the model based on monotonic loading is not dependent on the distortional parameter c.

The modification of the transverse part follows for the full cross-effect control. Unexpectedly, it is done only half way. Not applying transformation to the unit tensor $\mathbf{n}_{\mathrm{r}}$ in the yield condition offers a larger diversity of shapes. If it were the case of full application, the shape would be precisely scaled in transverse direction and would be lacking in front apex curvature. The structure of $\mathbf{b}$ for the nominal yield function (5.1) contains correction of elongation and scaling of cross-wise diameter in the following form:

$$\mathbf{b} = (\mathbf{s}-\boldsymbol{\alpha}) + \underbrace{\left[\frac{1}{\sqrt{1-c\mathbf{z}:\mathbf{z}}} - 1\right]}_{\text{counteracting distortion}} \underbrace{\mathbf{n}_{\mathrm{z}}\mathbf{n}_{\mathrm{z}}:(\mathbf{s}-\boldsymbol{\alpha})}_{\text{full axial part}} - c_{\mathrm{cr}}\|\mathbf{z}\| \underbrace{\left[(\mathbf{s}-\boldsymbol{\alpha}) - \underbrace{\mathbf{n}_{\mathrm{z}}\mathbf{n}_{\mathrm{z}}:(\mathbf{s}-\boldsymbol{\alpha})}_{\text{full axial part}}\right]}_{\text{full transverse part}}. \tag{5.6}$$

After reduction, the modified effective stress $\mathbf{b}$ yields

$$\mathbf{b} = \left(1 - c_{\mathrm{cr}}\|\mathbf{z}\|\right)(\mathbf{s}-\boldsymbol{\alpha}) + \left[\frac{1}{\sqrt{1-c\mathbf{z}:\mathbf{z}}} + c_{\mathrm{cr}}\|\mathbf{z}\| - 1\right]\mathbf{n}_{\mathrm{z}}\mathbf{n}_{\mathrm{z}}:(\mathbf{s}-\boldsymbol{\alpha}). \tag{5.7}$$

The new yield function is not conical any more, with negative implications for quick return mapping algorithms using central projection, as used in Eq. (29) in [A1].

The unification for limited distortion in Eq. (5.4) shapes Eq.(5.7) to the following form:

$$\mathbf{b} = \left(1 - c_{\mathrm{crz}}\right)(\mathbf{s}-\boldsymbol{\alpha}) + \left[\frac{1}{\sqrt{1-c_{\mathrm{zz}}}} + c_{\mathrm{crz}} - 1\right]\mathbf{n}_{\mathrm{z}}\mathbf{n}_{\mathrm{z}}:(\mathbf{s}-\boldsymbol{\alpha}), \tag{5.8}$$

where

$$c_{\mathrm{crz}} = \min\left\{c_{\mathrm{cr}} \left\|\mathbf{z}\right\|, c_{\mathrm{crz}}^{\max}\right\} \tag{5.9}$$

For a homogenous polycrystalline metal without cavities, the associative flow rule is chosen as

$$\dot{\boldsymbol{\varepsilon}}_{\mathrm{p}} = \lambda \frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}. \tag{5.10}$$

If needed, the cumulative plastic strain can be monitored by

$$\dot{\varepsilon}_{\mathrm{cum}} = \sqrt{\tfrac{3}{2}\dot{\boldsymbol{\varepsilon}}_{\mathrm{p}} : \dot{\boldsymbol{\varepsilon}}_{\mathrm{p}}}. \tag{5.11}$$

5.1.2 Yield function gradient

Implementation of the new model requires partial derivatives of the yield function according to all state variables, forming the consistency equation. As the stress tensor undergoes elimination of the volumetric part inside the yield function, a deviatoric part of the identity tensor is required, as in Eqs. (5.12) and (5.13). In both cases, symmetrization may be applied.

$$\frac{\partial \mathbf{n}_{\mathrm{r}}}{\partial \boldsymbol{\sigma}} = \frac{\mathbf{I}^D - \mathbf{n}_{\mathrm{r}} \otimes \mathbf{n}_{\mathrm{r}}}{\left\|\mathbf{s} - \boldsymbol{\alpha}\right\|}; \quad \mathbf{I}^D_{ijkl} = \delta_{ik}\delta_{jl} - \tfrac{1}{3}\delta_{ij}\delta_{kl} \tag{5.12}$$

$$\frac{\partial \mathbf{n}_{\mathrm{r}}}{\partial \boldsymbol{\alpha}} = \frac{\mathbf{n}_{\mathrm{r}} \otimes \mathbf{n}_{\mathrm{r}} - \mathbf{I}}{\left\|\mathbf{s} - \boldsymbol{\alpha}\right\|}; \quad I_{ijkl} = \delta_{ik}\delta_{jl} \tag{5.13}$$

The yield function gradient from Eq. (5.4) assumes the following form:

$$\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} = \left\|\mathbf{b}\right\| \frac{c_{\mathrm{zz}}}{\sqrt{\frac{8}{3}\left[1 - c_{\mathrm{zz}}\left(\mathbf{n}_{\mathrm{r}} : \mathbf{n}_{\mathrm{z}}\right)\right]}} \frac{\mathbf{n}_{\mathrm{r}} : \mathbf{n}_{\mathrm{z}}\mathbf{n}_{\mathrm{r}} - \mathbf{n}_{\mathrm{z}}}{\left\|\mathbf{s} - \boldsymbol{\alpha}\right\|} + \sqrt{\tfrac{3}{2}\left[1 - c_{\mathrm{zz}}\left(\mathbf{n}_{\mathrm{r}} : \mathbf{n}_{\mathrm{z}}\right)\right]} \frac{\mathbf{b}}{\left\|\mathbf{b}\right\|} : \frac{\partial \mathbf{b}}{\partial \boldsymbol{\sigma}}, \tag{5.14}$$

where

$$\frac{\partial \mathbf{b}}{\partial \boldsymbol{\sigma}} = \left(1 - c_{\mathrm{crz}}\right)\mathbf{I}^D + \left(\frac{1}{\sqrt{1 - c_{\mathrm{zz}}}} + c_{\mathrm{crz}} - 1\right)\mathbf{n}_{\mathrm{z}} \otimes \mathbf{n}_{\mathrm{z}}, \tag{5.15}$$

both using definitions (5.5) nad (5.9). Remaining gradients in respect to other relevant variables are put together in Appendix A.

5.1.3 Internal evolution rules

Isotropic softening is clearly measurable during depletion of the plateau. There may even be a discrete change in the plastic modulus when depletion occurs, see Fig. 2.1. Therefore, two rules of isotropic hardening were suggested. The first one represents

a rapid isotropic softening within the plateau region only, the second accounts for a very slow change after the plateau, likely as a means to simulate transient speed of ratcheting before stabilization occurs. κ_1 and κ_3 represent the rate of hardening, κ_2^{-1} and κ_4^{-1} represent their respective asymptotic limits. As the plateau region ends before k saturates to κ_2^{-1}, the isotropic parameter continues to evolve according to the second rule from its last value, making the limit κ_2^{-1} only theoretical, so that

$$\dot{k} = \lambda\kappa_1(1-\kappa_2 k) \Leftrightarrow \varepsilon_{\text{cum}} < \varepsilon_{\text{plat}}; \qquad \dot{k} = \lambda\kappa_3(1-\kappa_4 k) \Leftrightarrow \varepsilon_{\text{cum}} \geq \varepsilon_{\text{plat}}. \tag{5.16}$$

Rapid softening from the initial virgin state is counteracted by fast kinematic hardening, keeping the stress response constant. Significant isotropic softening allows smooth elastic-plastic transitions with a pronounced Bauschinger effect. Within the plateau, the Bounding Surface does not move from the origin and only its growing size allows backstress to cumulate. When plateau region is depleted, the movement of BS is unlocked and its speed is almost constant in the direction of normal to YS. Adding a recurrent term provides a degree of freedom as well as a necessary asymptotic limit, so that

$$\dot{\boldsymbol{\beta}} = \mathbf{0} \Leftrightarrow \varepsilon_{\text{cum}} < \varepsilon_{\text{plat}}; \qquad \dot{\boldsymbol{\beta}} = \lambda b1\,(\mathbf{n} - b_2\boldsymbol{\beta}) \Leftrightarrow \varepsilon_{\text{cum}} \geq \varepsilon_{\text{plat}}. \tag{5.17}$$

This limit stems from thermodynamic laws, as the amount of free plastic energy ψ_{p} stored in the material is very small and depends on the plastic modulus in respect to stress, see Figure 2.1.

Cyclic hardening is adopted by a simple isotropic hardening of the Bounding Surface as

$$\dot{K} = -\dot{k} \Leftrightarrow \varepsilon_{\text{cum}} < \varepsilon_{\text{plat}}; \qquad \dot{K} = -\dot{k} + K_1\,(1 - K_2\,(K-k)) \Leftrightarrow \varepsilon_{\text{cum}} \geq \varepsilon_{\text{plat}}. \tag{5.18}$$

In the case the material does not show plateau effect or is initiated from an arbitrary state, the bounding surface may have non-zero initial size as $K_0 > 0$. This directly influences the initial hardening of the material.

5.1.4 New kinematic hardening law

Due to the effect of distortion on the stress response during unloading, a new kinematic hardening rule had to be assembled. From the great many published kinematic rules that could be adopted, a new idea came to mind: A simplification of an otherwise well behaving multi-surface kinematic rule. This new approach rethinks the way the model remembers the history of loading, more specifically, it assumes that earlier history with less pronounced phenomena may be forgotten. In effect,

this model uses a limited amount of tensors forming a hierarchy to represent distinct groups of nested yield surfaces of a multisurface equivalent, see Fig. 5.1. The backstress is simply a summation of individually evolved components as

$$\boldsymbol{\alpha} \;=\; \boldsymbol{\beta} + \sum_{i=1}^{4} \boldsymbol{\alpha}_i \tag{5.19}$$

with the Bounding Surface center $\boldsymbol{\beta}$ added to the sum, making the interactions within backstress kinematic rule relative to the BS, as it does seem to be its effect on the stress response.

When new direction of loading is introduced, the primary component of backstress now respresents the newly formed band of the innermost equivalent surfaces. Smaller surfaces that are within the second band are continuously fed to the primary. During plastic loading, a flow of stress is present between the component tensors, while rapid transfers happen when new distinct direction of loading is introduced. In such a case, the history represented by the third component is irretrievably altered.

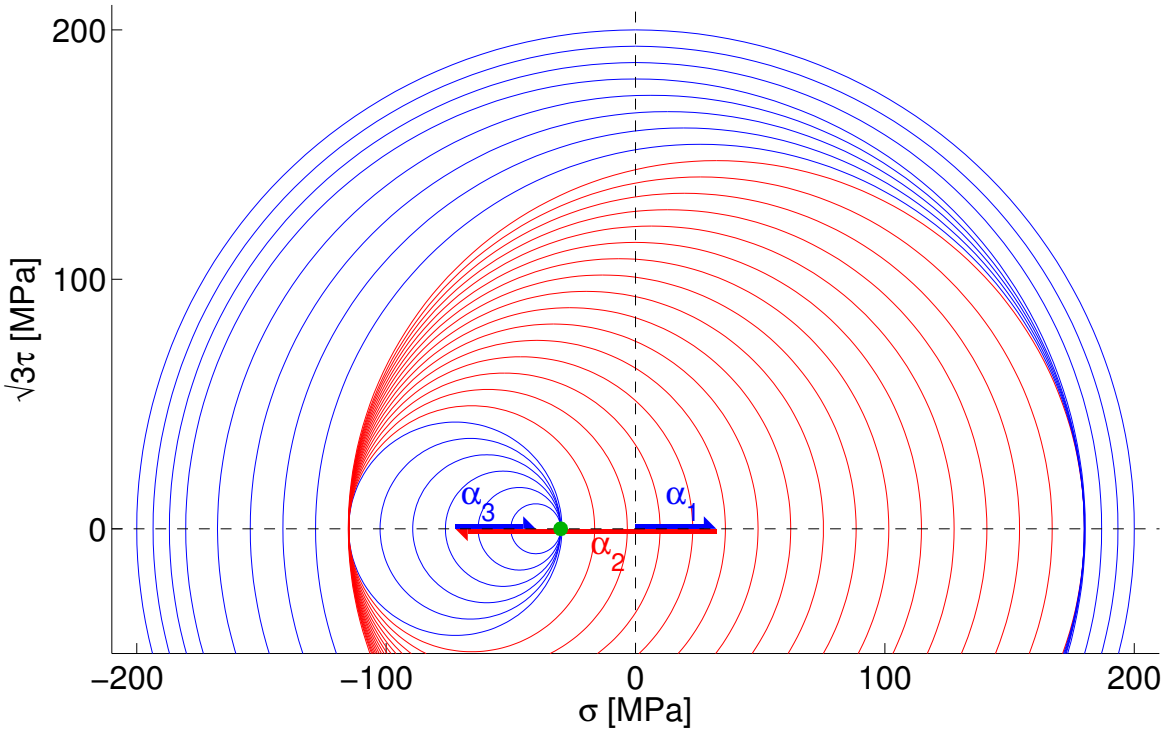

FIGURE 5.1: Backstress component tensors and their multi-surface equivalent for a virgin material after tension-compression-tension loading.

A multisurface model featuring such a high number of surfaces is capable of storing the effect of a long history of loading. Imagine each of the surfaces in an

arbitrary position within the confines of the "nested" condition. A general cyclic history with decreasing amplitude of stress would be able to create such a condition. It is however unimaginable, that a specimen in such a condition could undergo measurement to evaluate footprint of such a complex history. A very limited amount of tensors (4 in this rendition) should therefore be able to represent an optimal amount of history stored.

The multiaxial formulation of this model consists of the yield condition (5.4) with backstress confined in the Bounding Surface. The driving direction of the backstress kinematic rule uses its projection $\boldsymbol{\alpha}_{\mathrm{B}}$ on the Bounding Surface. This is consistent with the formulation in [25]. The choice to use interaction on the level of backstress rather than stress comes from problems related to the distortion feature and its interaction with BS. However, the coupled distortion of this particular model does not have such problems, as it is always aiming outwards with its frontal apex. Still, it is chosen for it to limit the number of differences between coupled and uncoupled models. The definition of the driving tensor is as follows:

$$\mathbf{n}_{\alpha\mathrm{B}} \equiv \frac{\boldsymbol{\alpha}_{\mathrm{B}} - \boldsymbol{\alpha}}{\|\boldsymbol{\alpha}_{\mathrm{B}} - \boldsymbol{\alpha}\|}, \tag{5.20}$$

where the projection of the backstress on the Bounding Surface is in the direction of the outer normal tensor, as

$$\boldsymbol{\alpha}^{\mathrm{B}} \equiv \boldsymbol{\beta} + \sqrt{\tfrac{2}{3}} K \mathbf{n}. \tag{5.21}$$

The situation is depicted in Figure 5.2, where blue is the Bounding Surface and green is the distorted Yield Surface.

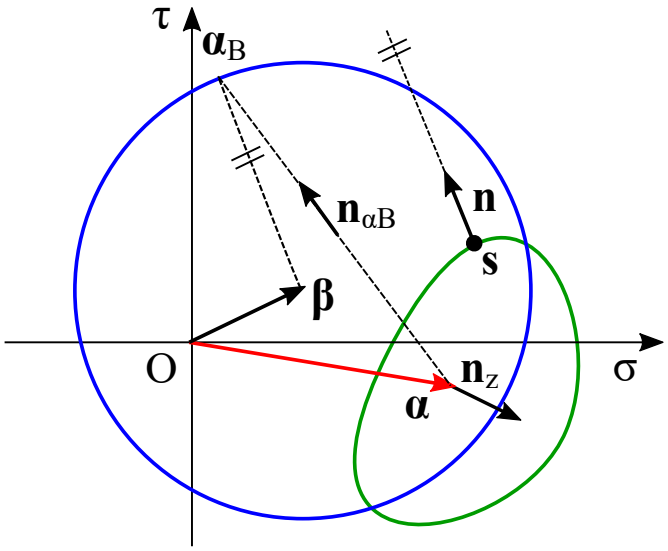

FIGURE 5.2: Definition of the backstress growing direction $\mathbf{n}_{\alpha\mathrm{B}}$

As the set of backstress components represent distinct bands of nested surfaces of a multisurface equivalent, a necessary transfer between the tensors is required. During loading, internal forces are stored by means of function

$$G_4 = \left(\sqrt{\tfrac{2}{3}} \bar{K} + h_1 \delta \right) \mathbf{n}_{\boldsymbol{\alpha}\mathrm{B}} \tag{5.22}$$

that represents opening of new slip systems for the storage of internal forces. A choice was made to apply this growth on the least active component, number 4. The distance towards the projected backstress

$$\delta \equiv \| \boldsymbol{\alpha}_{\mathrm{B}} - \boldsymbol{\alpha} \| \tag{5.23}$$

is used to shape and asymptotically limit the hardening curve when approaching the Bounding Surface. Once forces are stored, they can be very rapidly shared between components to simulate high values of plastic modulus during unloading. A dissipation function for such a process was contemplated, ultimately demonstrated to be unnecessary. The auxiliary components

$$\bar{\boldsymbol{\alpha}}_2 = -F_2 \mathbf{n}_{\boldsymbol{\alpha}2} + p_s \left(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2 \right) \tag{5.24}$$

$$\bar{\boldsymbol{\alpha}}_3 = -F_3 \mathbf{n}_{\boldsymbol{\alpha}3} + p_s \boldsymbol{\alpha}_2 \tag{5.25}$$

$$\bar{\boldsymbol{\alpha}}_4 = G_4 \mathbf{n}_{\boldsymbol{\alpha}\mathrm{B}} - F_4 \mathbf{n}_{\boldsymbol{\alpha}4} \tag{5.26}$$

provide internal forces for the primary component

$$\bar{\boldsymbol{\alpha}}_1 = \left(F_2 + F_3 + F_4 \right) \mathbf{n}_{\boldsymbol{\alpha}\mathrm{B}} - p_s \boldsymbol{\alpha}_1 \tag{5.27}$$

via the flow functions F_i acting in the directions represented by respective tensors $\mathbf{n}_{\boldsymbol{\alpha}i}$.

Initiated by a significant change in the direction of loading, a new primary component may be introduced. This would be done by a cascade of transfers, as described later. Such a discrete change necessarily creates discontinuities of the stress response. For that reason, function p_s represents smoothing that is activated in the transient space between proportional loading and cases of rapid change in direction of loading, as

$$p_{\mathrm{s}} = \left\langle \exp\left[-10 \left(\mathbf{n}_{\boldsymbol{\alpha}\mathrm{B}} : \mathbf{n}_{\boldsymbol{\alpha}i} - 0.8 \right) \right] - 1 \right\rangle . \tag{5.28}$$

For very distinct changes, p_s reaches virtual infinity and represents total transfer. The flow functions describe the rate in which the internal forces are shared among the components. An initial, memorized value of δ was used in [25] to speedup

the hardening response when short unloading occurs. The introduction of such a distinct choice is problematic in relation to the continuity of the stress response, especially for multiaxial loading.

Within the structure of the backstress components, a value that serves a similar purpose can be found as

$$S_i = \sum_{j=1}^{i-1} \|\boldsymbol{\alpha}_j\| \,, \tag{5.29}$$

which represents the radius of the fictitious nested surface on the leading edge of the i-th band. With it, the flow function is assembled as follows:

$$F_i = h_2 h_{\text{dist}} \underbrace{\frac{\delta}{S_i + \varepsilon}}_{\text{Hardening}} \underbrace{\frac{1}{(S_i + \delta)^2 + \varepsilon}}_{\text{Speedup}} \left[\underbrace{1 - \exp\left(-h_3 \|\boldsymbol{\alpha}_i\|\right)}_{\text{Depletion smoothing}} \right] F_{si}. \tag{5.30}$$

When using small yield surface, the return to previous level of stress after short unloading has to be that much quicker. So much in fact, that the Speedup function requires a quadratic term in the denominator. The idea is depicted in Figure 5.3:

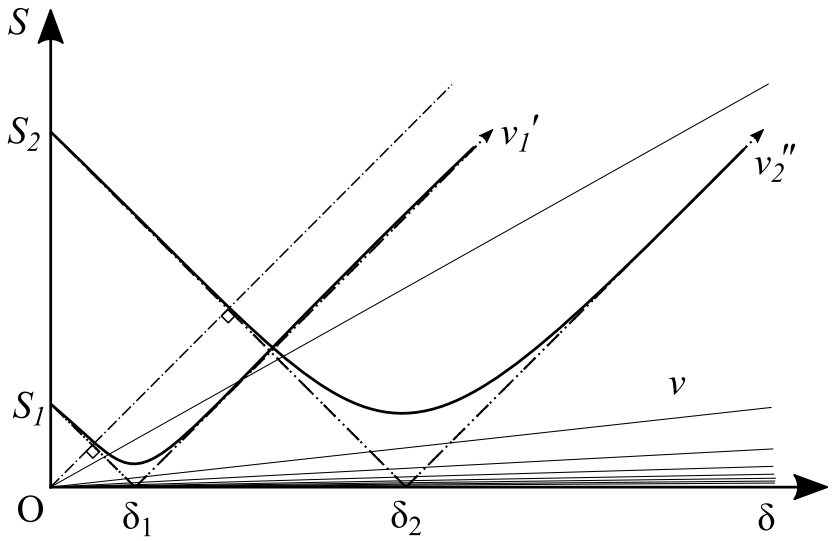

FIGURE 5.3: Sketched out idea behind backstress evolution speed.

A contour plot of evolution speed v is situated in the space of δ (distance to BS) and the size S_i of the leading edge of the i-th component. Two rotated views of v, cross sections v_1' and v_2", are depicted for better understanding.

Close to the bottom axis, S_i is almost zero and the rate of hardening is approaching infinity for smooth elastic-plastic transition. Near the vertical axis, δ is approaching zero and backstress asymptotically approaches the Bounding Surface. During

loading, the backstress component's hardening rate moves from a point on the horizontal axis towards the vertical axis in the top left direction. During this, the summation $S_i+\delta$ remains constant and represents the constant speedup coefficient required for the hardening curve.

In multiaxial case, the flow among tensors has to be controlled. When orthogonal direction is introduced to a prehistory of proportional loading, the contact points between nested surfaces rotate towards the new direction of loading. In this model, this is represented by a reduced or halted flow among backstress components. A shape function has been suggested to control the rate in relation to the angle between the particular component and the direction of the driving tensor $\mathbf{n}_{\boldsymbol{\alpha}\mathrm{B}}$ as

$$F_{\mathrm{s}i} = \frac{h_{\mathrm{a}}}{h_{\mathrm{b}} + (\mathbf{n}_{\boldsymbol{\alpha}\mathrm{B}} : \mathbf{n}_{\boldsymbol{\alpha}i})} + h_{\mathrm{c}}. \tag{5.31}$$

At the core, the shape function $F_{\mathrm{s}i}$ is set by parameters h_4 and h_5 representing the cosine of the angle of activation and its initial rate of growth. These form the actual parameters of Eq. (5.31) as:

$$h_{\mathrm{b}} = \frac{h_4 h_5 (h_4 - 1) + 1}{1 - h_5 \left(1 - h_4\right)} \tag{5.32}$$

$$h_{\mathrm{a}} = h_5 \left(h_{\mathrm{b}} - h_4\right)^2 \tag{5.33}$$

$$h_{\mathrm{c}} = \frac{h_{\mathrm{a}}}{h_4 - h_{\mathrm{b}}} \tag{5.34}$$

that are constant for a designated model calibration. Function F_{s} is equal to 0 for almost coincident tensors and 1 for opposite directions, see Fig. 5.4. Four different settings of the shape function are depicted. Initially equal to zero, they activate at two different values of $h_4 = 0$ and $h_4 = \sqrt{3}/2$, continuing in two different rates $h_5 = 0.5$ and $h_5 = 0.1$. The effect of this function is visible in equi-strain plot of the model. As the plastic modulus during plastic loading changes within several orders of magnitude from virtual infinity towards virtual zero, the constant h_4 opens up the flow late, when tensors are considerably opposing. If it were not the case, the stress response would show unrealistic rate of kinematic softening.

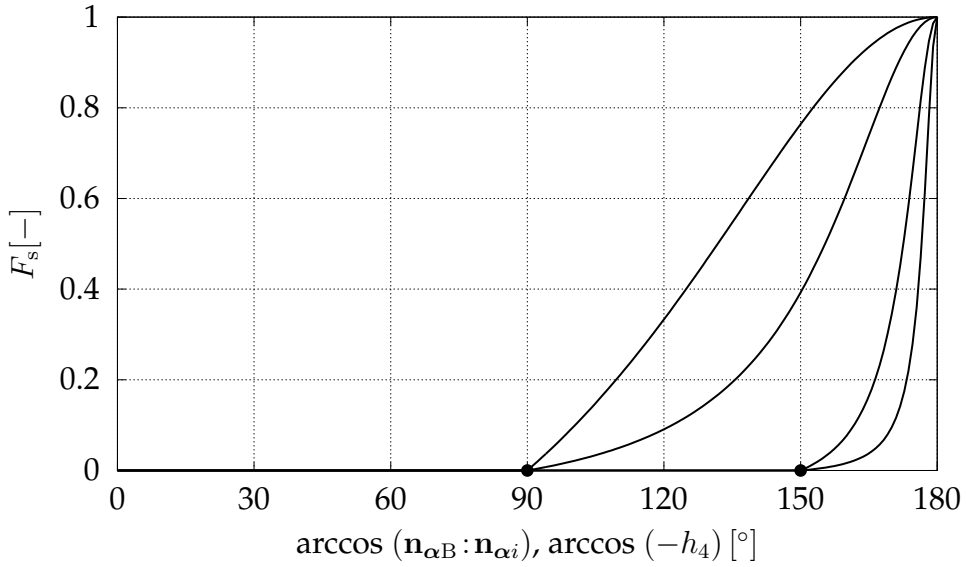

FIGURE 5.4: Various setting of flow intensity function for non-proportional loading

5.1.5 Convexity condition

For a stable return mapping algorithm required by the Euler method used in this model, the convexity of the Yield function has to be ensured. The needed sufficient and necessary condition is for its Hessian matrix to be positive definite. By differentiating the gradient (5.59), the Hessian matrix yields

$$\begin{aligned}
\frac{\partial^2 \mathrm{f}}{\partial\boldsymbol{\sigma}\partial\boldsymbol{\sigma}} = & \frac{c\,\|\mathbf{z}\|}{\sqrt{\frac{8}{3}\left[1-c\,\|\mathbf{z}\|\,(\mathbf{n}_\mathrm{r}:\mathbf{z})\right]}}\frac{\mathbf{b}}{\|\mathbf{b}\|}:\frac{\partial\mathbf{b}}{\partial\boldsymbol{\sigma}}\otimes\frac{\mathbf{n}_\mathrm{r}:\mathbf{z}\mathbf{n}_\mathrm{r}-\mathbf{z}}{\|\mathbf{s}-\boldsymbol{\alpha}\|}+ \\
& + \sqrt{\frac{3}{32}}\frac{c^2\,\|\mathbf{b}\|\,\|\mathbf{z}\|^2}{\sqrt{1-c\,\|\mathbf{z}\|\,(\mathbf{n}_\mathrm{r}:\mathbf{z})}^3}\frac{\mathbf{n}_\mathrm{r}:\mathbf{z}\mathbf{n}_\mathrm{r}-\mathbf{z}}{\|\mathbf{s}-\boldsymbol{\alpha}\|}\otimes\frac{\mathbf{z}-\mathbf{n}_\mathrm{r}\,(\mathbf{n}_\mathrm{r}:\mathbf{z})}{\|\mathbf{s}-\boldsymbol{\alpha}\|}+ \\
& + \frac{c\,\|\mathbf{z}\|\,\|\mathbf{b}\|}{\sqrt{\frac{8}{3}\left[1-c\,\|\mathbf{z}\|\,(\mathbf{n}_\mathrm{r}:\mathbf{z})\right]}}\cdot\frac{\left(\mathbf{I}^D-3\mathbf{n}_\mathrm{r}\otimes\mathbf{n}_\mathrm{r}\right)(\mathbf{n}_\mathrm{r}:\mathbf{z})+\mathbf{z}\otimes\mathbf{n}_\mathrm{r}+\mathbf{n}_\mathrm{r}\otimes\mathbf{z}}{\|\mathbf{s}-\boldsymbol{\alpha}\|^2}+ \\
& + \left[\frac{c\,\|\mathbf{z}\|}{\sqrt{\frac{8}{3}\left[1-c\,\|\mathbf{z}\|\,(\mathbf{n}_\mathrm{r}:\mathbf{z})\right]}}\frac{\mathbf{n}_\mathrm{r}:\mathbf{z}\mathbf{n}_\mathrm{r}-\mathbf{z}}{\|\mathbf{s}-\boldsymbol{\alpha}\|}\otimes\frac{\mathbf{b}}{\|\mathbf{b}\|}+\right. \\
& \left. + \sqrt{\tfrac{3}{2}\left[1-c\,\|\mathbf{z}\|\,(\mathbf{n}_\mathrm{r}:\mathbf{z})\right]}\frac{\partial\mathbf{b}}{\partial\boldsymbol{\sigma}}:\frac{\mathbf{I}\,(\mathbf{b}:\mathbf{b})-\mathbf{b}\otimes\mathbf{b}}{\|\mathbf{b}\|^3}\right]: \\
& : \left[\left(1-c_\mathrm{cr}\,\|\mathbf{z}\|\right)\mathbf{I}^D+\left(\frac{1}{\sqrt{1-c\mathbf{z}:\mathbf{z}}}+c_\mathrm{cr}\,\|\mathbf{z}\|-1\right)\mathbf{n}_\mathrm{z}\otimes\mathbf{n}_\mathrm{z}\right], \qquad (5.35)
\end{aligned}$$

where Eqs. (5.12) and (5.15) are used.

Closely following the method in [32], the Hessian is required to fulfill the following condition:

$$\forall \boldsymbol{\phi} \neq \mathbf{0}: \quad \boldsymbol{\phi} : \mathbf{H}\boldsymbol{\phi} > 0, \tag{5.36}$$

where $\boldsymbol{\phi}$ is an arbitrary second-order tensor. The structure of this condition enables normalizing $|\boldsymbol{\phi}| = 1$ and further simplifying the numerical search. Using a number of substitutions, Eq. (5.36) reaches a fully scalar form as

$$\begin{aligned} p\boldsymbol{\phi} : \mathbf{H}\boldsymbol{\phi} \;=\; & 2cL^2x\left(1 - cL^2x\right)\left(g^2 + 2ghx^2 + h^2x^2\right) \cdot \\ & \cdot \;\left[g^2\left(1 - y^2\right) + \left(h^2 + 2gh\right)\left(4xyz - 2z^2 + x^2 - 3x^2y^2\right)\right] + \\ & + \; 4g^2\left(1 - cL^2x\right)^2\left[g^2\left(1 - y^2\right) + h\left(h + 2g\right)\left(x^2 + z^2 - 2xyz\right)\right] - \\ & + \; c^2L^4\left(g^2 + 2ghx^2 + h^2x^2\right)^2\left(xy - z\right)^2 \geq 0 \end{aligned} \tag{5.37}$$

where $x = (\mathbf{n}_\mathrm{r} : \mathbf{z})$, $y = (\mathbf{n}_\mathrm{r} : \boldsymbol{\phi})$, $z = (\mathbf{z} : \boldsymbol{\phi})$, $g = (1 - Lc_\mathrm{cr})$, $h = \left[\frac{1}{\sqrt{1 - L^2|c|}} + Lc_\mathrm{cr} - 1\right]$. It is apparent from the Yield function (5.1, 5.7), that it is possible to state $\|\mathbf{z}\| \leftarrow L$, as it is always multiplied by either c, or c_cr. Therefore $L = \sqrt{\frac{2}{3}}\left(1/K_2 + 1/\kappa_4\right)$ as the limit size of the Bounding Surface coming from Eq. (5.18). The solution for limit convexity would therefore be relative to the value of L. The always positive part $p = 4\sqrt{\frac{2}{3}}\,\|\mathbf{s} - \boldsymbol{\alpha}\|\,\|g\mathbf{n}_\mathrm{r} + h\mathbf{z}x\|^3\left(1 - cL^2x\right)^{3/2} \geq 0$ can be factored out.

If the convexity limit were reached with non-saturated $\mathbf{z}$ only, it would appear by the existence of two valid solutions for c.

Convexity at a selected point

During initial investigation, two separate regions with concavity were found. One on the main axis at the back of the surface, the other on a ≈ 45 degree position. It was suspected that convexity limit on the main axis could possibly be derived analytically. Moreover, the yield function is linear in the direction of the main axis. For such points it is therefore expected that

$$\mathbf{H} = \mathbf{0}. \tag{5.38}$$

Assuming a point on the back of the yield surface with a normal tensor $\mathbf{n}$, the following terms were derived:

$$\begin{aligned}
\mathbf{s}-\boldsymbol{\alpha} &\leftarrow s\mathbf{n} \\
\mathbf{z} &\leftarrow -\mathbf{n};\ \text{saturated state for } \|\mathbf{z}\|=1 \\
\mathbf{n}_{\mathrm{z}} &\leftarrow -\mathbf{n} \\
\mathbf{n}_{\mathrm{r}} &\leftarrow \mathbf{n} \\
\mathbf{b} &\leftarrow (1-Lc_{\mathrm{cr}})\, s\mathbf{n} + \left[\frac{1}{\sqrt{1-L^2c}} + Lc_{\mathrm{cr}} - 1\right] s\mathbf{n} = \frac{s\mathbf{n}}{\sqrt{1-L^2c}} \qquad (5.39) \\
s|_{\mathrm{f}=0} &\leftarrow k\sqrt{\tfrac{3}{2}}\sqrt{\frac{1-L^2c}{1+L^2c}}. \qquad (5.40)
\end{aligned}$$

A special case of the Hessian for back side of the yield surface is therefore

$$\mathbf{H} = \frac{\partial^2 \mathrm{f}}{\partial\boldsymbol{\sigma}\partial\boldsymbol{\sigma}} = \left[\frac{-L^2c + 2\left(1-L^4c^2\right)\left(1-Lc_{\mathrm{cr}}\right)^2}{s\sqrt{\frac{8}{3}\left(1-L^4c^2\right)}}\right]\left(\mathbf{I}^D - \mathbf{n}\otimes\mathbf{n}\right). \qquad (5.41)$$

Ultimately, Eq. (5.40) is of no use, as the numerator defines the convexity limit via Eq. (5.38) and leads to a quadratic equation

$$2(1-Lc_{\mathrm{cr}})^2L^4c^2 + L^2c - 2(1-Lc_{\mathrm{cr}})^2 = 0. \qquad (5.42)$$

As the cross-effect is much more distinct than distortion and the distortion is expected to be very pronounced or even set as a limit capability of this model, it is reasonable to calculate c^1_{lim} from a calibrated value of c_{cr}. The axial necessary condition is therefore

$$c^{\mathrm{axial}}_{\mathrm{lim}} = \frac{-1+\sqrt{1+16\left(1-Lc_{\mathrm{cr}}\right)^4}}{4L^2(1-Lc_{\mathrm{cr}})^2}. \qquad (5.43)$$

Global convexity

The Yield function is at the convexity limit, if for the largest possible c holds

$$\inf\left(p\boldsymbol{\phi}:\mathbf{H}\boldsymbol{\phi}\right) = 0. \qquad (5.44)$$

Proof of rotation symmetry To simplify the numerical search for c^1_{lim} on the (x,y,z) space by limiting it to the meridian plane only, a rotation symmetry has to be proven. The simplest way is acknowledging that $\mathrm{f}(\boldsymbol{\sigma})$ remains constant in respect to the angular position around the main axis. If the meridian is convex, the outer normal

aims outwards from the main axis and the function is convex in all directions. In five-dimensional deviatoric space, there are three reciprocally orthogonal directions, all orthogonal to the actually investigated meridian plane. A linear combination of these three represent tensor ϕ in the following statement:

$$\left(\forall \boldsymbol{\phi} \in \mathbf{R}^9; \boldsymbol{\phi} = \boldsymbol{\phi}^{\mathrm{T}}; \mathrm{tr}\boldsymbol{\phi} = 0; (\mathbf{s} - \boldsymbol{\alpha}) : \boldsymbol{\phi} = 0; \mathbf{z} : \boldsymbol{\phi} = 0\right) : \left. \frac{\partial \mathrm{f}\left(\boldsymbol{\sigma} + \boldsymbol{\phi}\mathrm{t}\right)}{\partial t} \right|_{t=0} = 0 \quad (5.45)$$

Proof:

$$\mathrm{f}(\boldsymbol{\sigma} + \boldsymbol{\phi} t) = \sqrt{\frac{3}{2}\left[1 - \frac{c\|\mathbf{z}\|(\mathbf{s}-\boldsymbol{\alpha}):\mathbf{z}}{\sqrt{(\mathbf{s}-\boldsymbol{\alpha}):(\mathbf{s}-\boldsymbol{\alpha})+t^2}}\right]} \|\mathbf{b}\| - k, \quad (5.46)$$

where

$$\mathbf{b} = \left(1 - c_{\mathrm{cr}} \|\mathbf{z}\|\right)(\mathbf{s} - \boldsymbol{\alpha} + \boldsymbol{\phi} t) + \left[\frac{1}{\sqrt{1 - c\|\mathbf{z}\|^2}} + c_{\mathrm{cr}}\|\mathbf{z}\| - 1\right] \mathbf{n}_{\mathrm{z}}\mathbf{n}_{\mathrm{z}} : (\mathbf{s} - \boldsymbol{\alpha}). \quad (5.47)$$

Simplifying with substitution

$$\mathbf{b} = \left(1 - c_{\mathrm{cr}} \|\mathbf{z}\|\right) \boldsymbol{\phi} t + \mathbf{x}; \text{ where } \frac{\partial \mathbf{x}}{\partial t} = \mathbf{0}; \quad \boldsymbol{\phi} : \mathbf{x} = 0 \quad (5.48)$$

and differentiating, the gradient yields

$$\begin{aligned} \left. \frac{\partial \mathrm{f}\left(\boldsymbol{\sigma} + \boldsymbol{\phi}\mathrm{t}\right)}{\partial t} \right|_{t=0} = \|\mathbf{b}\| \frac{-3t\left(\mathbf{s} - \boldsymbol{\alpha}\right):\mathbf{z}}{2\sqrt{\frac{3}{2}\left[1 - \frac{c(\mathbf{s}-\boldsymbol{\alpha}):\mathbf{z}}{\sqrt{(\mathbf{s}-\boldsymbol{\alpha}):(\mathbf{s}-\boldsymbol{\alpha})+t^2}}\right]}\sqrt{(\mathbf{s}-\boldsymbol{\alpha}):(\mathbf{s}-\boldsymbol{\alpha})+t^2}^3} + \\ + \sqrt{\frac{3}{2}\left[1 - \frac{c(\mathbf{s}-\boldsymbol{\alpha}):\mathbf{z}}{\sqrt{(\mathbf{s}-\boldsymbol{\alpha}):(\mathbf{s}-\boldsymbol{\alpha})+t^2}}\right]} \frac{\left(1 - c_{\mathrm{cr}}\right)^2 t}{\sqrt{\left(1 - c_{\mathrm{cr}}\right)^2 t^2 + \mathbf{x}:\mathbf{x}}} = 0 \end{aligned} \quad (5.49)$$

and proves f is in the peripheral direction constant.

Numerical search The search for c_{lim} now moves to the meridian plane only, where for the direction cosines $x = \cos\beta$ and $y = \cos\gamma$ the rule of angular operations holds $z = \cos(\beta \pm \gamma)$.

The resulting function $c^1_{\mathrm{lim}} = c_{\mathrm{lim}}(c_{\mathrm{cr}})$ is plotted in Fig. 5.5. As analytical solution is inconceivable, a conservative sufficient condition was proposed. Unless the calibration procedure demands a more precise limit, mere scaling of function (5.43) should be sufficient and simple. The sufficient condition of convexity for the yield

function therefore is

$$c_{\text{lim}}^{1} = 0.968\, c_{\text{lim}}^{\text{axial}}. \tag{5.50}$$

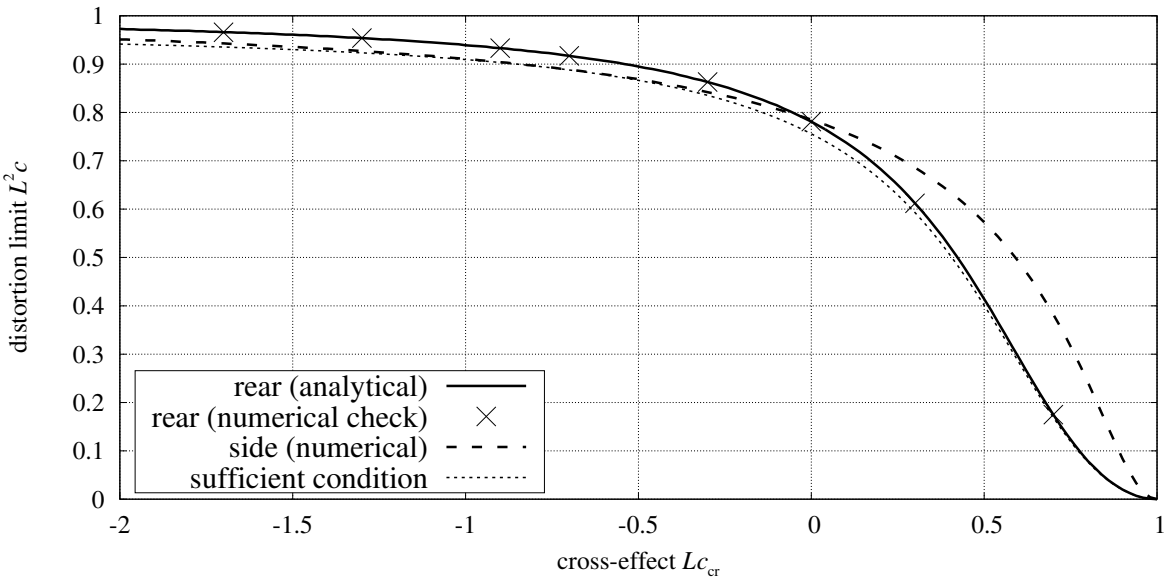

FIGURE 5.5: Parameter limits for the convexity condition

The discontinuous behavior connected to switching direction of distortion requires one additional correcting element h_{dist} in Eq. 5.30 as follows:

$$h_{\text{dist}} = 1 + \langle \mathbf{n}_{\boldsymbol{\alpha}\text{B}} : \mathbf{n}_{\text{z}} \rangle \left(\frac{h_{\text{ck}} \left\| \mathbf{s} - \boldsymbol{\alpha} \right\| + h_{\text{clim}}}{h_{\text{clim}}} - 1 \right) \tag{5.51}$$

where

$$h_{\text{clim}} = -\sqrt{\tfrac{3}{2} \tfrac{1+c_{\text{zz}}^{\max}}{1-c_{\text{zz}}^{\max}}}, \tag{5.52}$$

$$h_{\text{ck}} = \left[\frac{c_{\text{zz}}^{\max} h_{\text{clim}}}{1 - c_{\text{zz}}^{\max}} - \frac{c_{\text{zz}}^{\max}}{\sqrt{\tfrac{2}{3}\left[1 - (c_{\text{zz}}^{\max})^2\right]}} \right] / L \tag{5.53}$$

and where the fused maximum distortion parameter holds

$$c_{\text{zz}}^{\max} = 0.968 \frac{-1 + \sqrt{1 + 16\left(1 - c_{\text{cr0}}\right)^4}}{4\left(1 - c_{\text{cr0}}\right)^2}. \tag{5.54}$$

5.1.6 Model calibration

The model has been calibrated according to the experiment in Fig. 2.1 with parameters presented in Table 5.1. Parameters c_0 and c_{cr} are chosen arbitrarily, as there is no additional information about the rate and intensity of distortional hardening.

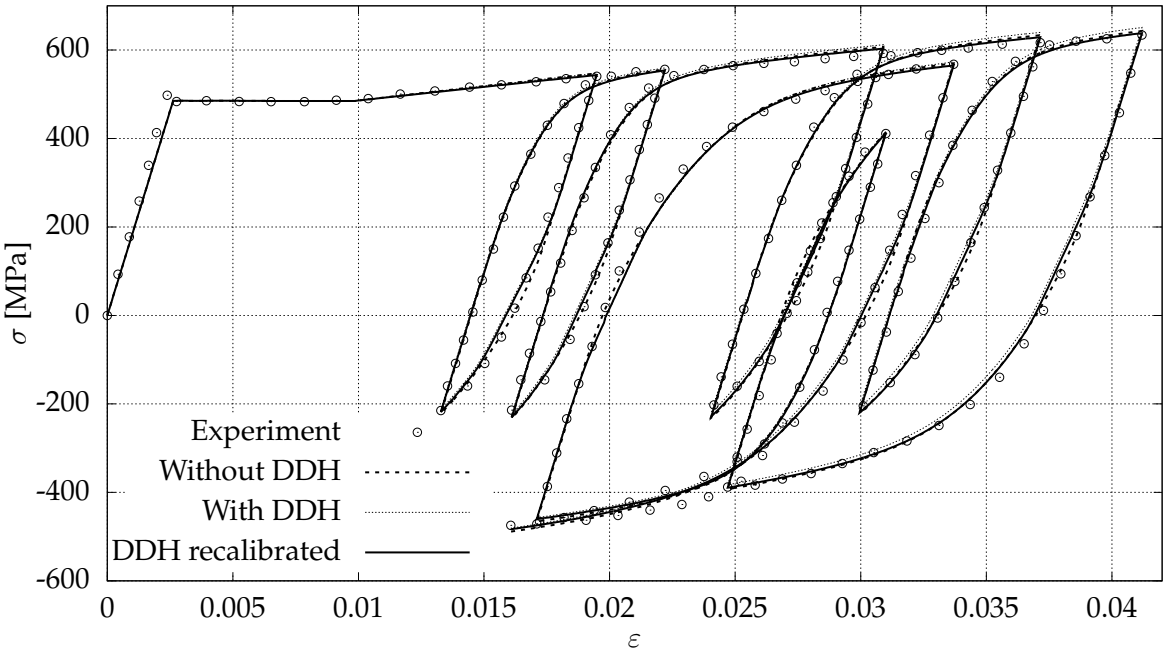

FIGURE 5.6: Demonstration of the model in random uniaxial cyclic loading calibrated to experimental data in [3].

Param.	Value	Units	Param.	Value	Units
E	185000	MPa	K_0	0.01	MPa
ν	0.29		K_1	89.86	MPa
k_0	485	MPa	K_2	3.2743×10^{-3}	MPa^{-1}
κ_1	56977	MPa	b_1	5358	MPa
κ_2	6.9095×10^{-3}	MPa^{-1}	b_2	2.366×10^{-3}	MPa^{-1}
κ_3	21489	MPa	ϵ_{plat}	7.2×10^{-3}	
κ_4	5.259×10^{-3}	MPa^{-1}	c_0	3	
h_1	30.0355	-	c_{cr0}	0	
h_2	3540845000	MPa^3			
h_3	0.026633	MPa^{-1}			
h_4	0.5	-			
h_5	0.5	-			
ε	0.01	MPa, MPa^2			

TABLE 5.1: Calibration from proportional test after [3]

Figure 5.7 depicts the same experiment with respect to the cumulative plastic strain. Initially, $\boldsymbol{\alpha}_4$ grows with the depletion of the plateau. First, a short unloading trajectory activates the first backstress segment. $\boldsymbol{\alpha}_4$ has stored forces in the opposite

direction and is therefore eager to provide component α_1 with its content. After that, the original prestress returns and the primary component quickly gains forces from the secondary component.

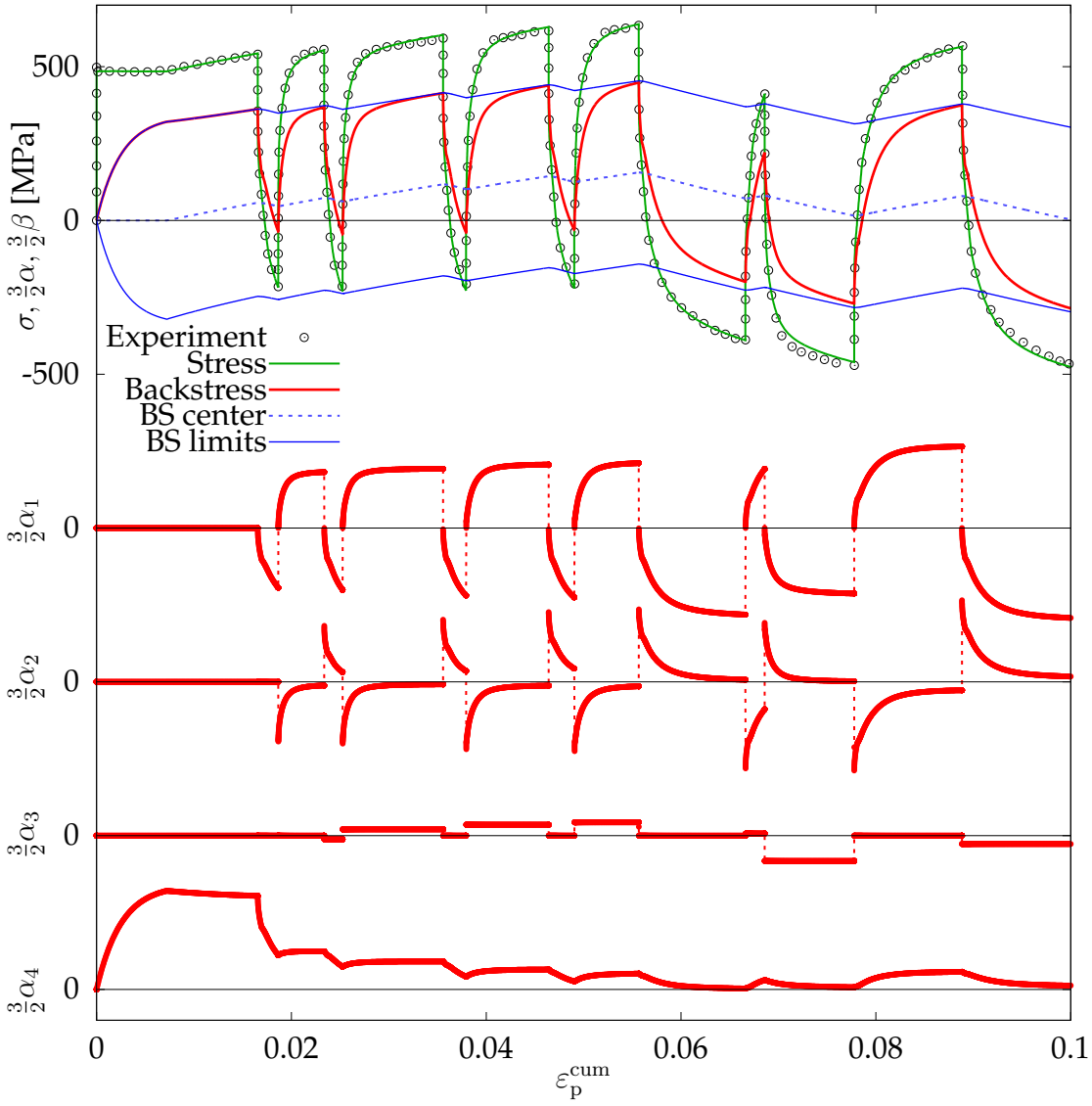

FIGURE 5.7: Backstress components during cyclic loading.

5.2 New decoupled bounding surface model

The latest model features an independently evolved distortion intensity c and therefore it may have a simplified yield condition as

$$\mathrm{f}(\boldsymbol{\sigma}) = \sqrt{\tfrac{3}{2}\left[1 - c\left(\mathbf{n}_\mathrm{r} : \mathbf{z}\right)\right]}\,\|\mathbf{b}\| - k = 0, \tag{5.55}$$

where

$$\mathbf{b} = \left(1 - c_\mathrm{cr}\,\|\mathbf{z}\|\right)(\mathbf{s} - \boldsymbol{\alpha}) + \left[\frac{1}{\sqrt{1 - \|c\mathbf{z}\|}} + c_\mathrm{cr}\,\|\mathbf{z}\| - 1\right]\mathbf{n}_\mathrm{z}\mathbf{n}_\mathrm{z} : (\mathbf{s} - \boldsymbol{\alpha}), \tag{5.56}$$

and where $\mathbf{z}$ undergoes complex evolution in non-proportional loading trajectories while maintaining

$$\|\mathbf{z}\| \leq 1. \tag{5.57}$$

Variable c undergoes complex evolution in positive and negative region. A necessary condition for its evolution law is enforced by the square root as

$$|c| < 1. \tag{5.58}$$

Later, the convexity condition will further tighten this limit. An original idea of this model is to define the oriented axis of the YS not by a single tensor $\mathbf{z}$, but as the multiplication $c\mathbf{z}$.

5.2.1 Yield function gradients

All necessary derivatives are included in Appendix A. They are a bit simpler compared to Model 1, however there are two additional derivatives due to c and $\mathbf{z}$ being evolving parameters.

$$\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} = \|\mathbf{b}\|\,\frac{c}{\sqrt{\frac{8}{3}\left[1 - c\left(\mathbf{n}_\mathrm{r} : \mathbf{z}\right)\right]}}\,\frac{\mathbf{n}_\mathrm{r} : \mathbf{z}\mathbf{n}_\mathrm{r} - \mathbf{z}}{\|\mathbf{s} - \boldsymbol{\alpha}\|} + \sqrt{\tfrac{3}{2}\left[1 - c\left(\mathbf{n}_\mathrm{r} : \mathbf{z}\right)\right]}\,\frac{\mathbf{b}}{\|\mathbf{b}\|} : \frac{\partial \mathbf{b}}{\partial \boldsymbol{\sigma}}, \tag{5.59}$$

where

$$\frac{\partial \mathbf{b}}{\partial \boldsymbol{\sigma}} = \left(1 - c_\mathrm{cr}\,\|\mathbf{z}\|\right)\mathbf{I}^D + \left(\frac{1}{\sqrt{1 - \|c\mathbf{z}\|}} + c_\mathrm{cr}\,\|\mathbf{z}\| - 1\right)\mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z}. \tag{5.60}$$

5.2.2 Internal evolution rules

The novelty of this particular model comes from observations of experiments in [16], where distinct cross effect was measured even after unloading. When preloaded in the opposite direction, it is only expected that such cross effect remains present while the directional part dealing with frontal apex curvature switches sides. That means the material during cyclic loading maintains a preferred direction, albeit with alternating definition. To deal with this, the main axis of the Yield Surface is therefore redefined as a multiplication $c\mathbf{z}$. Its evolution is quite complex. The norm $\|\mathbf{z}\|$ represents a potential for distortion. When $\mathbf{z}$ changes direction, it loses its size and does not allow pronounced distortion. This should help with reorientation, as the rotation of distorted shape is prevented. The evolution of tensor $\mathbf{z}$ consists of growing in the actual direction when loading is present in this or opposite direction. The second term acts as desaturation. When internal structure is disturbed by transverse loading, $\mathbf{z}$ rapidly diminishes to allow rotation to the new direction of loading. When $\mathbf{z}$ is too small to define, the third term initiates growth in the actual direction of radial stress.

$$\dot{\mathbf{z}} = \underbrace{\rho_1 \lambda \left(1 - \|\mathbf{z}\|\right) \mathbf{z}}_{\text{saturation}} - \underbrace{\rho_2 \lambda \left(1 - \left(\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z}\right)^2\right) \mathbf{z}}_{\text{desaturation with reorientation}} + \underbrace{\rho_3 \lambda \left(1 - \|\mathbf{z}\|\right)^2 \operatorname{sgn}\left(\mathbf{n}_\mathrm{r} : \mathbf{z}\right) \mathbf{n}_\mathrm{r}}_{\text{initiation in a new direction}} . \tag{5.61}$$

Distortion variable evolves according to the following rule:

$$\dot{c} = \lambda \bar{c} = c_1 \lambda \left(c_2 \left(\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z}\right) - c\right) \left|\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z}\right| \left(|c| + c_0\right) . \tag{5.62}$$

When cyclic loading is applied, tensor $\mathbf{z}$ saturates in one of its defining direction. Parameter c alternates from negative to positive value and defines the orientation of the distortion of the Yield Surface. At any time, the sign of both variables can be simultaneously changed. This is done every time opposite loading direction is introduced. Similarly, the distortional parameter asymptotically approaches alternating limits.

5.2.3 Convexity condition

The Hessian matrix contains several differences but the procedure is identical to the coupled model.

$$\begin{aligned}
\frac{\partial^2 \mathrm{f}}{\partial\boldsymbol{\sigma}\partial\boldsymbol{\sigma}} &= \frac{c}{\sqrt{\frac{8}{3}\left[1-c\left(\mathbf{n}_\mathrm{r}:\mathbf{z}\right)\right]}}\frac{\mathbf{b}}{\|\mathbf{b}\|}:\frac{\partial\mathbf{b}}{\partial\boldsymbol{\sigma}}\otimes\frac{\mathbf{n}_\mathrm{r}:\mathbf{z}\mathbf{n}_\mathrm{r}-\mathbf{z}}{\|\mathbf{s}-\boldsymbol{\alpha}\|}+\\
&+\sqrt{\frac{3}{32}}\frac{c^2\|\mathbf{b}\|}{\sqrt{1-c\left(\mathbf{n}_\mathrm{r}:\mathbf{z}\right)}^3}\frac{\mathbf{n}_\mathrm{r}:\mathbf{z}\mathbf{n}_\mathrm{r}-\mathbf{z}}{\|\mathbf{s}-\boldsymbol{\alpha}\|}\otimes\frac{\mathbf{z}-\mathbf{n}_\mathrm{r}\left(\mathbf{n}_\mathrm{r}:\mathbf{z}\right)}{\|\mathbf{s}-\boldsymbol{\alpha}\|}+\\
&+\frac{c\|\mathbf{b}\|}{\sqrt{\frac{8}{3}\left[1-c\left(\mathbf{n}_\mathrm{r}:\mathbf{z}\right)\right]}}\cdot\frac{\left(\mathbf{I}^D-3\mathbf{n}_\mathrm{r}\otimes\mathbf{n}_\mathrm{r}\right)\left(\mathbf{n}_\mathrm{r}:\mathbf{z}\right)+\mathbf{z}\otimes\mathbf{n}_\mathrm{r}+\mathbf{n}_\mathrm{r}\otimes\mathbf{z}}{\|\mathbf{s}-\boldsymbol{\alpha}\|^2}+\\
&+\left[\frac{c}{\sqrt{\frac{8}{3}\left[1-c\left(\mathbf{n}_\mathrm{r}:\mathbf{z}\right)\right]}}\frac{\mathbf{n}_\mathrm{r}:\mathbf{z}\mathbf{n}_\mathrm{r}-\mathbf{z}}{\|\mathbf{s}-\boldsymbol{\alpha}\|}\otimes\frac{\mathbf{b}}{\|\mathbf{b}\|}+\right.\\
&\left.+\sqrt{\frac{3}{2}\left[1-c\left(\mathbf{n}_\mathrm{r}:\mathbf{z}\right)\right]}\frac{\partial\mathbf{b}}{\partial\boldsymbol{\sigma}}:\frac{\mathbf{I}\left(\mathbf{b}:\mathbf{b}\right)-\mathbf{b}\otimes\mathbf{b}}{\|\mathbf{b}\|^3}\right]:\\
&:\left[\left(1-c_\mathrm{cr}\|\mathbf{z}\|\right)\mathbf{I}^D+\left(\frac{1}{\sqrt{1-\|c\mathbf{z}\|}}+c_\mathrm{cr}\|\mathbf{z}\|-1\right)\mathbf{n}_\mathrm{z}\otimes\mathbf{n}_\mathrm{z}\right],
\end{aligned}\tag{5.63}$$

where Eqs. (5.12) and (5.60) are used. Eq. (5.36) reaches a fully scalar form as

$$\begin{aligned}
p\boldsymbol{\phi}:\mathbf{H}\boldsymbol{\phi} &= 2cx\left(1-cx\right)\left(g^2+2ghx^2+h^2x^2\right)\cdot\\
&\cdot\left[g^2\left(1-y^2\right)+\left(h^2+2gh\right)\left(4xyz-2z^2+x^2-3x^2y^2\right)\right]+\\
&+4g^2\left(1-cx\right)^2\left[g^2\left(1-y^2\right)+h\left(h+2g\right)\left(x^2+z^2-2xyz\right)\right]-\\
&+c^2\left(g^2+2ghx^2+h^2x^2\right)^2\left(xy-z\right)^2\geq 0,
\end{aligned}\tag{5.64}$$

where $x=(\mathbf{n}_\mathrm{r}:\mathbf{z})$, $y=(\mathbf{n}_\mathrm{r}:\boldsymbol{\phi})$, $z=(\mathbf{z}:\boldsymbol{\phi})$, $g=(1-c_\mathrm{cr})$ and $h=\left[\frac{1}{\sqrt{1-|c|}}+c_\mathrm{cr}-1\right]$ and finally an always positive $p=4\sqrt{\frac{2}{3}}\|\mathbf{s}-\boldsymbol{\alpha}\|\left\|g\mathbf{n}_\mathrm{r}+h\mathbf{z}x\right\|^3\left(1-cx\right)^{3/2}\geq 0$

It is apparent from the Yield function (5.55), that it is possible to state $\|\mathbf{z}\|=1$, as it is always multiplied by either c, or c_cr.

Convexity at a selected point

A special case of the Hessian for back side of the yield surface

$$\mathbf{H}=\frac{\partial^2\mathrm{f}}{\partial\boldsymbol{\sigma}\partial\boldsymbol{\sigma}}=\left[\frac{-c+2\left(1-c^2\right)\left(1-c_\mathrm{cr}\right)^2}{s\sqrt{\frac{8}{3}\left(1-c^2\right)}}\right]\left(\mathbf{I}^D-\mathbf{n}\otimes\mathbf{n}\right).\tag{5.65}$$

leads to the quadratic equation

$$2(1-c_\mathrm{cr})^2c^2+c-2(1-c_\mathrm{cr})^2=0.\tag{5.66}$$

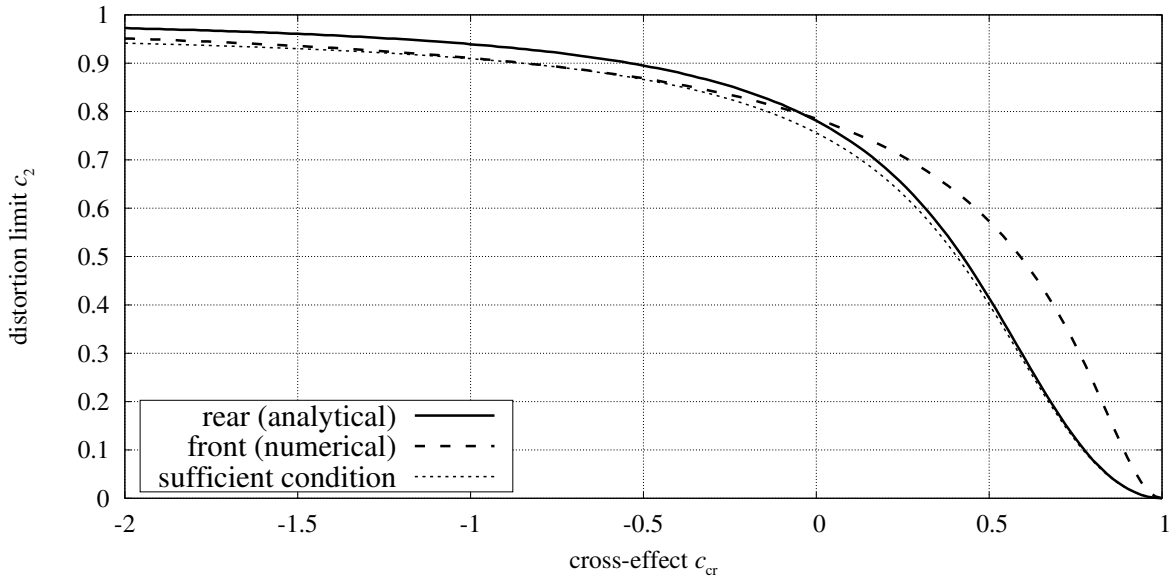

FIGURE 5.8: Parameter limits for the convexity condition

and a similar function is reached as

$$c_{\text{lim}}^{\text{axial}} = \frac{-1 + \sqrt{1 + 16\,(1 - c_{\text{cr}})^4}}{4(1 - c_{\text{cr}})^2}. \tag{5.67}$$

Global convexity

The Yield function is at the convexity limit, if for the largest possible c holds

$$\inf\,(p\boldsymbol{\phi} : \mathbf{H}\boldsymbol{\phi}) = 0. \tag{5.68}$$

Numerical search The search for c_{lim} moves again to the meridian plane only. The resulting function $c_{\text{lim}} = c_{\text{lim}}(c_{\text{cr}})$ is plotted in Fig. 5.8. The analytical solution for a sufficient condition is virtually identical and sets the limit for the asymptotic limit of c as

$$c_2 = c_{\text{lim}}^1 = 0.968\, c_{\text{lim}}^{\text{axial}}, \tag{5.69}$$

as seen in Figure 5.8.

5.2.4 Model calibration

A simple calibration from proportional loading in Figure 5.2 is performed to verify the model's behavior is as intended.

Param.	Value	Units	Param.	Value	Units
E	185000	MPa	K_0	0.01	MPa
ν	0.29		K_1	123.65	MPa
k_0	485	MPa	K_2	1.0049×10^{-3}	MPa^{-1}
κ_1	53242	MPa	b_1	6469.9	MPa
κ_2	8.3244×10^{-3}	MPa^{-1}	b_2	2.1986×10^{-3}	MPa^{-1}
κ_3	21554	MPa	ϵ_{plat}	7.2×10^{-3}	
κ_4	5.063×10^{-3}	MPa^{-1}	c_0	0.01	
h_1	36.4465	-	c_1	100	
h_2	3263561000	MPa^3	c_2	0.6	
h_3	0.043278	MPa^{-1}	c_{cr}	0.3	
h_4	0.5	-	ρ_1	100	
h_5	0.5	-	ρ_2	2000	
ε	0.01	MPa, MPa^2	ρ_3	3000	

TABLE 5.2: Calibration of decoupled model from proportional test after [3]

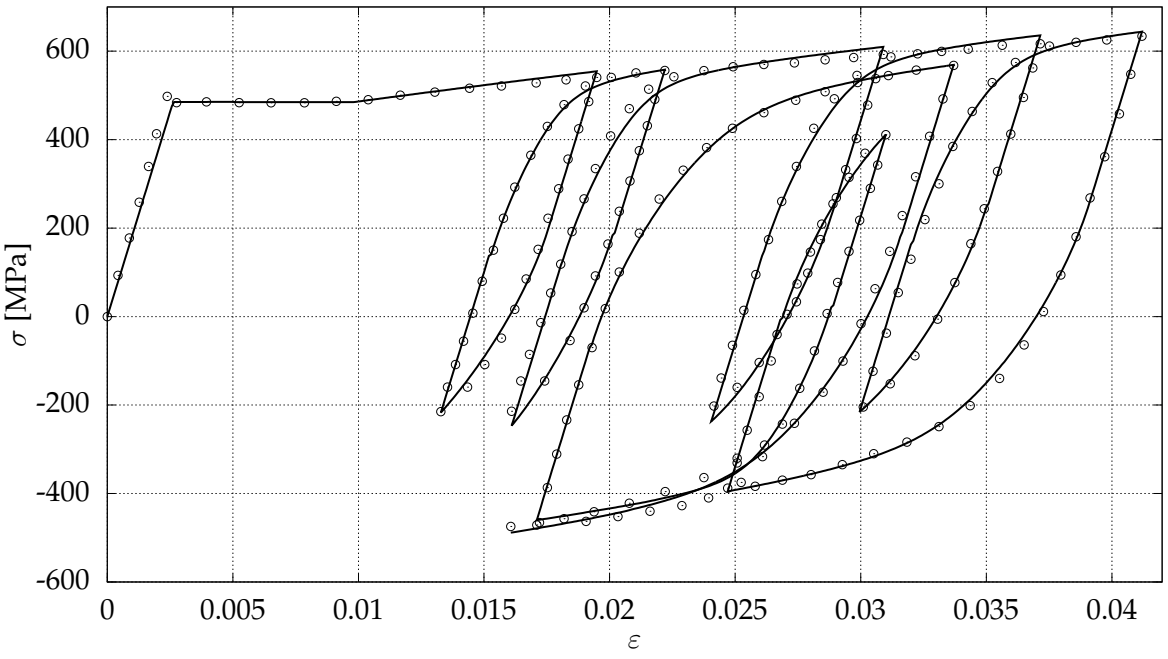

FIGURE 5.9: Demonstration of the uncoupled model in random uniaxial cyclic loading calibrated to experimental data in [3].

A simple test is performed to demonstrate the model's behavior in ratcheting with two different settings of the levels of the alternating stress.

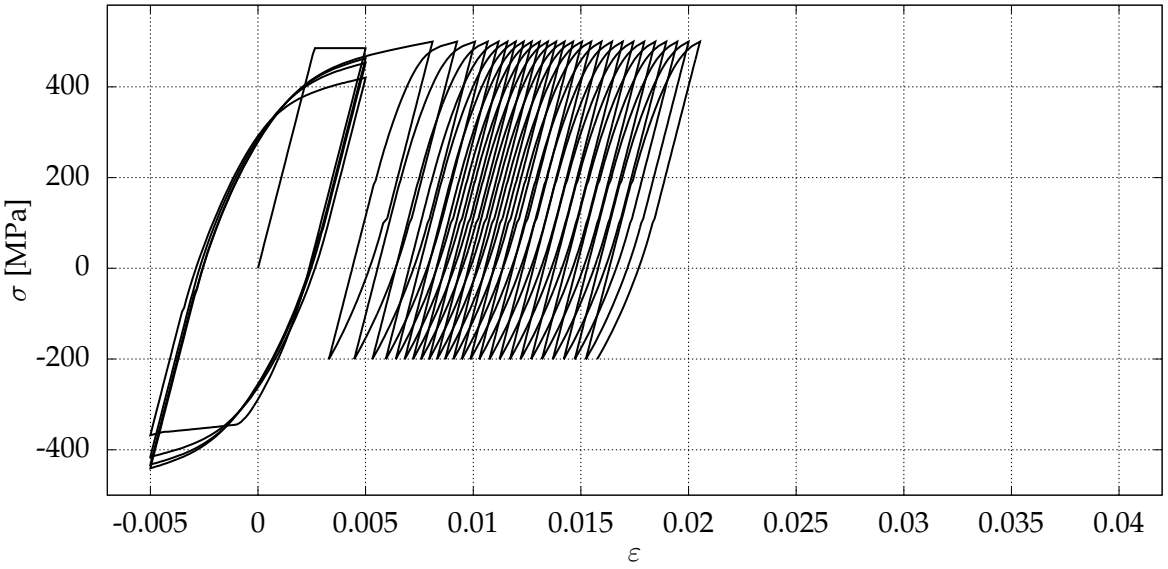

FIGURE 5.10: Simulation of ratcheting after depletion of plateau region.

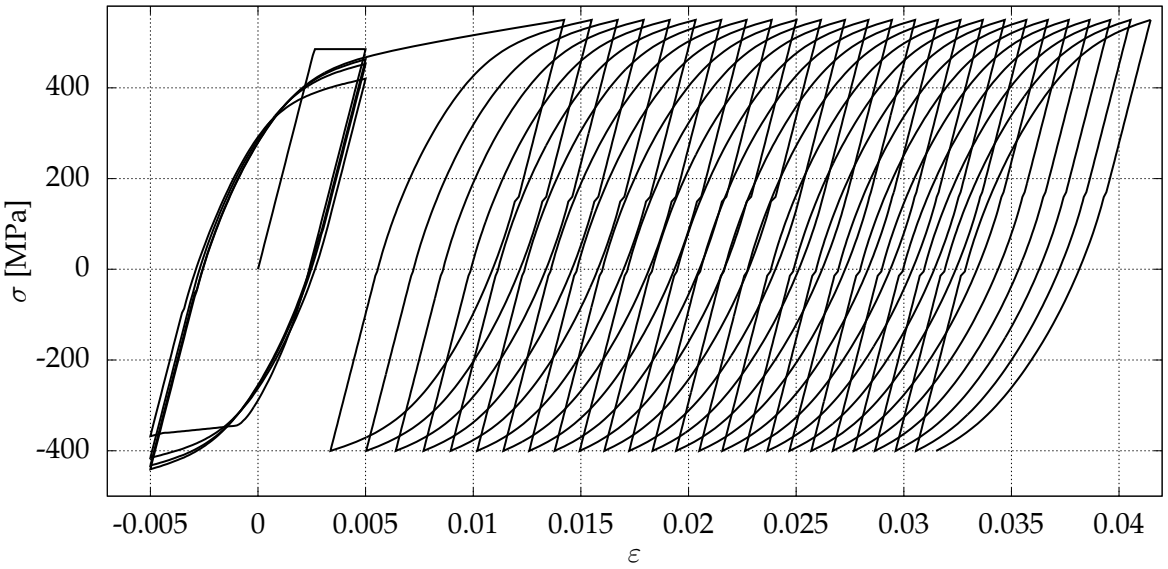

FIGURE 5.11: Simulation of ratcheting for greater levels of stress.

5.3 Implementation

Due to highly volatile nature of the kinematic rule with plastic modulus changing by several orders of magnitude, a slight modification of the integration procedure has been suggested. Instead of calculating the plastic multiplicator from Eq. (4.6) with eventual overshooting requiring recalculation, the plastic multiplier is instead proposed at the beginning of the integration step and checked against a series of

exact or conservative limits. This approach enables direct control of the integration, activating discrete changes in the model at the right time, correctly at the end of the integration step. The main function returning stress response of the material is in Algorithm 5.1.

It contains several inner functions. The Yield Function evaluates either Eq. (5.4) or Eq. (5.55) as yield condition and is called several times during the search for plastic part and during stress corrections. Algorithm Flow Rule calculates the necessary derivatives for the consistency equation. Algorithm 5.2 handles logistics of the backstress components. Based on the rates of inner variables and their respective limits, the inner variables have assigned suggested values of the plastic multiplicator. Its smallest permitted value

$$\lambda = \min\left\{\lambda_\alpha, \lambda_\beta, \lambda_c, \lambda_k, \lambda_K, \lambda_{\text{plat}}\right\}. \tag{5.70}$$

is then used to calculate the new state of the material.

5.3.1 Integration step optimization

The permissible size of the integration step is set to prevent overshooting the asymptotic limit of respective internal variable. Parameter $\texttt{NINT} \geq 10$ further controls subdivision to maintain desired numerical precision. `NINT` represents the effect of each variable on the overall stress response in relation to the initial yield stress k_0. The limits induced by isotropic parameters are simple to deduce. The isotropic hardening rule creates two limitations as

$$\lambda_k = \min\left\{\frac{1/\kappa_2 - k}{3\bar{k}}; \frac{k_0}{\texttt{NINT}\left|\bar{k}\right|}\right\} \quad \text{for} \quad \epsilon_{\text{p}}^{\text{cum}} < \epsilon_{\text{plat}} \tag{5.71}$$

and

$$\lambda_k = \min\left\{\frac{1/\kappa_4 - k}{3\bar{k}}; \frac{k_0}{\texttt{NINT}\left|\bar{k}\right|}\right\} \quad \text{for} \quad \epsilon_{\text{p}}^{\text{cum}} \geq \epsilon_{\text{plat}}. \tag{5.72}$$

By its definition, the Bounding Surface moves too slow to have relevant limitations either for its isotropic or kinematic rule.

$$\lambda_K = \min\left\{\frac{1/K_2 - K}{3\bar{K}}; \frac{k_0}{\texttt{NINT}\left|\bar{K}\right|}\right\} \tag{5.73}$$

By principle, the Bounding Surface has to move much more slowly compared to the backstress. Therefore, it does not need to be checked for overshooting. The maximum plastic multiplier for the overall backstress kinematic rule is set with respect

to the Bounding Surface as

$$\lambda_{\boldsymbol{\alpha}} = \min\left\{ \frac{-\left(\bar{\mathbf{z}}:\mathbf{z} - \frac{2}{3}\bar{K}K\right) + \sqrt{D}}{3\left(\bar{\mathbf{z}}:\bar{\mathbf{z}} - \frac{2}{3}\bar{K}^2\right)}; \sqrt{\tfrac{2}{3}}\frac{k_0}{\|\bar{\boldsymbol{\alpha}}\|\,\mathtt{NINT}} \right\} \tag{5.74}$$

where the simplified discriminant, using Eq. (5.2), yields

$$D = \left(\bar{\mathbf{z}}:\mathbf{z} - \tfrac{2}{3}\bar{K}K\right)^2 - \left(\bar{\mathbf{z}}:\bar{\mathbf{z}} - \tfrac{2}{3}\bar{K}^2\right)\left(\mathbf{z}:\mathbf{z} - \tfrac{2}{3}K^2\right). \tag{5.75}$$

Each of the backstress components need to fulfill additional conditions when their depletion is imminent, as

$$\lambda_{\boldsymbol{\alpha}i} = \frac{\boldsymbol{\alpha}_i}{3\mathbf{n}_{\boldsymbol{\alpha}i}:\bar{\boldsymbol{\alpha}}_i} \tag{5.76}$$

The limit plastic multiplier for the evolving distortion parameter is applicable for the second model only. As it has two mirrored asymptotic limits, the active one is chosen by the use of $(\mathbf{n}_{\mathrm{r}}:\mathbf{n}_{\mathrm{z}})$ as

$$\lambda_c = \min\left\{ \frac{c_2\left(\mathbf{n}_{\mathrm{r}}:\mathbf{n}_{\mathrm{z}}\right) - c}{3\bar{c}}; \frac{1}{\mathtt{NINT}\,|\bar{c}|} \right\}. \tag{5.77}$$

1. Compute elastic trial stress:
2. $\Delta\boldsymbol{\sigma}^{\text{trial}} = \mathbf{C}\Delta\boldsymbol{\epsilon}^{\text{trial}}$
3. call YIELD FUNCTION
4. **IF** $\mathrm{f} \leq tol$
   $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} + \Delta\boldsymbol{\sigma}^{\text{trial}}$

   ELSE
   (a) $g \leftarrow 0$
   (b) Evaluate plastic part ratio g of the trial stress:
   Loop while $\mathrm{f} > \texttt{tol}$
   i. $\Delta g = f \cdot \left[\dfrac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \Delta\boldsymbol{\sigma}^{\text{trial}}\right]^{-1}$
   ii. $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} - \Delta g \Delta\boldsymbol{\sigma}^{\text{trial}}$
   iii. $g \leftarrow g + \Delta g$
   iv. call YIELD FUNCTION
   (c) Evolve internal variables:
   Loop while $g > 0$
   i. Evaluate flow direction:
   call FLOW RULE
   ii. Estimate integration step:
   Evaluate $\lambda = \min\{\lambda_\alpha, \lambda_\beta, \lambda_c, \lambda_k, \lambda_K\}$
   iii. $\Delta g = \lambda \dfrac{K_{\mathrm{p}} + \frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \mathbf{C} \frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}}}{\frac{\partial \mathrm{f}}{\partial \boldsymbol{\sigma}} : \Delta\boldsymbol{\sigma}^{\text{trial}}}$
   iv. **IF** $\Delta g > g$
   $\lambda \leftarrow \lambda \frac{g}{\Delta g}$
   $\Delta g \leftarrow g$
   END IF
   v. call EVOLUTION
   vi. $g \leftarrow g - \Delta g$

   END IF

ALGORITHM 5.1: Main algorithm for stress response

1. Collapse empty components:

   IF $\|\boldsymbol{\alpha}_2\| < \texttt{tol}$

   (a) $\boldsymbol{\alpha}_2 \leftarrow \boldsymbol{\alpha}_2 + \boldsymbol{\alpha}_3$

   (b) $\boldsymbol{\alpha}_3 \leftarrow \mathbf{0}$

   END IF

2. Fuse similarly oriented components:

   IF $\mathbf{n}_{\boldsymbol{\alpha}1} : \mathbf{n}_{\boldsymbol{\alpha}2} > 0.95$

   (a) $\boldsymbol{\alpha}_1 \leftarrow \boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2$

   (b) $\boldsymbol{\alpha}_2 \leftarrow \boldsymbol{\alpha}_3$

   (c) $\boldsymbol{\alpha}_3 \leftarrow \mathbf{0}$

   END IF

   IF $\mathbf{n}_{\boldsymbol{\alpha}2} : \mathbf{n}_{\boldsymbol{\alpha}3} > 0.95$

   (a) $\boldsymbol{\alpha}_2 \leftarrow \boldsymbol{\alpha}_2 + \boldsymbol{\alpha}_3$

   (b) $\boldsymbol{\alpha}_3 \leftarrow \mathbf{0}$

   END IF

3. Create new component for new loading direction:

   IF $\mathbf{n}_{\boldsymbol{\alpha}1} : \mathbf{n}_{\boldsymbol{\alpha}\mathrm{B}} < 0.5$

   (a) $\boldsymbol{\alpha}_3 \leftarrow \boldsymbol{\alpha}_2$

   (b) $\boldsymbol{\alpha}_2 \leftarrow \boldsymbol{\alpha}_1$

   (c) $\boldsymbol{\alpha}_1 \leftarrow \mathbf{0}$

   ELSEIF $\mathbf{n}_{\boldsymbol{\alpha}1} : \mathbf{n}_{\boldsymbol{\alpha}\mathrm{B}} < 0.8$

   $p = \langle \exp\left(-p_1(\mathbf{n}_{\boldsymbol{\alpha}1} : \mathbf{n}_{\boldsymbol{\alpha}\mathrm{B}} - 0.8)\right) - 1 \rangle$

   END IF

ALGORITHM 5.2: Backstress component logistic procedure

5.4 Numerical tests

A novel approach accompanying the development of these models is the previously unseen focus on numerical testing from a wide range of perspectives. The value of virtually any internal variable can be meaningfully plotted and its changes observed in various scenarios, bringing new insight into the inner workings of the model or uncovering previously hidden inconsistencies.

5.4.1 Error maps

A helpful and well known way to test a model's implementation is an iso-error map. The models in question produce error fields with much complexity, a contour plot would not be readable. Therefore, a combined plot presented in Fig. 5.12 includes an equi-strain map to give the reader better idea about the scale of the involved deformation. The error, defined by

$$\eta = \frac{\|\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}\|}{k_0} \cdot 100\,[\%]\,, \tag{5.78}$$

is calculated by means of a 10 times finer subincrementation of the integration step. This particular map is important from several perspectives. First, with the setting of $NINT = 50$, the error reaches up to 2%, which is still manageable. Better precision is easily accessible via parameter $NINT$. A single, thin sliver at the very frontal apex reaches error of up to 20%. This is caused by fast behavior of Eq. 5.21 combined with low rate of plastic strain in tangential loading direction. This same behavior causes non-convex equi-strain lines within the first milistrain range, as the Yield Surface rotates fast around β.

Probably the most important notion is the preservation of high error even far away from the initial prestress (black circle). Contrary to simple models, where the error diminishes with the subsiding plastic modulus, in this model, the Bounding Surface continues to harden and maintains the acquired level of error.

A related case in a more general state of stress is in Figure 5.13. The same problem with precison is clearly visible at the frontal apex. A map of the count of numerical steps is in Figure 5.14.

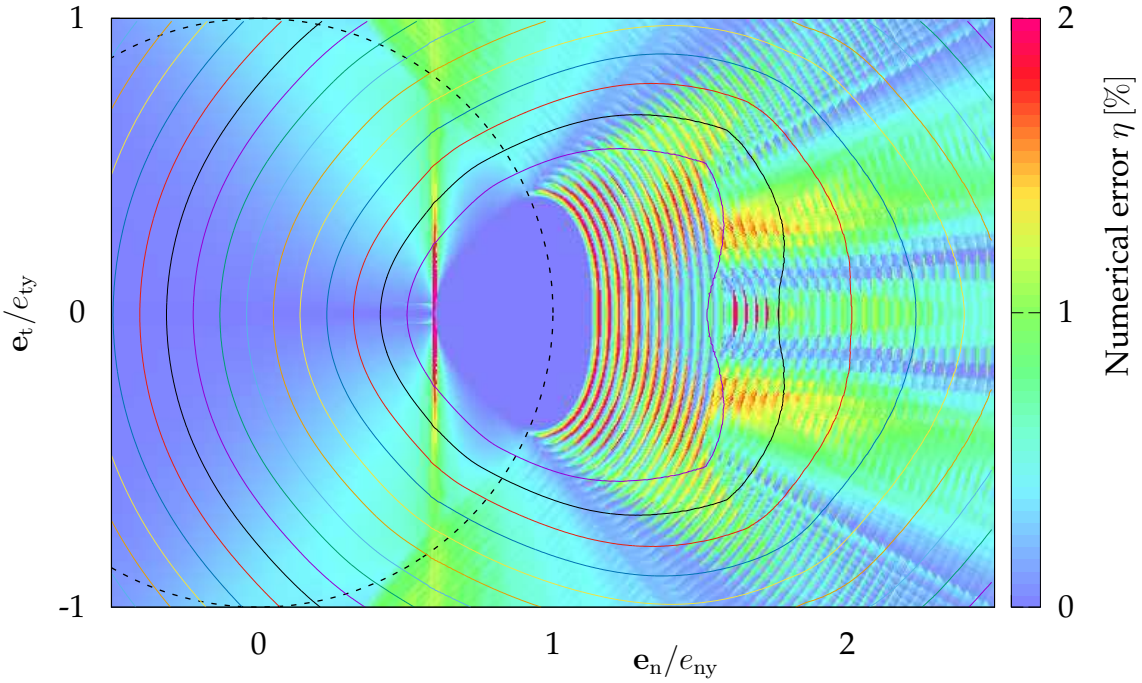

FIGURE 5.12: Combined plot of numerical error and plastic strain in 200 microstrain increments in deviatoric space.

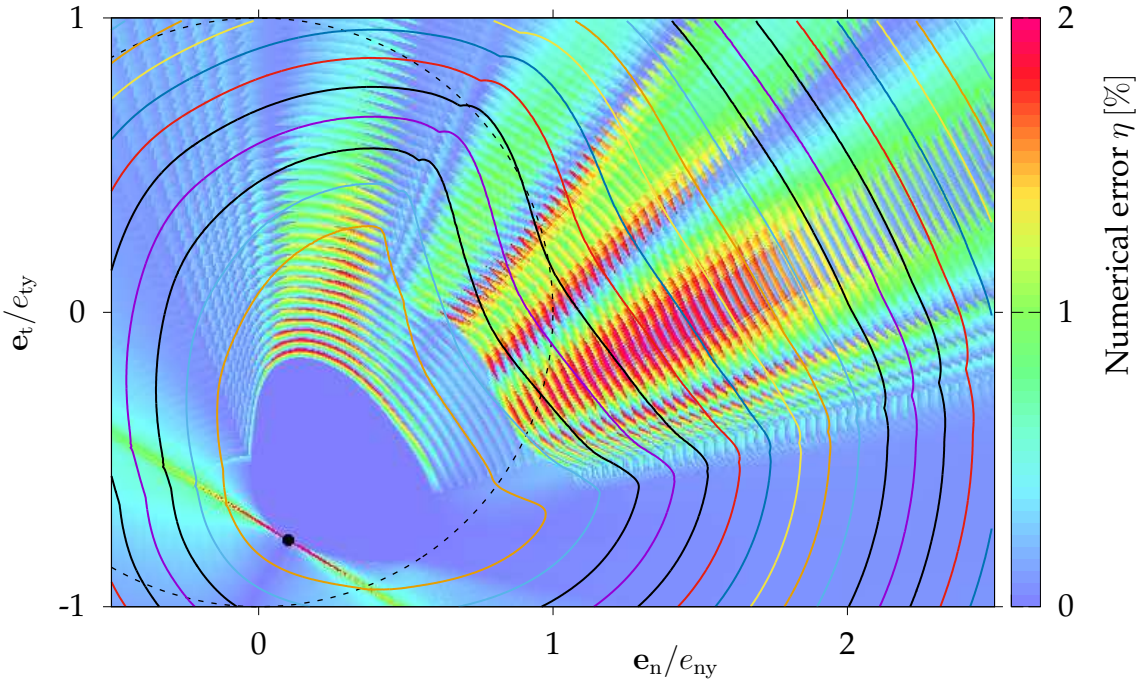

FIGURE 5.13: Combined plot of numerical error and plastic strain in deviatoric space.

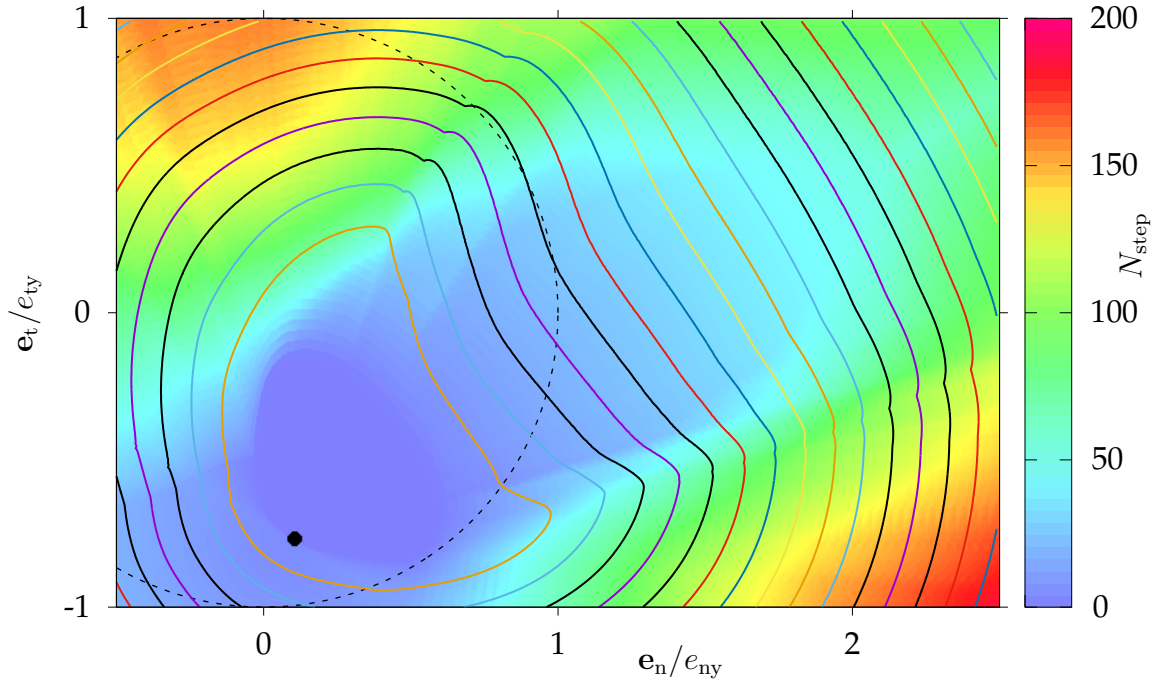

FIGURE 5.14: Combined plot of the number of integration steps and plastic strain in deviatoric space.

5.4.2 Benchmarks

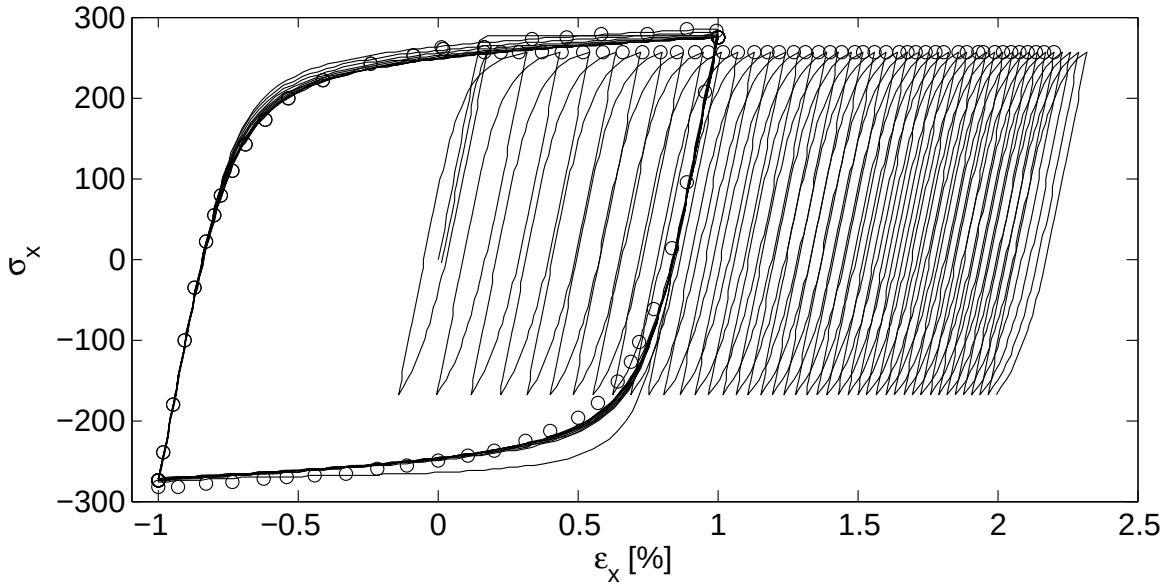

FIGURE 5.15: Demonstration of cyclic ratcheting after 12 cycle stabilization after [13].

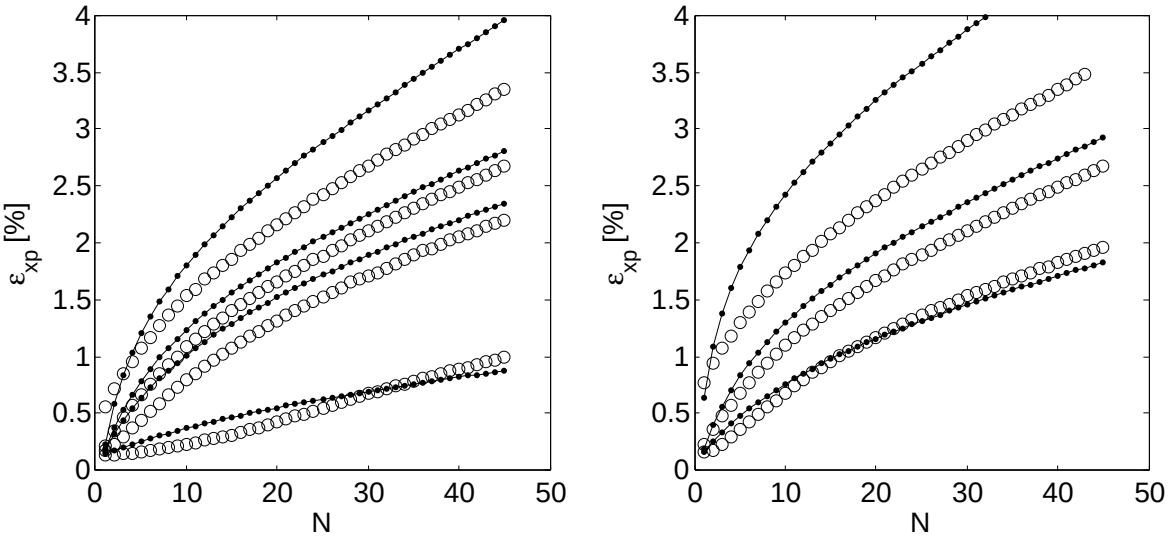

FIGURE 5.16: Simulation of a set of uniaxial ratcheting experiments after [13].

Chapter 6

Theoretical and Practical Outcomes of the Thesis

Regarding the **theoretical outcomes of the thesis**, the two new models serve as a launching pad for research into low-cycle fatigue. Specific nature of experiments has been proposed to help tweak and calibrate these models for expected practical application, focusing on very small deformation in the scope of elasto-plastic transition, for which the internal evolution rules were intended. The simulation of springback is the next proposed step, however, the experimental investigation of sheet metal is much more difficult to master, especially considering the requirements of these particular models. The new experimental kinematic rule opens the possibility to model sequential distortion, the exact observed behavior of metal materials within the realm of very small deformation. Unifying the backstress components with the individual distortion directors would create a logically consistent model.

The **practical outcomes of the thesis** lie in the UMAT subroutines for Abaqus, making these models available to the community of engineers. Changing the evolution rules within the model is very easy, as it does not interfere with derivatives of the Yield Function. Preparing the model for particular practical application is now straight forward process, thanks to many tools developed during the work on this thesis, namely numerical calibration on a library of experiments and visualizing the internal state of the model, energy dissipation and other phenomena in tension-torsion as well as deviatoric stress plane.

Chapter 7

Conclusions and Future Prospectives

From a multitude of tests it is evident that the act of proposing a new advanced model for multiaxial loading has to be accompanied by a deep study in its behavior. Analytical methods, used for testing some very specific conditions, do help eliminate most of the encountered problems. Still, there are great many problems hiding in general application. To search for them, it is necessary to use brute-force approach in a large variety of tests.

The integration speed of the presented models is satisfactory. There are a number of possible ways of optimization. Using osculation hyperspheres for analytical solutions within a small angle at the apex of the surface could reduce the computational costs when calculating the plastic part of the trial stress as well as radial return stress correction. Using a midpoint rule for more precise evolution of internal variables is the most immediate proposition. Smooth positioning of the midpoint can eliminate discontinuities when the system chooses between Euler forward and midpoint rule. Functions with extreme acceleration, like the hardening curve exhibiting smooth elastic-plastic transition, may have an optimal position of the midpoint to get the best precision.

Many unfounded decisions on the kinematic rule were made in the presented model. It is expected these models will continue to evolve when new information about the behavior of the material comes to light. The Institute of Thermomechanics of the CAS recently acquired means of advanced experimental testing, meeting the requirements of effective research into multiaxial ratcheting. One of many questions is whether counteracting and eliminating elongation due to distortion at the front apex is even practical.

7.1 Future ways of distortion generation

After acquiring experience with segmented kinematic rule, new ideas could potentially be applicable. When observing the behavior in Fig. 2.3, one could argue the yield surface behaves as a mesh molded by external forces. When loading occurs,

a simple field of volumetric force would incrementally pronounce the frontal apex and flatten the rear. A similar, more sophisticated approach comprising of controlled movement of up to 30 individual slip systems within the theory of crystal plasticity was proposed in [5], recently also in [23], both for plane stress problems. Expansion to the whole deviatoric space is difficult due to the multiplicative number of systems needed if three new dimensions were added. This approach is therefore unusable due to hardware demands. It is unknown, how this principle holds for more complex loading histories.

A purely phenomenological approach is expected to be more successful in cases, where the trajectory of the operational loading is known and similar tests can be performed for calibration. Unlike a continuous summation of distortions in [40], a sequence of discrete distortions has been recently studied. The new proposed yield function

$$\mathrm{f}\left(\boldsymbol{\sigma}\right) = \sqrt{\tfrac{3}{2}}\left\|\mathbf{b}_N\right\| - k \tag{7.1}$$

employs the modified radial tensor $\mathbf{b}^N$. The initial radial tensor $\mathbf{b}_0 = \mathbf{s} - \boldsymbol{\alpha}$ undergoes N successive transformations as

$$\mathbf{b}_{i+1} = \mathbf{b}_i + c_{i1}\mathbf{n}_{\mathbf{z}i}x_i\left[\frac{1}{1+c_{i2}x_i} + c_{i3}\right],\ i = 0, 1, ..., N-1, \tag{7.2}$$

where $x_i = \mathbf{n}_{\mathbf{z}i} : \mathbf{b}_i$. Each of the transformation is driven by unit-norm director $\mathbf{n}_{zi}$. Three evolved variables c_{1i}, c_{2i} and c_{3i} for every single transformation control the axial distortion at the front and back as well as induced curvature of the frontal apex. The cross effect is controlled only by the isotropic parameter k. To demonstrate its capabilities, the yield function's internal variables have been calibrated to fit the previously mentioned series of measured yield surfaces, see Figure 7.1. The distortion is induced by a hyperbolic function enabling high relative compression of the rear. Rather than choosing a two-value elongation of front and rear separately, this continuous function maintains first order continuity when applied on an arbitrary shape. Still, the results for more complex loading histories are uncertain. The combined distortion of non-orthogonal directors (black line on the right) was achieved by using incorrect choice of directors. To effectively solve this problem, much more data need to be measured.

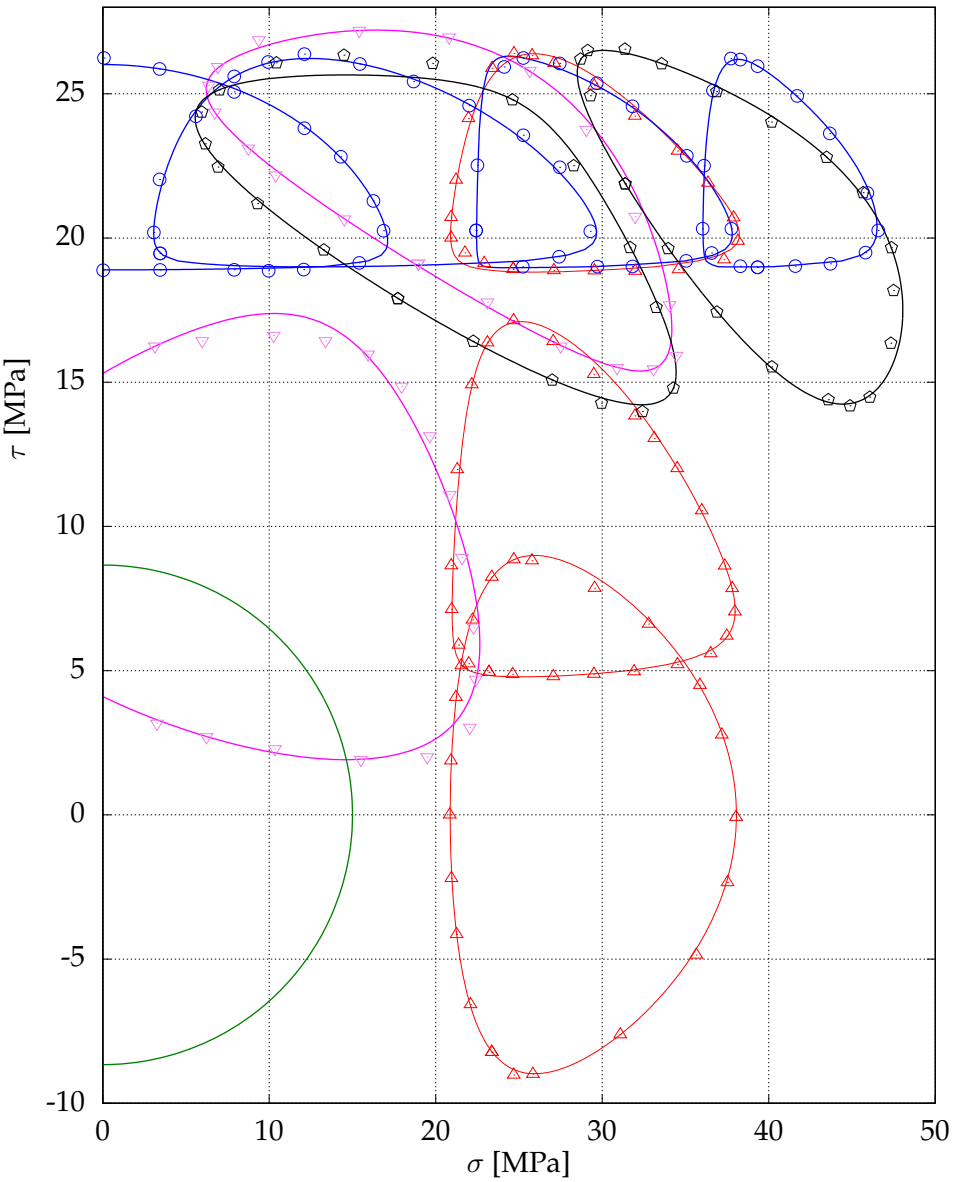

FIGURE 7.1: Demonstration of sequential multidirectional distortion.

The main problem of this model is the discrete series of directors needed for the distortion. Their movement, their introduction and disappearance has to be smooth, so that the stress response of such a model will be continuous. Only initial steps have been made to remedy this problem.

Author's References

[A1] MAREK, R., PLEŠEK, J., HRUBÝ, Z., PARMA, S., FEIGENBAUM, H., AND DAFALIAS, Y. Numerical implementation of a model with directional distortional hardening. *Journal of Engineering Mechanics 141*, 12 (2015), pp. 12–21.

[A2] PARMA, S., PLEŠEK, J., MAREK, R., HRUBÝ, Z., FEIGENBAUM, H. P., AND DAFALIAS, Y. F. Calibration of a simple directional distortional hardening model for metal plasticity. *International Journal of Solids and Structures 143* (2018), pp. 113–124.

[A3] WELLING, C., MAREK, R., FEIGENBAUM, H., DAFALIAS, Y., PLEŠEK, J., HRUBÝ, Z., AND PARMA, S. Numerical convergence in simulations of multiaxial ratcheting with directional distortional hardening. *International Journal of Solids and Structures* (2017), pp. 105–121.

Bibliography

[1] CORMEAU, I. Numerical stability in quasi-static elasto/visco-plasticity. *International Journal for Numerical Methods in Engineering 9*, 1 (1975), pp. 109–127.

[2] CORONA, E., HASSAN, T., AND KYRIAKIDES, S. On the performance of kinematic hardening rules in predicting a class of biaxial ratcheting histories. *International Journal of Plasticity 12*, 1 (1996), pp. 117–145.

[3] DAFALIAS, Y. F., AND POPOV, E. P. A model of nonlinearly hardening materials for complex loading. *Acta Mechanica 21*, 3 (1975), pp. 173–192.

[4] DAFALIAS, Y. F., AND POPOV, E. P. Plastic internal variables formalism of cyclic plasticity. *ASME Journal of Applied Mechanics 43*, 4 (1976), pp. 645–651.

[5] DLUZEWSKI, P. H. The effect of slips on the anisotropic behaviour of polycrystalline aluminium at elevated temperature. *Textures and Microstructures 19*, 1–2 (1992), pp. 81–99.

[6] FEIGENBAUM, H. P., AND DAFALIAS, Y. F. Directional distortional hardening in metal plasticity within thermodynamics. *International Journal of Solids and Structures 44*, 22 (2007), pp. 7526–7542.

[7] FEIGENBAUM, H. P., AND DAFALIAS, Y. F. Simple model for directional distortional hardening in metal plasticity within thermodynamics. *Journal of Engineering Mechanics 134*, 9 (2008), pp. 730–738.

[8] FEIGENBAUM, H. P., DUGDALE, J., DAFALIAS, Y. F., KOUROUSIS, K. I., AND PLESEK, J. Multiaxial ratcheting with advanced kinematic and directional distortional hardening rules. *International Journal of Solids and Structures 49*, 22 (2012), pp. 3063–3076.

[9] FRANÇOIS, M. A plasticity model with yield surface distortion for non proportional loading. *International Journal of Plasticity 17*, 5 (2001), pp. 703–717.

[10] FREUND, M., SHUTOV, A., AND IHLEMANN, J. Simulation of distortional hardening by generalizing a uniaxial model of finite strain viscoplasticity. *International Journal of Plasticity 36* (2012), pp. 113–129.

[11] GEAR, C. W. *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs, New Jersey: Prentice-Hall, 1971.

[12] HASSAN, T., CORONA, E., AND KYRIAKIDES, S. Ratcheting in cyclic plasticity, part ii: Multiaxial behavior. *International Journal of Plasticity 8*, 2 (1992), pp. 117–146.

[13] HASSAN, T., AND KYRIAKIDES, S. Ratcheting in cyclic plasticity, part i: Uniaxial behavior. *International Journal of Plasticity 8*, 1 (1992), pp. 91–116.

[14] HRUBÝ, Z. *Directional Distortional Hardening–Models' Parameters Calibration and FE-code Implementation*. PhD thesis, CTU Prague, Faculty of Mechanical Engineering, 2012.

[15] KHAN, A. S., KAZMI, R., PANDEY, A., AND STOUGHTON, T. Evolution of subsequent yield surfaces and elastic constants with finite plastic deformation. Part-I: A very low work hardening aluminum alloy (Al6061-T6511). *International Journal of Plasticity 25*, 9 (2009), pp. 1611–1625. Exploring New Horizons of Metal Forming Research.

[16] KHAN, A. S., PANDEY, A., AND STOUGHTON, T. Evolution of subsequent yield surfaces and elastic constants with finite plastic deformation. Part II: A very high work hardening aluminum alloy (annealed 1100 Al). *International Journal of Plasticity 26*, 10 (2010), pp. 1421–1431.

[17] KHAN, A. S., PANDEY, A., AND STOUGHTON, T. Evolution of subsequent yield surfaces and elastic constants with finite plastic deformation. part iii: Yield surface in tension–tension stress space (al 6061-t 6511 and annealed 1100 al). *International Journal of Plasticity 26*, 10 (2010), pp. 1432–1441.

[18] KURTYKA, T., AND ŻYCZKOWSKI, M. A geometric description of distorsional plastic hardening of deviatoric materials. *Archives of Mechanics 37*, 4–5 (1985), pp. 383–395.

[19] KURTYKA, T., AND ŻYCZKOWSKI, M. Evolution equations for distorstional plastic hardening. *International Journal of Plasticity 12*, 2 (1996), pp. 191–213.

[20] LAMBERT, J. D. *Computational Methods in Ordinary Differential Equations*. Wiley, London, 1973.

[21] LAPIDUS, L., AND SEINFELD, J. H. *Numerical Solution of Ordinary Differential Equations*. Mathematics in Science and Engineering. A Series of Monographs and Textbooks. Academic Press, 1971.

[22] LI-WEI, L., AND HONG-KI, H. A description of three-dimensional yield surfaces by cubic polynomials. *Journal of Engineering Mechanics 143* (2017), pp. 86–98.

[23] LIU, F., FU, Q., CHEN, C., AND LIANG, N. An elasto-plastic damage constitutive theory and its prediction of evolution of subsequent yield surfaces and elastic constants. *International Journal of Plasticity 27*, 9 (2011), pp. 1355–1383.

[24] LIU, G.-L., HUANG, S.-H., SHI, C.-S., ZENG, B., ZHANG, K.-S., AND ZHONG, X.-C. Experimental investigations on subsequent yield surface of pure copper by single-sample and multi-sample methods under various predeformation. *Materials 11* (2018), pp. 277–272.

[25] MAHAN, M., DAFALIAS, Y. F., TAIEBAT, M., HEO, Y., AND KUNNATH, S. K. Sanisteel: Simple anisotropic steel plasticity model. *Journal of Structural Engineering 137*, 2 (2011), pp. 185–194.

[26] MATTHIES, H., AND STRANG, G. The solution of nonlinear finite element equations. *International journal for numerical methods in engineering 14*, 11 (1979), pp. 1613–1626.

[27] OHASHI, Y., KAWASHIMA, K., AND YOKOCHI, T. Anisotropy due to plastic deformation of initially isotropic mild steel and its analytical formulation. *Journal of the Mechanics and Physics of Solids 23*, 4 (1975), pp. 277–294.

[28] ORTIZ, M., AND POPOV, E. P. Distortional hardening rules for metal plasticity. *Journal of Engineering Mechanics 109*, 4 (1983), pp. 1042–1057.

[29] ORTIZ, M., AND POPOV, E. P. Accuracy and stability of integration algorithms for elastoplastic constitutive relations. *International Journal for Numerical Methods in Engineering 21*, 9 (1985), pp. 1561–1576.

[30] PHILLIPS, A., AND TANG, J.-L. The effect of loading path on the yield surface at elevated temperatures. *International Journal of Solids and Structures 8*, 4 (1972), pp. 463–474.

[31] PHILLIPS, A., TANG, J.-L., AND RICCIUTI, M. Some new observations on yield surfaces. *Acta Mechanica 20*, 1–2 (1974), pp. 23–39.

[32] PLEŠEK, J., FEIGENBAUM, H. P., AND DAFALIAS, Y. F. Convexity of yield surface with directional distortional hardening rules. *Journal of Engineering Mechanics 136*, 4 (2010), pp. 477–484.

[33] PLEŠEK, J., AND KŘÍSTEK, A. Assessments of methods for locating the point of initial yield. *Computer Methods in Applied Mechanics and Engineering 141*, 3–4 (1997), pp. 389–397.

[34] PRAGER, W. Recent developments in the mathematical theory of plasticity. *Journal of Applied Physics 20*, 3 (1949), pp. 235–241.

[35] ROKHGIREH, H., NAYEBI, A., AND CHABOCHE, J. Application of a new distortional yield surface model in cyclic uniaxial and multiaxial loading. *International Journal of Solids and Structures 110–111* (2017), pp. 219–238.

[36] SHUTOV, A. V., AND IHLEMANN, J. A viscoplasticity model with an enhanced control of the yield surface distortion. *International Journal of Plasticity 39*, 0 (2012), pp. 152–167.

[37] SHUTOV, A. V., AND IHLEMANN, J. On the phenomenological modeling of yield surface distortion.

[38] SLOAN, S. W. Substepping schemes for the numerical integration of elastoplastic stress–strain relations. *International Journal for Numerical Methods in Engineering 24*, 5 (1987), pp. 893–911.

[39] SZCZEPIŃSKI, W. On the effect of plastic deformation on yield criterion. *Archives of Mechanics 15* (1963), pp. 276–296.

[40] VOYIADJIS, G. Z., AND FOROOZESH, M. Anisotropic distortional yield model. *J. App. Mech. 57*, 3 (1990), pp. 537–547.

[41] WILKINS, M. L. Calculation of elastic-plastic flow. *Methods of Computational Physics 3*, 0 (1964), pp. 389–397.

[42] WILLIAMS, J. F., AND SVENSSON, N. L. Effect of tensile prestrain on the yield locus of 1100-f aluminium. *Journal of Strain Analysis 5*, 2 (1970), pp. 128–139.

[43] WU, H. C., AND YEH, W. C. On the experimental determination of yield surfaces and some results on annealed 304 stainless steel. *International Journal of Plasticity 7*, 8 (1991), pp. 803–826.

[44] ZHANG, M., BENÍTEZ, J. M., AND MONTÁNS, F. J. Capturing yield surface evolution with a multilinear anisotropic kinematic hardening model. *International Journal of Solids and Structures 81* (2016), pp. 329–336.

[45] ZIENKIEWICZ, O. C., AND CORMEAU, I. C. Visco-plasticity–plasticity and creep in elastic solids: a unified numerical solution approach. *International Journal for Numerical Methods in Engineering 8*, 4 (1974), pp. 821–845.

Appendix A

Yield function gradients

The following gradients enter the consistency equation:

A.1 Model 1 Yield Function gradients

Due to the abrupt saturation of the distortion feature in Model 1, different gradients have to be calculated for the saturated state.

$$\begin{aligned}
\frac{\partial \mathrm{f}}{\partial \boldsymbol{\alpha}} &= \|\mathbf{b}\| \frac{-c}{\sqrt{\frac{8}{3}\left[1 - c\,\|\mathbf{z}\|\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]}} \left[\mathbf{n}_\mathrm{z}\,(\mathbf{n}_\mathrm{r} : \mathbf{z}) + \|\mathbf{z}\| \left(\frac{\mathbf{n}_\mathrm{r} : \mathbf{z}\mathbf{n}_\mathrm{r} - \mathbf{z}}{\|\mathbf{s} - \boldsymbol{\alpha}\|} + \mathbf{n}_\mathrm{r}\right)\right] + \\
&+ \sqrt{\tfrac{3}{2}\left[1 - c\,\|\mathbf{z}\|\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]} \left[\left(\frac{1}{\sqrt{1 - c\mathbf{z} : \mathbf{z}}} + c_\mathrm{cr}\,\|\mathbf{z}\| - 1\right) \cdot \right. \\
&\cdot \left.\left(\left(\frac{\mathbf{I} - \mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z}}{\|\mathbf{z}\|} : (\mathbf{s} - \boldsymbol{\alpha}) - \mathbf{n}_\mathrm{z}\right) \otimes \mathbf{n}_\mathrm{z} + \frac{\mathbf{I} - \mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z}}{\|\mathbf{z}\|} \mathbf{n}_\mathrm{z} : (\mathbf{s} - \boldsymbol{\alpha})\right) + \right. \\
&+ \left.(c_\mathrm{cr}\,\|\mathbf{z}\| - 1)\,\mathbf{I} - c_\mathrm{cr}\mathbf{n}_\mathrm{z} \otimes (\mathbf{s} - \boldsymbol{\alpha}) + \right. \\
&+ \left.\left(\frac{c\mathbf{z}}{\sqrt{1 - c\mathbf{z} : \mathbf{z}^3}} + c_\mathrm{cr}\mathbf{n}_\mathrm{z}\right) \otimes \mathbf{n}_\mathrm{z}\mathbf{n}_\mathrm{z} : (\mathbf{s} - \boldsymbol{\alpha})\right] : \frac{\mathbf{b}}{\|\mathbf{b}\|}
\end{aligned} \tag{A.1}$$

$$\begin{aligned}
\frac{\partial \mathrm{f}_\mathrm{sat}}{\partial \boldsymbol{\alpha}} &= \|\mathbf{b}\| \frac{c_\mathrm{zz}}{\sqrt{\frac{8}{3}\left[1 - c_\mathrm{zz}\,(\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z})\right]}} \left(\frac{\mathbf{n}_\mathrm{z} - (\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z})\,\mathbf{n}_\mathrm{r}}{\|\mathbf{s} - \boldsymbol{\alpha}\|} + \frac{\mathbf{n}_\mathrm{z}\,(\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z}) - \mathbf{n}_\mathrm{r}}{\|\mathbf{z}\|}\right) + \\
&+ \sqrt{\tfrac{3}{2}\left[1 - c_\mathrm{zz}\,(\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z})\right]} \left[\left(\frac{1}{\sqrt{1 - c_\mathrm{zz}}} + c_\mathrm{crz} - 1\right) \cdot \right. \\
&\cdot \left.\left(\left(\frac{\mathbf{I} - \mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z}}{\|\mathbf{z}\|} : (\mathbf{s} - \boldsymbol{\alpha}) - \mathbf{n}_\mathrm{z}\right) \otimes \mathbf{n}_\mathrm{z} + \frac{\mathbf{I} - \mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z}}{\|\mathbf{z}\|} \mathbf{n}_\mathrm{z} : (\mathbf{s} - \boldsymbol{\alpha})\right) + \right. \\
&+ \left.(c_\mathrm{crz} - 1)\,\mathbf{I}\right] : \frac{\mathbf{b}}{\|\mathbf{b}\|}
\end{aligned} \tag{A.2}$$

$$\begin{aligned}\frac{\partial \mathrm{f}}{\partial \boldsymbol{\beta}} &= \|\mathbf{b}\| \frac{c}{\sqrt{\frac{8}{3}\left[1 - c\,\|\mathbf{z}\|\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]}} \left(\mathbf{n}_\mathrm{z}\,(\mathbf{n}_\mathrm{r} : \mathbf{z}) + \|\mathbf{z}\|\,\mathbf{n}_\mathrm{r}\right) + \\ &+ \sqrt{\tfrac{3}{2}\left[1 - c\,\|\mathbf{z}\|\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]} \left[\left(\frac{1}{\sqrt{1 - c\mathbf{z} : \mathbf{z}}} + c_\mathrm{cr}\,\|\mathbf{z}\| - 1 \right) \cdot \right. \\ &\cdot \left. \left(\left(\frac{\mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z} - \mathbf{I}}{\|\mathbf{z}\|} : (\mathbf{s} - \boldsymbol{\alpha}) \right) \otimes \mathbf{n}_\mathrm{z} + \frac{\mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z} - \mathbf{I}}{\|\mathbf{z}\|} \mathbf{n}_\mathrm{z} : (\mathbf{s} - \boldsymbol{\alpha}) \right) + \right. \\ &+ \left. c_\mathrm{cr} \mathbf{n}_\mathrm{z} \otimes (\mathbf{s} - \boldsymbol{\alpha}) - \left(\frac{c\mathbf{z}}{\sqrt{1 - c\mathbf{z} : \mathbf{z}^3}} + c_\mathrm{cr} \mathbf{n}_\mathrm{z} \right) \otimes \mathbf{n}_\mathrm{z} \mathbf{n}_\mathrm{z} : (\mathbf{s} - \boldsymbol{\alpha}) \right] : \frac{\mathbf{b}}{\|\mathbf{b}\|} \end{aligned} \tag{A.3}$$

$$\begin{aligned}\frac{\partial \mathrm{f}_\mathrm{sat}}{\partial \boldsymbol{\beta}} &= \|\mathbf{b}\| \frac{c_\mathrm{zz}}{\sqrt{\frac{8}{3}\left[1 - c_\mathrm{zz}\,(\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z})\right]}} \left(\frac{\mathbf{n}_\mathrm{r} - \mathbf{n}_\mathrm{z}\,(\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z})}{\|\mathbf{z}\|} \right) + \\ &+ \sqrt{\tfrac{3}{2}\left[1 - c_\mathrm{zz}\,(\mathbf{n}_\mathrm{r} : \mathbf{n}_\mathrm{z})\right]} \left[\left(\frac{1}{\sqrt{1 - c_\mathrm{zz}}} + c_\mathrm{crz} - 1 \right) \cdot \right. \\ &\cdot \left. \left(\left(\frac{\mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z} - \mathbf{I}}{\|\mathbf{z}\|} : (\mathbf{s} - \boldsymbol{\alpha}) \right) \otimes \mathbf{n}_\mathrm{z} + \frac{\mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z} - \mathbf{I}}{\|\mathbf{z}\|} \mathbf{n}_\mathrm{z} : (\mathbf{s} - \boldsymbol{\alpha}) \right) \right] : \frac{\mathbf{b}}{\|\mathbf{b}\|} \end{aligned} \tag{A.4}$$

A.2 Model 2 Yield Function gradients

$$\begin{aligned}\frac{\partial \mathrm{f}}{\partial \boldsymbol{\alpha}} &= \left[(c_\mathrm{cr}\,\|\mathbf{z}\| - 1)\,\mathbf{I} - \left(\frac{1}{\sqrt{1 - \|c\mathbf{z}\|}} + c_\mathrm{cr}\,\|\mathbf{z}\| - 1 \right) \mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z} \right] : \\ &: \frac{\mathbf{b}}{\|\mathbf{b}\|} \sqrt{\tfrac{3}{2}\left[1 - c\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]} + \|\mathbf{b}\| \frac{-c}{\sqrt{\frac{8}{3}\left[1 - c\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]}} \frac{\mathbf{n}_\mathrm{r} : \mathbf{z}\mathbf{n}_\mathrm{r} - \mathbf{z}}{\|\mathbf{s} - \boldsymbol{\alpha}\|} \end{aligned} \tag{A.5}$$

$$\begin{aligned}\frac{\partial \mathrm{f}}{\partial \mathbf{z}} &= \sqrt{\tfrac{3}{2}\left[1 - c\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]} \frac{\mathbf{b}}{\|\mathbf{b}\|} : \left[\mathbf{n}_\mathrm{z} \otimes \left(\frac{c\mathbf{n}_\mathrm{z}}{2\sqrt{1 - \|c\mathbf{z}\|^3}} + c_\mathrm{cr} \mathbf{n}_\mathrm{z} \right) \mathbf{n}_\mathrm{z} : (\mathbf{s} - \boldsymbol{\alpha}) - \right. \\ &- c_\mathrm{cr}\,(\mathbf{s} - \boldsymbol{\alpha}) \otimes \mathbf{n}_\mathrm{z} + \left(\frac{1}{\sqrt{1 - \|c\mathbf{z}\|}} + c_\mathrm{cr}\,\|\mathbf{z}\| - 1 \right) \left(\mathbf{n}_\mathrm{z} \otimes \frac{\mathbf{I} - \mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z}}{\|\mathbf{z}\|} : \right. \\ &: \left. \left. (\mathbf{s} - \boldsymbol{\alpha}) + \frac{\mathbf{I} - \mathbf{n}_\mathrm{z} \otimes \mathbf{n}_\mathrm{z}}{\|\mathbf{z}\|} \mathbf{n}_\mathrm{z} : (\mathbf{s} - \boldsymbol{\alpha}) \right) \right] - \|\mathbf{b}\| \frac{c\mathbf{n}_\mathrm{r}}{\sqrt{\frac{8}{3}\left[1 - c\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]}} \end{aligned} \tag{A.6}$$

$$\frac{\partial \mathrm{f}}{\partial c} = \frac{-\mathbf{n}_\mathrm{r} : \mathbf{z}}{\sqrt{\frac{8}{3}\left[1 - c\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]}} \|\mathbf{b}\| - \sqrt{\tfrac{3}{2}\left[1 - c\,(\mathbf{n}_\mathrm{r} : \mathbf{z})\right]} \frac{\mathbf{b} : \mathbf{n}_\mathrm{z}\,\|\mathbf{z}\|}{2\,\|\mathbf{b}\|\,\sqrt{1 - \|c\mathbf{z}\|^3}} \mathbf{n}_\mathrm{z} : (\mathbf{s} - \boldsymbol{\alpha}) \tag{A.7}$$

Appendix B

ABAQUS UMAT implementation

B.1 Model 1 ABAQUS implementation

```
************************************************************************
*     ABAQUS USER MATERIAL                                             *
*                                                                      *
*     Coupled Directional Distortional Hardening                       *
*     Contracted Alpha model with fixed distortion c                   *
*     Transverse width (cross effect) ccr                              *
*     Axis of distortion goes through Bounding Surface origin at beta. *
************************************************************************
*     State variable array length ... NSTATV = 39    NSTAV = 3+6*NTENS *
*                           NTENS is the num. of values per tensor.    *
************************************************************************
*     STATEV(1) = EPSPc          ... Cummulative plastic strain        *
*     STATEV(2) = k              ... Isotropic part                    *
*     STATEV(3) = Kbs            ... Bounding Surface size             *
*     STATEV(4:3+NTENS) = Alpha            ... Backstress              *
*     STATEV(4+NTENS:3+2*NTENS)   = Alpha1 ... Backstress comp. 1      *
*     STATEV(4+2*NTENS:3+3*NTENS) = Alpha2 ... Backstress comp. 2      *
*     STATEV(4+3*NTENS:3+4*NTENS) = Alpha3 ... Backstress comp. 3      *
*     STATEV(4+4*NTENS:3+5*NTENS) = Alpha4 ... Backstress comp. 4      *
*     STATEV(4+5*NTENS:3+6*NTENS) = Beta   ... Bounding Surface center *
*                                                                      *
*     PROPS(1) ... E                   PROPS(14)... Kbs0 (BS iso)      *
*     PROPS(2) ... nu                  PROPS(15)... K_1  (BS iso)      *
*     PROPS(3) ... Initial yield k_0   PROPS(16)... K_2  (BS iso)      *
*     PROPS(4) ... kappa_1 (iso)       PROPS(17)... b_1  (BS kin)      *
*     PROPS(5) ... kappa_2 (iso)       PROPS(18)... b_2  (BS kin)      *
*     PROPS(6) ... kappa_3 (iso)       PROPS(19)... eps_plateau        *
*     PROPS(7) ... kappa_4 (iso)       PROPS(20)... c0 (d. distortion) *
*     PROPS(8) ... h_1     (kin)       PROPS(21)... ccr0 (cross effect)*
*     PROPS(9) ... h_2     (kin)                                       *
*     PROPS(10)... h_3     (kin)                                       *
*     PROPS(11)... h_4     (kin)                                       *
*     PROPS(12)... h_5     (kin)                                       *
*     PROPS(13)... eps     (kink)                                      *
*                                  version 1   Rene Marek 28.01.2018   *
************************************************************************
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3),
     4 Sig(6),dEps(6)
```

```
C
      COMMON /KUMAT/ PROPSI(21),DEPROP(6)
     3 tol,sqrt23,sqrt32,EPSPc,SAnorm,zNorm,bNorm,Snznr,Sroot,Sg,Sh,
     4 SAlpnz,Const,Sk,f,SKbs,SdfCdf,SKp,
     5 Sig0(6),S(6),Alpha(6),SAlp(6),tnr(6),Beta(6),tz(6),tnz(6),
     6 tb(6),dfdsig(6),dfdAlp(6),dfdBet(6),tn(6),dSig(6),tCedf(6),
     7 Alpha1(6),Alpha2(6),Alpha3(6),Alpha4(6),
     8 TCe(6,6),TIId(6,6),Tnzonz(6,6),TCfoCf(6,6)
      EQUIVALENCE (PROPS(5),Ckap2),(PROPSI(11),h4),(PROPSI(12),h5),
     2 (PROPS(16),CK2),(PROPS(20),c0),(PROPS(21),ccr0),
     3 (DEPROP(1),hAA),(DEPROP(2),hBB),(DEPROP(3),hCC),
     2 (DEPROP(4),zLim),(DEPROP(5),czzlim),(DEPROP(6),hclim)
************************************************************************
C  Relative stress tolerance, default 1e-4
      tol = 1.D-4
C  Nominal division of model integration
      NINTn = 50
************************************************************************
      sqrt32 = 1.224744871391589D0
      sqrt23 = 0.816496580927726D0
C  Lamme 1st constant
      CLAM = PROPS(2)*PROPS(1)/(1.+PROPS(2))/(1.-2.D0*PROPS(2))
C  Lamme 2nd constant
      CG = 0.5D0*PROPS(1)/(1.+PROPS(2))
C  Model parameters
      do i=1,21
        PROPSI(i) = PROPS(i)
      end do
C  Cummulative effective plastic strain recovery
      EPSPc = STATEV(1)
C  Stress tolerance
      tol = tol*PROPS(3)
C  k recovery
      if (STATEV(2).EQ.0.D0) then
        STATEV(2) = PROPS(3)
      end if
      Sk = STATEV(2)
C  Kbs recovery
      if (STATEV(3).EQ.0.D0) then
        STATEV(3) = tol
      end if
      SKbs = STATEV(3)
C  Backstress recovery
      j = 4
      do i=1,NTENS
        Alpha(i) = STATEV(j)
        j = j+1
      end do
C  Backstress component 1 recovery
      do i=1,NTENS
        Alpha1(i) = STATEV(j)
        j = j+1
      end do
C  Backstress component 2 recovery
      do i=1,NTENS
        Alpha1(i) = STATEV(j)
        j = j+1
      end do
C  Backstress component 3 recovery
      do i=1,NTENS
        Alpha1(i) = STATEV(j)
        j = j+1
      end do
C  Backstress component 4 recovery
      do i=1,NTENS
        Alpha1(i) = STATEV(j)
        j = j+1
      end do
C  Bounding Surface position recovery
      do i=1,NTENS
        Beta(i) = STATEV(j)
        j = j+1
```

```
      end do
C  Identity, Elastic stiffness tensors
      do i=1,NTENS
        do j=1,NTENS
          TIId(i,j) = 0.D0
          TCe(i,j) = 0.D0
        end do
      end do
      do i=1,NDI
        TIId(i,i) = 2.D0/3.D0
        TCe(i,i) = CLAM + 2.D0*CG
      end do
      do i=2,NDI
        k = i-1
        do j=1,k
          TIId(i,j) = -1.D0/3.D0
          TIId(j,i) = -1.D0/3.D0
          TCe(i,j) = CLAM
          TCe(j,i) = CLAM
        end do
      end do
      do i=NDI+1,NTENS
        TIId(i,i) = 0.5D0
        TCe(i,i) = 2.D0*CG
      end do
C ------------------------------ Dependent properties ------------------
      if (c0.GT.0) then
        zLim = 0.816496581D0*max(1./Ckap2,1./CK2)/c0
      else
        zLim = 1e9
      end if
      czzlim = 0.99*0.968*(-1.+sqrt(1.+16.*(1.-ccr0)**4))
     /                    /4./(1.-ccr0)**2
      hclim = -sqrt(1.5D0*(1.+czzlim)/(1.-czzlim))
      hck = (czzlim*hclim/(1.-czzlim) -
     -       czzlim/sqrt(2./3.*(1.-czzlim**2)))/zLim

      hBB = (h4*h5*(h4-1.)+1.)/(1.-h5*(1.-h4))
      hAA = h5 * (hBB-h4)**2
      hCC = hAA/(h4-hBB)
C -------------------------------------------------------------------------
      do i=1,NTENS
        Sig(i) = STRESS(i)
      end do
      do i=1,NDI
        dEps(i) = DSTRAN(i)
      end do
C ------------------------------ Trial stress evaluation --------------
      do i=NDI+1,NTENS
        dEps(i) = 0.5D0*DSTRAN(i)
      end do
      call KMAP42(TCe,dEps,dSig,NDI,NTENS)
      do i=1,NTENS
        Sig(i) = Sig(i) + dSig(i)
      end do
      call KPLCON(Sig,NDI,NTENS)
      IF (f.LT.tol) THEN
        do i=1,NTENS
          STRESS(i) = Sig(i)
          do j=1,NTENS
            DDSDDE(i,j) = TCe(i,j)
          end do
        end do
        RETURN
      END IF
C ------------------------------ Searching for plastic part -----------
      Ppart = 0.D0
      call KGRAD(NDI,NTENS)
  100 continue
        call KSCAL2(dfdSig,dSig,Stemp,NDI,NTENS)
        DPpart = f/Stemp
        do i=1,NTENS
```

```
          Sig(i) = Sig(i) - DPpart*dSig(i)
        end do
        Ppart = Ppart + DPpart
        call KPLCON(Sig,NDI,NTENS)
      IF (f.GT.tol) THEN
        GOTO 100
      END IF
      do i=1,NTENS
        Sig0(i) = Sig(i)
      end do
      Ppart = max(0.,1.-Ppart)
C -----------------------------------------------------------------------
      do i=1,NTENS
        dEps0(i) = dEps(i)
        dEps(i) = Ppart*dEps(i)
        dSig(i) = Ppart*dSig(i)
        Eps(i) = Eps(i) + dEps(i)
      end do
      Ppart = 1.
      call KSC2SQ(dSig,dSig,dSigNorm,NDI,NTENS)
C ---------------------------- Model evolution --------------------------
      call KFRULE(Sig,NDI,NTENS)
C ---------------------------- Radial r. mapping ------------------------
  200 call KPLCON(Sig,NDI,NTENS)
        IF (abs(f).LT.tol) goto 300
        call KGRAD(NDI,NTENS)
        call KSC2SQ(dfdSig,dfdSig,Stemp,NDI,NTENS)
        do i=1,NTENS
          Sig(i) = Sig(i) - dfdSig(i)*f/Stemp**2
        end do
      goto 200
C ------------------------- Continuous tangent matrix DDSDDE ------------
  300 call KSCAL2(dfdSig,tCedf,dfCdf,NDI,NTENS)
      call KOtims(tCedf,tCedf,TCfoCf,NTENS)
      SdfCdf = SdfCdf + SKp
C
      do i=1,NTENS
        STRESS(i) = Sig(i)
        do j=1,NTENS
          DDSDDE(i,j) = TCe(i,j) - TCfoCf(i,j)/SdfCdf
        end do
      end do
C ------------------- Store updated internal variables ------------------
C  Cummulative plastic strain update
      STATEV(1) = EPSPc
C  k update
      STATEV(2) = Sk
C  Kbs update
      STATEV(3) = SKbs
C  BACKSTRESS UPDATE
      j=4
      do i=1,NTENS
        STATEV(j) = Alpha(i)
        j = j+1
      end do
      do i=1,NTENS
        STATEV(j) = Alpha1(i)
        j = j+1
      end do
      do i=1,NTENS
        STATEV(j) = Alpha2(i)
        j = j+1
      end do
      do i=1,NTENS
        STATEV(j) = Alpha3(i)
        j = j+1
      end do
      do i=1,NTENS
        STATEV(j) = Alpha4(i)
        j = j+1
      end do
      do i=1,NTENS
```

```
        STATEV(j) = Beta(i)
        j = j+1
      end do
      RETURN
      END
C
*********************************************************************
*     Plastic Condition - Yield fun. evaluation                     *
*********************************************************************
      SUBROUTINE KPLCON(Sig,NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION Sig(6)
      COMMON /KUMAT/ E,Cnu,Ck0,Ckap1,Ckap2,Ckap3,Ckap4,h0,h1,h2,h3,h4,
     1 epsknk,CKbs0,CK1,CK2,b1,b2,eplt,c0,ccr0,
     2 hAA,hBB,hCC,zLim,czzlim,hclim,hck,c,ccr,
     3 tol,sqrt23,sqrt32,EPSPc,SAnorm,zNorm,bNorm,Snznr,Sroot,Sg,Sh,
     4 SAlpnz,Const,Sk,f,SKbs,SdfCdf,SKp,ckey,
     5 Sig0(6),S(6),Alpha(6),SAlp(6),tnr(6),Beta(6),tz(6),tnz(6),
     6 tb(6),dfdsig(6),dfdAlp(6),dfdBet(6),tn(6),Alphai(6),AlphaB(6),
     7 AlpBin(6),tzin(6),dSig(6),tCedf(6),
     8 TCe(6,6),TIId(6,6),Tnzonz(6,6),TCfCf(6,6)
C
      call KDEV(Sig,S,NDI,NTENS)
      do i=1,NTENS
         SAlp(i) = S(i) - Alpha(i)
      end do
      call KSC2SQ(SAlp,SAlp,SAnorm,NDI,NTENS)
      tz = Alpha-Beta
      call KSC2SQ(tz,tz,zNorm,NDI,NTENS)
      if (zNorm.GT.tol) then
        do i=1,NTENS
          tnz(i) = tz(i)/zNorm
        end do
      else
        do i=1,NTENS
          tnz(i) = 0.D0
        end do
      end if
C
      if ((c*zNorm**2).LT.czzlim) then
        czz = c*zNorm**2
        ccrz = ccr*zNorm
        ckey = 0.
      else
        czz = czzlim
        ccrz = ccr0
        ckey = 1.
      end if
C
      if (SAnorm.GT.tol) then
        do i=1,NTENS
          tnr(i) = SAlp(i)/SAnorm
        end do
      else
        do i=1,NTENS
          tnr(i) = 0.D0
        end do
        f = -Sk
        Sroot = 0.D0
        Snznr = 0.D0
        RETURN
      end if
      call KSCAL2(tnz,tnr,Snznr,NDI,NTENS)
      Sroot = sqrt(1.5D0*(1.-czz*Snznr))
      call KSCAL2(SAlp,tnz,SAlpnz,NDI,NTENS)
      Sg = 1. - ccrz
      Sh = 1./sqrt(1.-czz) + ccrz - 1.
      ShS = Sh*SAlpnz
      do i=1,NTENS
         tb(i) = Sg*SAlp(i) + ShS*tnz(i)
```

```
      end do
      call KSC2SQ(tb,tb,bNorm,NDI,NTENS)
C     Yield Func.
      f = Sroot*bNorm - Sk
      RETURN
      END
C
*********************************************************************
*     Yield Func. gradient                                          *
*********************************************************************
      SUBROUTINE KGRAD(NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      COMMON /KUMAT/ E,Cnu,Ck0,Ckap1,Ckap2,Ckap3,Ckap4,h0,h1,h2,h3,h4,
     1 epsknk,CKbs0,CK1,CK2,b1,b2,eplt,c0,ccr0,
     2 hAA,hBB,hCC,zLim,czzlim,hclim,hck,c,ccr,
     3 tol,sqrt23,sqrt32,EPSPc,SAnorm,zNorm,bNorm,Snznr,Sroot,Sg,Sh,
     4 SAlpnz,Const,Sk,f,SKbs,SdfCdf,SKp,ckey,
     5 Sig0(6),S(6),Alpha(6),SAlp(6),tnr(6),Beta(6),tz(6),tnz(6),
     6 tb(6),dfdsig(6),dfdAlp(6),dfdBet(6),tn(6),Alphai(6),AlphaB(6),
     7 AlpBin(6),tzin(6),dSig(6),tCedf(6),
     8 TCe(6,6),TIId(6,6),Tnzonz(6,6),TCfCf(6,6)
      DIMENSION TTemp(6,6),Tdnzdz(6,6)
C
      Const = 4.D0/3.D0*Sroot
      if (bNorm.GT.tol) then
         Stemp = Sroot/bNorm
         call KSCAL2(tb,tnz,bnz,NDI,NTENS)
      else
         Stemp = 0.D0
         bnz = 0.
      end if
      do i=1,NTENS
        dfdSig(i) = Stemp*(tb(i)*Sg + Sh*bnz*tnz(i))
      end do
      if (SAnorm.GT.tol) then
        Stemp = czz*bNorm/Const/SAnorm
      else
        Stemp = 0.D0
      end if
      do i=1,NTENS
        dfdSig(i) = dfdSig(i) + Stemp*(Snznr*tnr(i)-tnz(i))
      end do
      str = 0.D0
      do i=1,NDI
         str = str + dfdSig(i)
      end do
      str = str/3.D0
      do i=1,NDI
        dfdSig(i) = dfdSig(i) - str
      end do
      RETURN
      END
C
*********************************************************************
*     Other Yield Fun. gradients                                    *
*     tn,df/dAlpha, df/dBeta, AlphaB                                *
*********************************************************************
      SUBROUTINE KFRULE(Sig,NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      COMMON /KUMAT/ E,Cnu,Ck0,Ckap1,Ckap2,Ckap3,Ckap4,h0,h1,h2,h3,h4,
     1 epsknk,CKbs0,CK1,CK2,b1,b2,eplt,c0,ccr0,
     2 hAA,hBB,hCC,zLim,czzlim,hclim,hck,c,ccr,
     3 tol,sqrt23,sqrt32,EPSPc,SAnorm,zNorm,bNorm,Snznr,Sroot,Sg,Sh,
     4 SAlpnz,Const,Sk,f,SKbs,SdfCdf,SKp,ckey,
     5 Sig0(6),S(6),Alpha(6),SAlp(6),tnr(6),Beta(6),tz(6),tnz(6),
     6 tb(6),dfdsig(6),dfdAlp(6),dfdBet(6),tn(6),Alphai(6),AlphaB(6),
     7 AlpBin(6),tzin(6),dSig(6),tCedf(6),
     8 TCe(6,6),TIId(6,6),Tnzonz(6,6),TCfCf(6,6)
```

```
      DIMENSION Sig(6),tempt(6),Alpdot(6),Betdot(6),tzzin(6),tABAin(6),
     ,tABA(6),Tdnzdz(6,6),Tdnzsa(6,6),TTemp(6,6),TdbdX(6,6),Tnzosa(6,6),
     ,TII(6,6),RHOE(6)
      do i=1,NTENS
        do j=1,NTENS
          TII(i,j) = 0.
        end do
      end do
      do i=1,NDI
        TII(i,i) = 1.
      end do
      do i=1+NDI,NTENS
        TII(i,i) = 0.5
      end do
C
      call KGRAD(NDI,NTENS)
C ------------------------------------ YF outer normal ---------------
      call KSC2SQ(dfdSig,dfdSig,GNorm,NDI,NTENS)
      do i=1,NTENS
        tn(i) = dfdSig(i)/GNorm
      end do
C ---------------------------------------- df/dBeta ------------------
      IF (zNorm.LT.tol) THEN
        do i=1,NTENS
          do j=1,NTENS
            TdbdX(i,j) = 0
          end do
        end do
      ELSE
        IF (keyc.EQ.0) THEN
          call KOtims(tnz,tnz,Tnzonz,NTENS)
          do i=1,NTENS
            do j=1,NTENS
              Tdnzdz(i,j) = (Tnzonz(i,j)-TII(i,j))/zNorm
            end do
          end do
          call KMAP42(Tdnzdz,SAlp,tdnzSA,NDI,NTENS)
          call KOtims(tdnzSA,tnz,TTemp,NTENS)
          do i=1,NTENS
            do j=1,NTENS
              Tdnzsa(i,j) = TTemp(i,j)+Tdnzdz(i,j)*SAlpnz
            end do
          end do
          Stemp = czz/(1.-czz)**1.5
          do i=1,NTENS
            tempt(i) = Stemp*tz(i)+ccrz*tnz(i)
          end do
          call KOtims(tempt,tnz,TTemp,NTENS)
          call KSCAL2(SAlp,tnz,sanz,NDI,NTENS)
          call KOtims(tnz,SAlp,Tnzosa,NTENS)
          do i=1,NTENS
            do j=1,NTENS
              TdbdX(i,j) = ccrz*Tnzosa(i,j)-
     -                     TTemp(i,j)*sanz+Sh*Tdnzsa(i,j)
            end do
          end do
        ELSE
         call KOtims(tnz,tnz,Tnzonz,NTENS)
          do i=1,NTENS
            do j=1,NTENS
              Tdnzdz(i,j) = (Tnzonz(i,j)-TII(i,j))/zNorm
            end do
          end do
          call KMAP42(Tdnzdz,SAlp,tdnzSA,NDI,NTENS)
          call KOtims(tdnzSA,tnz,TTemp,NTENS)
          do i=1,NTENS
            do j=1,NTENS
              Tdnzsa(i,j) = TTemp(i,j)+Tdnzdz(i,j)*SAlpnz
            end do
          end do
          do i=1,NTENS
            do j=1,NTENS
```

```
              TdbdX(i,j) = TTemp(i,j)*sanz+Sh*Tdnzsa(i,j)
            end do
          end do
        END IF
      END IF
      call KMAP42(TdbdX,tb,dfdBet,NDI,NTENS)
      do i=1,NTENS
        dfdBet(i) = bNorm*czz/Const*(tnz(i)*Snznr+tnr(i))/zNorm+
     +              Sroot/bnorm*dfdBet(i)
      end do
C ---------------------------------------- df/dAlpha ------------------
      IF (keyc.EQ.0) THEN
        do i=1,NTENS
          do j=1,NTENS
          TdbdX(i,j)=-TdbdX(i,j)-Sh*Tnzonz(i,j)+(ccr*zNorm-1.)*TII(i,j)
          end do
        end do
        call KMAP42(TdbdX,tb,dfdAlp,NDI,NTENS)
        do i=1,NTENS
          tempt(i)  = (Snznr*tnr(i)-tnz(i))/SANorm+tnr(i)
          dfdAlp(i) = -bNorm*czz/zNorm/Const*(tnz(i)*Snznr+tempt(i))+
     +                Sroot/bnorm*dfdAlp(i)
        end do
      ELSE
        do i=1,NTENS
          do j=1,NTENS
          TdbdX(i,j)=-TdbdX(i,j)-Sh*Tnzonz(i,j)+(ccr*zNorm-1.)*TII(i,j)
          end do
        end do
        call KMAP42(TdbdX,tb,dfdAlp,NDI,NTENS)
        do i=1,NTENS
          tempt(i)  = (Snznr*tnr(i)-tnz(i))/SANorm+tnr(i)
          dfdAlp(i) = -bNorm*czz/zNorm/Const*(tnz(i)*Snznr+tempt(i))+
     +                Sroot/bnorm*dfdAlp(i)
        end do
      END IF
C
**********************************************************************
      do i=1,NTENS
        AlphaB(i) = Beta(i) + 0.816496580928*nn(i)*(Kbs+tol)
        tnalpB(i) = AlphaB(i)-Alpha(i)
      end do
      call KSC2SQ(tnalpB,tnalpB,delta,NDI,NTENS)
      if (delta.GT.tol) then
        do i=1,NTENS
          tnalpB(i) = nalpB(i)/delta
        end do
        delta = delta-tol
      else
        do i=1,NTENS
          tnalpB(i) = nn(i)
        end do
        delta = 0.
      end if
C ----------------- Collapse empty component -----------------------------
        call KSC2SQ(Alpha2,Alpha2,Alpha2Norm,NDI,NTENS)
        if (Alpha2Norm.LT.tol) then
          do i=1,NTENS
            Alpha2(i) = Alpha2(i)+Alpha3(i)
            Alpha3(i) = 0.
          end do
          call KSC2SQ(Alpha2,Alpha2,Alpha2Norm,NDI,NTENS)
        end if
C ----------------- Fuse similarly oriented components 1 and 2 ----------
        call KSC2SQ(Alpha1,Alpha1,Alpha1Norm,NDI,NTENS)
        if (Alpha1Norm.GT.tol) then
          do i=1,NTENS
            tnalp1(i) = Alpha1(i)/Alpha1Norm
          end do
        else
          do i=1,NTENS
            tnalp1(i) = tnalpB(i)
```

```
          end do
        end if
        if (Alpha2Norm.GT.tol) then
          do i=1,NTENS
            tnalp2(i) = Alpha2(i)/Alpha2Norm
          end do
        else
          do i=1,NTENS
            tnalp2(i) = 0
          end do
        end if
        call KSCAL2(tnalp1,tnalp2,tempA,NDI,NTENS)
        IF (tempA.GT.0.95) THEN ! Fuse threshold
          do i=1,NTENS
            Alpha1(i) = Alpha1(i) + Alpha2(i)
            Alpha2(i) = Alpha3(i)
            Alpha3(i) = 0
          end do
          call KSC2SQ(Alpha1,Alpha1,Alpha1Norm,NDI,NTENS)
          if (Alpha1Norm.GT.tol) then
           do i=1,NTENS
             tnalp1(i) = Alpha1(i)/Alpha1Norm
           end do
          else
            do i=1,NTENS
              tnalp1(i) = tnalpB(i)
            end do
          end if
          call KSC2SQ(Alpha2,Alpha2,Alpha2Norm,NDI,NTENS)
          if (Alpha2Norm.GT.tol) then
            do i=1,NTENS
              tnalp2(i) = Alpha2(i)/Alpha2Norm
            end do
          else
            do i=1,NTENS
              tnalp2(i) = 0
            end do
          end if
        END IF
C ----------------- Fuse similarly oriented components 2 and 3 ----------
        call SC2SQ(Alpha3,Alpha3,Alpha3Norm,NDI,NTENS)
        if (Alpha3Norm.GT.tol) then
          do i=1,NTENS
            tnalp3(i) = Alpha3(i)/Alpha3Norm
          end do
        else
          do i=1,NTENS
            tnalp3(i) = 0
          end do
        end if
        call KSCAL2(tnalp2,tnalp3,tempA,NDI,NTENS)
        IF (tempA.GT.0.95) THEN  ! Fuse threshold
          do i=1,NTENS
            Alpha2(i) = Alpha2(i) + Alpha3(i)
          end do
          call SC2SQ(Alpha2,Alpha2,Alpha2Norm,NDI,NTENS)
          if (Alpha2Norm.GT.tol) then
            do i=1,NTENS
              tnalp2(i) = Alpha2(i)/Alpha2Norm
            end do
          else
            do i=1,NTENS
              tnalp2(i) = 0
            end do
          end if
          do i=1,NTENS
            Alpha3(i) = 0
            tnalp3(i) = 0
          end do
        END IF
C ------------- Create new component for new loading direction ----------
        call KSCAL2(tnalp1,tnalpB,tempA,NDI,NTENS)
```

```
        ps = 0.
        IF (tempA.LT.0.5) THEN  ! Initiation threshold
          do i=1,NTENS
            Alpha3(i) = Alpha3(i) + Alpha2(i)
            Alpha2(i) = Alpha1(i)
            Alpha1(i) = 0
          end do
        ELSEIF (tempA.LT.0.8) THEN  ! Smoothing threshold
          ps = max(0.D0,exp(-10.*(tempA-0.8))-1.)
        END IF
C -----------------------------------------------------------------------
        call KSC2SQ(Alpha4,Alpha4,Alpha4Norm,NDI,NTENS)
        if (Alpha4Norm.GT.tol) then
            do i=1,NTENS
              tnalp4(i) = Alpha4(i)/Alpha4Norm
            end do
          else
            do i=1,NTENS
              tnalp4(i) = 0
            end do
        end if
C -----------------------------------------------------------------------
        call KSCAL2(AlphaB-Alpha,tn,Stemp,NDI,NTENS)
        Stemp = max(0.D0,Stemp-delta)
        call SCAL2(tnalpB,tnz,Stemp)
        if ((Stemp.GT.0).AND.(delta.LT.tol/10.)) then
          delta = 0.
        end if
        if (zNorm.GT.zLim) then
          Stemp = max(0.D0,-Stemp)
          hdist = 1. + Stemp*((SAlpNorm*hck+hclim)/hclim-1.)
        else
          hdist = 1.
        end if
C ------------------------------ kbar, Kbsbar + Betabar -------------------------
      IF (EPSPc.LT.eplt) THEN
        Skbar = Ckap1*(1.D0-Ckap2*Sk)
        do i=1,NTENS
          Betbar(i) = 0.
        end do
        SKbsB = -Skbar
      ELSE
        Skbar = Ckap3*(1.D0-Ckap4*Sk)
        do i=1,NTENS
          Betbar(i) = b1*(tn(i)-b2*Beta(i))
        end do
        SKbsB = -Skbar + CK1*(1.D0-CK2*(SKbs-Sk))
      END IF
C --------------------------------- Alphabar ----------------------------
      call KSCAL2(tnalp2,tnalpB,tempA)
        tempB = A1Norm
        V2 = h2*hdist/((tempB+delta)**2+epsknk)
        V2 = V2*delta/(tempB+epskink)
        F2 = V2*(1.-exp(-h3*Al2Norm))*
     *       max(0.,hAA/(hBB+tempA)+hCC)
C -------------------
        call KSCAL2(nalp3,nalpB,tempA)
        tempB = A1Norm+A2Norm
        V3 = h2*hdist/((tempB+delta)**2+epsknk)
        V3 = V3*delta/(tempB+epskink)
        F3 = V3*(1.-exp(-h3*A3Norm))*
     *       max(0.,hAA/(hBB+tempA)+hCC)
C --------------------
        call SCAL2(nalp4,nalpB,tempA)
        Grow = 0.816496581*Kbsbar + h1*delta
        tempB = A1Norm+A2Norm+A3Norm
        V4 = h2*hdist/((tempB+delta)**2+epsknk)
        V4 = V4*delta/(tempB+epskink)
        F4 = V4*(1.-exp(-h3*A4Norm))*
     *       max(0.,hAA/(hBB+tempA)+hCC)
C ---------------------
        do i=1,NTENS
```

```
          A1bar(i) = (F2+F3+F4)*tnalpB(i) - ps*Alpha1(i)
          A2bar(i) =-F2*tnalp2(i) + ps*Alpha1(i) - ps*Alpha2(i)
          A3bar(i) =-F3*tnalp3(i) + ps*Alpha2(i)
          A4bar(i) = Grow*tnalpB(i)-F4*tnalp4(i)
          Alpbar(i) = A1bar(i)+A2bar(i)+A3bar(i)+A4bar(i) + Betabar(i)
        end do
C ---------------------------- Plastic modulus Kp -----------------------
      call KSCAL2(dfdAlp,Alpdot,Stmp,NDI,NTENS)
      call KSCAL2(dfdBet,Betdot,Stemp,NDI,NTENS)
      SKp = -Stmp - Stemp + Skdot
C ------------------------ Plastic multiplier lambda --------------------
      call KSCAL2(dfdSig,dSig,FDE,NDI,NTENS)
      call KMAP42(TCe,tn,Cdfdsig,NDI,NTENS)
      call KSCAL2(dfdSig,Cdfdsig,FDR,NDI,NTENS)
      Slamb = max(0.,FDE/(SKp+FDR))
      if (Slamb.EQ.0) goto 100
C --------------------------- lambda limits -----------------------------
      lambdak = 1e6
      IF (skbar.NE.0.) THEN
        if (cepsp.LT.cepsplat) then
          lambdak = min((1./kappa2-kk)/3./kbar, k0/NINT/abs(kbar))
        else
          lambdak = min((1./kappa4-kk)/3./kbar, k0/NINT/abs(kbar))
        end if
      END IF
*  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
      lambdakbs = 1e6
      if (SKbsB.NE.0) lambdakbs = min(k0/NINT/abs(SKbsB),
     ,            abs((1./CK2-SKbs+Sk)/3./(SKbsB-Skbar)))
*  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
      lamc = 1e6
      lambet = 1e6
      call SCAL2SQ(Alpbar,Alpbar,tempC)
      call SCAL2SQ(Alpha,Alpha,tempA)
      if (tempA.GT.tol) then
        call SCAL2(Alpha/tempA,tnalpB,tempA)
        tempA = 1.+5.*(1.-tempA**2)
      else
        tempA = 1.
      end if
      lamalp = min(1e6,0.8164965809*k0/tempC/NINT/tempA)
      call KSC2SQ(Alpha-Beta,Alpha-Beta,zNorm)
      lama1 = 1e6
      lama2 = 1e6
      lama3 = 1e6
      lama4 = 1e6
C     critical lambda for disappearing segments
      call SCAL2(tnalp1,Alpha1bar,tempA)
      if (tempA*lambda.LT.0.) lamalp1 = -A1Norm/tempA/3.
      call SCAL2(tnalp2,Alpha2bar,tempA)
      if (tempA*lambda.LT.0.) lamalp2 = -A2Norm/tempA/3.
      call SCAL2(tnalp3,Alpha3bar,tempA)
      if (tempA*lambda.LT.0.) lamalp3 = -A3Norm/tempA/3.
      call SCAL2(tnalp4,Alpha4bar,tempA)
      if (tempA*lambda.LT.0.) lamalp4 = -A4Norm/tempA/3.
      lambda = min(lambda,lambdak,lambdakbs,lambdac,
     ,             lama,lama1,lama2,lama3,lama4,lambet)
C ------------------------------- Evolution ----------------------------
      EPSPc = EPSPc + DEPSP
      Sk = Sk + Slamb*SkDot
      SKbs = SKbs + Slamb*SKbsD
C
      do i=1,NTENS
        Alpha(i) = Alpha(i) + Slamb*Alpdot(i)
        Beta(i) = Beta(i) + Slamb*Betdot(i)
        tz(i) = Alpha(i) - Beta(i)
      end do
      call KSC2SQ(tz,tz,zNorm,NDI,NTENS)
      if (zNorm.GT.sqrt23*SKbs) then
        do i=1,NTENS
          Alpha(i) = Beta(i) + tz(i)/zNorm*sqrt23*SKbs
        end do
```

```
      end if
      call KMAP42(TCe,dfdSig,tCedf,NDI,NTENS)
      do i=1,NTENS
         Sig(i) = Sig0(i) + dSig(i) - Slamb*tCedf(i)
      end do
C
      END
************************************************************************
*     ABAQUS SDVINI SUBROUTINE                                         *
*                                                                      *
*     INITIAL VALUES OF STATEV FIELD                                   *
************************************************************************
      SUBROUTINE SDVINI(STATEV,COORDS,NSTATV,NCRDS,NOEL,NPT,
     1 LAYER,KSPT)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION STATEV(NSTATV),COORDS(NCRDS)
C
      do i=1,NSTATV
         STATEV(i) = 0.D0
      end do
C
      RETURN
      END
C
************************************************************************
*     Deviatoric part of tensor                                        *
************************************************************************
      SUBROUTINE KDEV(tA,tD,NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION tA(6),tD(6)
C
      str = 0.D0
      do i=1,NDI
         str = str + tA(i)
      end do
      str = str/3.D0
      do i=1,NDI
         tD(i) = tA(i) - str
      end do
      do i=NDI+1,NTENS
         tD(i) = tA(i)
      end do
C
      RETURN
      END
************************************************************************
*     KMAP24                                                           *
************************************************************************
      SUBROUTINE KMAP24(tA,TB,tC,NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION tA(6),TB(6,6),tC(6)
C
      DO j=1,NTENS
        x = 0.D0
        do i=NDI+1,NTENS
          x = x + tA(i)*TB(i,j)
        end do
        x = 2.*x
        do i=1,NDI
          x = x + tA(i)*TB(i,j)
        end do
        tC(j) = x
      END DO
      RETURN
      END
************************************************************************
```

```
*     KMAP42                                                           *
**********************************************************************
      SUBROUTINE KMAP42(TA,tB,tC,NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION TA(6,6),tB(6),tC(6)
C
      DO i=1,NTENS
        x = 0.D0
        do j=NDI+1,NTENS
          x = x + TA(i,j)*tB(j)
        end do
        x = 2.*x
        do j=1,NDI
          x = x + TA(i,j)*tB(j)
        end do
        tC(i) = x
      END DO
      RETURN
      END
**********************************************************************
*     Scalar prod. c = A_ij B_ij                                       *
**********************************************************************
      SUBROUTINE KSCAL2(tA,tB,SC,NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION tA(6),tB(6)
C
      SC = 0.D0
      do i=NDI+1,NTENS
         SC = SC + tA(i)*tB(i)
      end do
      SC = 2.D0*SC
      do i=1,NDI
         SC = SC + tA(i)*tB(i)
      end do
C
      RETURN
      END
**********************************************************************
*     Square root of scalar prod. c = A_ij B_ij                        *
**********************************************************************
      SUBROUTINE KSC2SQ(tA,tB,SQC,NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION tA(6),tB(6)
C
      SQC = 0.D0
      do i=NDI+1,NTENS
         SQC = SQC + tA(i)*tB(i)
      end do
      SQC = 2.D0*SQC
      do i=1,NDI
         SQC = SQC + tA(i)*tB(i)
      end do
      SQC = sqrt(max(0.D0,SQC))
C
      RETURN
      END
**********************************************************************
*     Tensor multiplication T_ijkl = a_ij b_kl                         *
**********************************************************************
      SUBROUTINE KOtims(tA,tB,TAB,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION tA(6),tB(6),TAB(6,6)
C
      do i=1,NTENS
```

```
         do j=1,NTENS
            TAB(i,j) = tA(i)*tB(j)
         end do
      end do
C
      RETURN
      END
************************************************************************
*     END OF SUBROUTINES                                               *
************************************************************************
```

B.2 Model 2 ABAQUS implementation - original subroutines

```
************************************************************************
*     ABAQUS USER MATERIAL                                             *
*                                                                      *
*     Uncoupled Directional Distortional Hardening                     *
*     Contracted Alpha model with uncoupled evolving distortion c      *
*     transverse width (cross effect) ccr                              *
*     Axis of distortion has independent evolution rules (r tensor)    *
************************************************************************
*     State variable array length ... NSTATV = 47  NSTAV = 5+7*NTENS,  *
*                           NTENS is the num. of values per tensor.    *
************************************************************************
*     STATEV(1) = EPSPc         ... Cummulative plastic strain         *
*     STATEV(2) = k             ... Isotropic part                     *
*     STATEV(3) = Kbs           ... Bounding Surface size              *
*     STATEV(4:3+NTENS) = Alpha            ... Backstress              *
*     STATEV(4+NTENS:3+2*NTENS)   = Alpha1 ... Backstress comp. 1      *
*     STATEV(4+2*NTENS:3+3*NTENS) = Alpha2 ... Backstress comp. 2      *
*     STATEV(4+3*NTENS:3+4*NTENS) = Alpha3 ... Backstress comp. 3      *
*     STATEV(4+4*NTENS:3+5*NTENS) = Alpha4 ... Backstress comp. 4      *
*     STATEV(4+5*NTENS:3+6*NTENS) = Beta   ... Bounding Surface center *
*     STATEV(4+6*NTENS:3+7*NTENS) = z      ... Yield surface axis      *
*     STATEV(5+7*NTENS) = c ... Yield surface distortion               *
*                                                                      *
*     PROPS(1) ... E                   PROPS(14)... Kbs0 (BS iso)      *
*     PROPS(2) ... nu                  PROPS(15)... K_1  (BS iso)      *
*     PROPS(3) ... Initial yield k_0   PROPS(16)... K_2  (BS iso)      *
*     PROPS(4) ... kappa_1 (iso)       PROPS(17)... b_1  (BS kin)      *
*     PROPS(5) ... kappa_2 (iso)       PROPS(18)... b_2  (BS kin)      *
*     PROPS(6) ... kappa_3 (iso)       PROPS(19)... eps_plateau        *
*     PROPS(7) ... kappa_4 (iso)       PROPS(20)... c0 (d. distortion) *
*     PROPS(8) ... h_1     (kin)       PROPS(21)... c1                 *
*     PROPS(9) ... h_2     (kin)       PROPS(22)... c2                 *
*     PROPS(10)... h_3     (kin)       PROPS(23)... ccr0 (cross effect)*
*     PROPS(11)... h_4     (kin)       PROPS(24)... rho1 (YS axis sat.)*
*     PROPS(12)... h_5     (kin)       PROPS(25)... rho2 (YS axis rot.)*
*     PROPS(13)... eps     (kink)      PROPS(26)... rho3 (YS axis ini.)*
*                                   version 1   Rene Marek 28.01.2018  *
************************************************************************
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,
     2 STRAN,DSTRAN,TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,CMNAME,
     3 NDI,NSHR,NTENS,NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,
     4 CELENT,DFGRD0,DFGRD1,NOEL,NPT,LAYER,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
C
      CHARACTER*80 CMNAME
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),DFGRD0(3,3),DFGRD1(3,3),
     4 Sig(6),dEps(6)
C
      COMMON /KUMAT/ PROPSI(14),
     2 tol,sqrt23,sqrt32,EPSPc,SAnorm,zNorm,bNorm,znr,Sroot,Sg,Sh,
     3 SAlpnz,Const,Sk,f,SKbs,SdfCdf,SKp,
     4 Sig0(6),S(6),Alpha(6),SAlp(6),tnr(6),Beta(6),tz(6),tnz(6),
     5 tb(6),dfdsig(6),dfdAlp(6),dfdBet(6),tn(6),Alphai(6),AlphaB(6),
     6 AlpBin(6),tzin(6),dSig(6),tCedf(6),
     7 TCe(6,6),TIId(6,6),Tnzonz(6,6),TCfoCf(6,6)
************************************************************************
C  Relative stress tolerance 2 initial yield limit
      tol = 1.D-4
C  Nominal division of model integration relative 2 initial yield limit
      NINTn = 20
************************************************************************
      sqrt32 = 1.224744871391589D0
```

```
      sqrt23 = 0.816496580927726D0
C  1st LAMME
      CLAM = PROPS(2)*PROPS(1)/(1.+PROPS(2))/(1.-2.D0*PROPS(2))
C  2nd LAMME
      CG = 0.5D0*PROPS(1)/(1.+PROPS(2))
C  Model parameters
      do i=1,14
        PROPSI(i) = PROPS(i+2)
      end do
C  Cummulative effective plastic strain recovery
      EPSPc = STATEV(1)
C  k recovery
      if (STATEV(2).EQ.0.D0) then
        STATEV(2) = PROPS(3)
        if (PROPS(16).LE.0.) then
          STATEV(3) = 1./PROPS(5)
        end if
      end if
      Sk = STATEV(2)
C  Kbs recovery
      SKbs = STATEV(3)
C  Backstress recovery
      j = 4
      do i=1,NTENS
        Alpha(i) = STATEV(j)
        j = j+1
      end do
C  Initial backstress recovery
      do i=1,NTENS
        Alphai(i) = STATEV(j)
        j = j+1
      end do
C  Initial projected backstress recovery
      do i=1,NTENS
        AlpBin(i) = STATEV(j)
        j = j+1
      end do
C  Bounding Surface recovery
      do i=1,NTENS
        Beta(i) = STATEV(j)
        j = j+1
      end do
C  Initial z recovery
      do i=1,NTENS
        tzin(i) = STATEV(j)
        j = j+1
      end do
C  Stress tolerance
      tol = tol*PROPS(3)
C  Identity, Elastic stiffness tensors
      do i=1,NTENS
        do j=1,NTENS
          TIId(i,j) = 0.D0
          TCe(i,j) = 0.D0
        end do
      end do
      do i=1,NDI
        TIId(i,i) = 2.D0/3.D0
        TCe(i,i) = CLAM + 2.D0*CG
      end do
      do i=2,NDI
        k = i-1
        do j=1,k
          TIId(i,j) = -1.D0/3.D0
          TIId(j,i) = -1.D0/3.D0
          TCe(i,j) = CLAM
          TCe(j,i) = CLAM
        end do
      end do
      do i=NDI+1,NTENS
        TIId(i,i) = 0.5D0
        TCe(i,i) = 2.D0*CG
```

```
      end do
C ------------------------------ Dependent properties ------------------
      hBB = (h4*h5*(h4-1.)+1.)/(1.-h5*(1.-h4))
      hAA = h5 * (hBB-h4)**2
      hCC = hAA/(h4-hBB)
C --------------------------------------------------------------------
C --------------------------------------------------------------------
      do i=1,NTENS
        Sig(i) = STRESS(i)
      end do
      do i=1,NDI
        dEps(i) = DSTRAN(i)
      end do
C ------------------------------ Trial stress evaluation --------------
      do i=NDI+1,NTENS
        dEps(i) = 0.5D0*DSTRAN(i)
      end do
      call KMAP42(TCe,dEps,dSig,NDI,NTENS)
      do i=1,NTENS
        Sig(i) = Sig(i) + dSig(i)
      end do
      call KPLCON(Sig,NDI,NTENS)
      IF (f.LT.tol) THEN
        do i=1,NTENS
          STRESS(i) = Sig(i)
          do j=1,NTENS
            DDSDDE(i,j) = TCe(i,j)
          end do
        end do
        RETURN
      END IF
C ------------------------------ Searching for plastic part -----------
      Ppart = 0.D0
      call KGRAD(NDI,NTENS)
  100 continue
        call KSCAL2(dfdSig,dSig,Stemp,NDI,NTENS)
        DPpart = f/Stemp
        do i=1,NTENS
          Sig(i) = Sig(i) - DPpart*dSig(i)
        end do
        Ppart = Ppart + DPpart
        call KPLCON(Sig,NDI,NTENS)
      IF (f.GT.tol) THEN
        GOTO 100
      END IF
      do i=1,NTENS
        Sig0(i) = Sig(i)
      end do
      Ppart = max(0.,1.-Ppart)
C --------------------------------------------------------------------
      do i=1,NTENS
        dEps0(i) = dEps(i)
        dEps(i) = Ppart*dEps(i)
        dSig(i) = Ppart*dSig(i)
        Eps(i) = Eps(i) + dEps(i)
      end do
      Ppart = 1.
      call KSC2SQ(dSig,dSig,dSigNorm,NDI,NTENS)
C ---------------------------- Model evolution --------------------------
      call KFRULE(Sig,NDI,NTENS)
C ---------------------------- Radial r. mapping ------------------------
  200 call KPLCON(Sig,NDI,NTENS)
        IF (abs(f).LT.tol) goto 300
        call KGRAD(NDI,NTENS)
        call KSC2SQ(dfdSig,dfdSig,Stemp,NDI,NTENS)
        do i=1,NTENS
          Sig(i) = Sig(i) - dfdSig(i)*f/Stemp**2
        end do
      goto 200
C ------------------------- Continuous tangent matrix DDSDDE ------------
  300 call KSCAL2(dfdSig,tCedf,dfCdf,NDI,NTENS)
      call KOtims(tCedf,tCedf,TCfoCf,NTENS)
```

```
      SdfCdf = SdfCdf + SKp
C
      do i=1,NTENS
        STRESS(i) = Sig(i)
        do j=1,NTENS
          DDSDDE(i,j) = TCe(i,j) - TCfoCf(i,j)/SdfCdf
        end do
      end do
C ------------------- Store updated internal variables ------------------
C  Cummulative plastic strain update
      STATEV(1) = EPSPc
C  k update
      STATEV(2) = Sk
C  Kbs update
      STATEV(3) = SKbs
C  Backstress update
      j=4
      do i=1,NTENS
        STATEV(j) = Alpha(i)
        j = j+1
      end do
      do i=1,NTENS
        STATEV(j) = Alpha1(i)
        j = j+1
      end do
      do i=1,NTENS
        STATEV(j) = Alpha2(i)
        j = j+1
      end do
      do i=1,NTENS
        STATEV(j) = Alpha3(i)
        j = j+1
      end do
      do i=1,NTENS
        STATEV(j) = Alpha4(i)
        j = j+1
      end do
      do i=1,NTENS
        STATEV(j) = Beta(i)
        j = j+1
      end do
C  Yield surface axis
      do i=1,NTENS
        STATEV(j) = tz(i)
        j = j+1
      end do
      RETURN
      END
*********************************************************************
*     Plastic Condition - Yield func. evaluation                    *
*********************************************************************
      SUBROUTINE KPLCON(Sig,NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      DIMENSION Sig(6)
      COMMON /KUMAT/
     1 Ck0,Ckap1,Ckap2,Ckap3,Ckap4,h0,h1,h2,epskink,b1,b2,c,ccr,eplt,
     2 tol,sqrt23,sqrt32,EPSPc,SAnorm,zNorm,bNorm,znr,Sroot,Sg,Sh,
     3 SAlpnz,Const,Sk,f,SKbs,SdfCdf,SKp,
     4 Sig0(6),S(6),Alpha(6),SAlp(6),tnr(6),Beta(6),tz(6),tnz(6),
     5 tb(6),dfdsig(6),dfdAlp(6),dfdBet(6),tn(6),Alphai(6),AlphaB(6),
     6 AlpBin(6),tzin(6),dSig(6),tCedf(6),
     7 TCe(6,6),TIId(6,6),Tnzonz(6,6),TCfCf(6,6)
C
      call KDEV(Sig,S,NDI,NTENS)
      do i=1,NTENS
         SAlp(i) = S(i) - Alpha(i)
      end do
      call KSC2SQ(SAlp,SAlp,SAnorm,NDI,NTENS)
      call KSC2SQ(tz,tz,zNorm,NDI,NTENS)
      if (zNorm.GT.tol) then
```

```
        do i=1,NTENS
          tnz(i) = tz(i)/zNorm
        end do
      else
        do i=1,NTENS
          tnz(i) = 0.D0
        end do
      end if
      if (SAnorm.GT.tol) then
        do i=1,NTENS
          tnr(i) = SAlp(i)/SAnorm
        end do
      else
        do i=1,NTENS
          tnr(i) = 0.D0
        end do
        f = -Sk
        Sroot = 0.D0
        znr = 0.D0
        RETURN
      end if
      call KSCAL2(tz,tnr,znr,NDI,NTENS)
      Sroot = sqrt(1.5D0*(1.-c*znr))
      call KSCAL2(SAlp,tnz,SAlpnz,NDI,NTENS)
      Sg = 1. - ccr*zNorm
      Sh = 1./sqrt(1.-c*zNorm) + ccr*zNorm - 1.
      ShS = Sh*SAlpnz
      do i=1,NTENS
         tb(i) = Sg*SAlp(i) + ShS*tnz(i)
      end do
      call KSC2SQ(tb,tb,bNorm,NDI,NTENS)
C     Yield Func.
      f = Sroot*bNorm - Sk
C
      RETURN
      END
*************************************************************************
*     Yield Func. gradient                                              *
*************************************************************************
      SUBROUTINE KGRAD(NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      COMMON /KUMAT/
     1 Ck0,Ckap1,Ckap2,Ckap3,Ckap4,h0,h1,h2,epskink,b1,b2,c,ccr,eplt,
     2 tol,sqrt23,sqrt32,EPSPc,SAnorm,zNorm,bNorm,znr,Sroot,Sg,Sh,
     3 SAlpnz,Const,Sk,f,SKbs,SdfCdf,SKp,
     4 Sig0(6),S(6),Alpha(6),SAlp(6),tnr(6),Beta(6),tz(6),tnz(6),
     5 tb(6),dfdsig(6),dfdAlp(6),dfdBet(6),tn(6),Alphai(6),AlphaB(6),
     6 AlpBin(6),tzin(6),dSig(6),tCedf(6),
     7 TCe(6,6),TIId(6,6),Tnzonz(6,6),TCfCf(6,6)
      DIMENSION TTemp(6,6),Tdnzdz(6,6)
C
      Const = 4.D0/3.D0*Sroot
      if (bNorm.GT.tol) then
         Stemp = Sroot/bNorm
         call KSCAL2(tb,tnz,bnz,NDI,NTENS)
      else
         Stemp = 0.D0
         bnz = 0.
      end if
      do i=1,NTENS
        dfdSig(i) = Stemp*(tb(i)*Sg + Sh*bnz*tnz(i))
      end do
      if (SAnorm.GT.tol) then
        Stemp = c*bNorm*zNorm/Const/SAnorm
      else
        Stemp = 0.D0
      end if
      do i=1,NTENS
        dfdSig(i) = dfdSig(i) + Stemp*(znr*tnr(i)-tz(i))
      end do
```

```
      str = 0.D0
      do i=1,NDI
         str = str + dfdSig(i)
      end do
      str = str/3.D0
      do i=1,NDI
        dfdSig(i) = dfdSig(i) - str
      end do
      RETURN
      END
*************************************************************************
*     Other Yield Fun. gradients                                        *
*     tn,df/dAlpha, df/dBeta, AlphaB                                    *
*************************************************************************
      SUBROUTINE KFRULE(Sig,NDI,NTENS)
C
      INCLUDE 'ABA_PARAM.INC'
C
      COMMON /KUMAT/ E,Cnu,Ck0,Ckap1,Ckap2,Ckap3,Ckap4,h0,h1,h2,h3,h4,
     1 epsknk,CKbs0,CK1,CK2,b1,b2,eplt,c0,ccr0,
     2 hAA,hBB,hCC,zLim,czzlim,hclim,hck,c,ccr,
     3 tol,sqrt23,sqrt32,EPSPc,SAnorm,zNorm,bNorm,Snznr,Sroot,Sg,Sh,
     4 SAlpnz,Const,Sk,f,SKbs,SdfCdf,SKp,ckey,
     5 Sig0(6),S(6),Alpha(6),SAlp(6),tnr(6),Beta(6),tz(6),tnz(6),
     6 tb(6),dfdsig(6),dfdAlp(6),dfdBet(6),tn(6),Alphai(6),AlphaB(6),
     7 AlpBin(6),tzin(6),dSig(6),tCedf(6),
     8 TCe(6,6),TIId(6,6),Tnzonz(6,6),TCfCf(6,6)
      DIMENSION Sig(6),tempt(6),Alpdot(6),Betdot(6),tzzin(6),tABAin(6),
     ,tABA(6),Tdnzdz(6,6),Tdnzsa(6,6),TTemp(6,6),TdbdX(6,6),Tnzosa(6,6),
     ,TII(6,6),RHOE(6)
      do i=1,NTENS
        do j=1,NTENS
          TII(i,j) = 0.
        end do
      end do
      do i=1,NDI
        TII(i,i) = 1.
      end do
      do i=1+NDI,NTENS
        TII(i,i) = 0.5
      end do
C
      call KGRAD(NDI,NTENS)
C ------------------------------------- YF outer normal ---------------
      call KSC2SQ(dfdSig,dfdSig,GNorm,NDI,NTENS)
      do i=1,NTENS
        tn(i) = dfdSig(i)/GNorm
      end do
C ---------------------------------------- df/dBeta -------------------
      IF (zNorm.LT.tol) THEN
        do i=1,NTENS
          do j=1,NTENS
            TdbdX(i,j) = 0
          end do
        end do
      ELSE
          call KOtims(tnz,tnz,Tnzonz,NTENS)
          do i=1,NTENS
            do j=1,NTENS
              Tdnzdz(i,j) = (Tnzonz(i,j)-TII(i,j))/zNorm
            end do
          end do
          call KMAP42(Tdnzdz,SAlp,tdnzSA,NDI,NTENS)
          call KOtims(tdnzSA,tnz,TTemp,NTENS)
          do i=1,NTENS
            do j=1,NTENS
              Tdnzsa(i,j) = TTemp(i,j)+Tdnzdz(i,j)*SAlpnz
            end do
          end do
          Stemp = czz/(1.-czz)**1.5
          do i=1,NTENS
            tempt(i) = Stemp*tz(i)+ccrz*tnz(i)
```

```
          end do
          call KOtims(tempt,tnz,TTemp,NTENS)
          call KSCAL2(SAlp,tnz,sanz,NDI,NTENS)
          call KOtims(tnz,SAlp,Tnzosa,NTENS)
          do i=1,NTENS
            do j=1,NTENS
              TdbdX(i,j) = ccrz*Tnzosa(i,j)-
     -                     TTemp(i,j)*sanz+Sh*Tdnzsa(i,j)
            end do
          end do
       END IF
       call KMAP42(TdbdX,tb,dfdBet,NDI,NTENS)
       do i=1,NTENS
         dfdBet(i) = bNorm*czz/Const*(tnz(i)*Snznr+tnr(i))/zNorm+
     +             Sroot/bnorm*dfdBet(i)
       end do
C ---------------------------------------- df/dAlpha ------------------
         do i=1,NTENS
           do j=1,NTENS
           TdbdX(i,j)=-TdbdX(i,j)-Sh*Tnzonz(i,j)+(ccr*zNorm-1.)*TII(i,j)
           end do
         end do
         call KMAP42(TdbdX,tb,dfdAlp,NDI,NTENS)
         do i=1,NTENS
           tempt(i)  = (Snznr*tnr(i)-tnz(i))/SANorm+tnr(i)
           dfdAlp(i) = -bNorm*czz/zNorm/Const*(tnz(i)*Snznr+tempt(i))+
     +               Sroot/bnorm*dfdAlp(i)
         end do
C
*********************************************************************
       do i=1,NTENS
         AlphaB(i) = Beta(i) + 0.816496580928*nn(i)*(Kbs+tol)
         tnalpB(i) = AlphaB(i)-Alpha(i)
       end do
       call KSC2SQ(tnalpB,tnalpB,delta,NDI,NTENS)
       if (delta.GT.tol) then
         do i=1,NTENS
           tnalpB(i) = nalpB(i)/delta
         end do
         delta = delta-tol
       else
         do i=1,NTENS
           tnalpB(i) = nn(i)
         end do
         delta = 0.
       end if
C ----------------- Collapse empty component -----------------------------
         call KSC2SQ(Alpha2,Alpha2,Alpha2Norm,NDI,NTENS)
         if (Alpha2Norm.LT.tol) then
           do i=1,NTENS
             Alpha2(i) = Alpha2(i)+Alpha3(i)
             Alpha3(i) = 0.
           end do
           call KSC2SQ(Alpha2,Alpha2,Alpha2Norm,NDI,NTENS)
         end if
C ----------------- Fuse similarly oriented components 1 and 2 ----------
         call KSC2SQ(Alpha1,Alpha1,Alpha1Norm,NDI,NTENS)
         if (Alpha1Norm.GT.tol) then
           do i=1,NTENS
             tnalp1(i) = Alpha1(i)/Alpha1Norm
           end do
         else
           do i=1,NTENS
             tnalp1(i) = tnalpB(i)
           end do
         end if
         if (Alpha2Norm.GT.tol) then
           do i=1,NTENS
             tnalp2(i) = Alpha2(i)/Alpha2Norm
           end do
         else
           do i=1,NTENS
```

```
            tnalp2(i) = 0
          end do
        end if
        call KSCAL2(tnalp1,tnalp2,tempA,NDI,NTENS)
        IF (tempA.GT.0.95) THEN ! Fuse threshold
          do i=1,NTENS
            Alpha1(i) = Alpha1(i) + Alpha2(i)
            Alpha2(i) = Alpha3(i)
            Alpha3(i) = 0
          end do
          call KSC2SQ(Alpha1,Alpha1,Alpha1Norm,NDI,NTENS)
          if (Alpha1Norm.GT.tol) then
           do i=1,NTENS
             tnalp1(i) = Alpha1(i)/Alpha1Norm
           end do
          else
            do i=1,NTENS
              tnalp1(i) = tnalpB(i)
            end do
          end if
          call KSC2SQ(Alpha2,Alpha2,Alpha2Norm,NDI,NTENS)
          if (Alpha2Norm.GT.tol) then
            do i=1,NTENS
              tnalp2(i) = Alpha2(i)/Alpha2Norm
            end do
          else
            do i=1,NTENS
              tnalp2(i) = 0
            end do
          end if
        END IF
C ----------------- Fuse similarly oriented components 2 and 3 ----------
        call SC2SQ(Alpha3,Alpha3,Alpha3Norm,NDI,NTENS)
        if (Alpha3Norm.GT.tol) then
          do i=1,NTENS
            tnalp3(i) = Alpha3(i)/Alpha3Norm
          end do
        else
          do i=1,NTENS
            tnalp3(i) = 0
          end do
        end if
        call KSCAL2(tnalp2,tnalp3,tempA,NDI,NTENS)
        IF (tempA.GT.0.95) THEN  ! Fuse threshold
          do i=1,NTENS
            Alpha2(i) = Alpha2(i) + Alpha3(i)
          end do
          call SC2SQ(Alpha2,Alpha2,Alpha2Norm,NDI,NTENS)
          if (Alpha2Norm.GT.tol) then
            do i=1,NTENS
              tnalp2(i) = Alpha2(i)/Alpha2Norm
            end do
          else
            do i=1,NTENS
              tnalp2(i) = 0
            end do
          end if
          do i=1,NTENS
            Alpha3(i) = 0
            tnalp3(i) = 0
          end do
        END IF
C ------------- Create new component for new loading direction ----------
        call KSCAL2(tnalp1,tnalpB,tempA,NDI,NTENS)
        ps = 0.
        IF (tempA.LT.0.5) THEN  ! Initiation threshold
          do i=1,NTENS
            Alpha3(i) = Alpha3(i) + Alpha2(i)
            Alpha2(i) = Alpha1(i)
            Alpha1(i) = 0
          end do
        ELSEIF (tempA.LT.0.8) THEN  ! Smoothing threshold
```

```
          ps = max(0.D0,exp(-10.*(tempA-0.8))-1.)
        END IF
C -----------------------------------------------------------------------
        call KSC2SQ(Alpha4,Alpha4,Alpha4Norm,NDI,NTENS)
        if (Alpha4Norm.GT.tol) then
            do i=1,NTENS
              tnalp4(i) = Alpha4(i)/Alpha4Norm
            end do
          else
            do i=1,NTENS
              tnalp4(i) = 0
            end do
        end if
C -----------------------------------------------------------------------
        call KSCAL2(AlphaB-Alpha,tn,Stemp,NDI,NTENS)
        Stemp = max(0.D0,Stemp-delta)
        call SCAL2(tnalpB,tnz,Stemp)
        if ((Stemp.GT.0).AND.(delta.LT.tol/10.)) then
          delta = 0.
        end if
        if (zNorm.GT.zLim) then
          Stemp = max(0.D0,-Stemp)
          hdist = 1. + Stemp*((SAlpNorm*hck+hclim)/hclim-1.)
        else
          hdist = 1.
        end if
C ------------------------------ kbar, Kbsbar + Betabar -------------------------
      IF (EPSPc.LT.eplt) THEN
        Skbar = Ckap1*(1.D0-Ckap2*Sk)
        do i=1,NTENS
          Betbar(i) = 0.
        end do
        SKbsB = -Skbar
      ELSE
        Skbar = Ckap3*(1.D0-Ckap4*Sk)
        do i=1,NTENS
          Betbar(i) = b1*(tn(i)-b2*Beta(i))
        end do
        SKbsB = -Skbar + CK1*(1.D0-CK2*(SKbs-Sk))
      END IF
C --------------------------------- Alphabar ----------------------------
      call KSCAL2(tnalp2,tnalpB,tempA)
        tempB = A1Norm
        V2 = h2*hdist/((tempB+delta)**2+epsknk)
        V2 = V2*delta/(tempB+epskink)
        F2 = V2*(1.-exp(-h3*A12Norm))*
     *       max(0.,hAA/(hBB+tempA)+hCC)
C -------------------
        call KSCAL2(nalp3,nalpB,tempA)
        tempB = A1Norm+A2Norm
        V3 = h2*hdist/((tempB+delta)**2+epsknk)
        V3 = V3*delta/(tempB+epskink)
        F3 = V3*(1.-exp(-h3*A3Norm))*
     *       max(0.,hAA/(hBB+tempA)+hCC)
C --------------------
        call SCAL2(nalp4,nalpB,tempA)
        Grow = 0.816496581*Kbsbar + h1*delta
        tempB = A1Norm+A2Norm+A3Norm
        V4 = h2*hdist/((tempB+delta)**2+epsknk)
        V4 = V4*delta/(tempB+epskink)
        F4 = V4*(1.-exp(-h3*A4Norm))*
     *       max(0.,hAA/(hBB+tempA)+hCC)
C --------------------
        do i=1,NTENS
          A1bar(i) = (F2+F3+F4)*tnalpB(i) - ps*Alpha1(i)
          A2bar(i) =-F2*tnalp2(i) + ps*Alpha1(i) - ps*Alpha2(i)
          A3bar(i) =-F3*tnalp3(i) + ps*Alpha2(i)
          A4bar(i) = Grow*tnalpB(i)-F4*tnalp4(i)
          Alpbar(i) = A1bar(i)+A2bar(i)+A3bar(i)+A4bar(i) + Betabar(i)
        end do
C ---------------------------- Plastic modulus Kp ----------------------
      call KSCAL2(dfdAlp,Alpdot,Stmp,NDI,NTENS)
```

```
      call KSCAL2(dfdBet,Betdot,Stemp,NDI,NTENS)
      SKp = -Stmp - Stemp + Skdot
C ------------------------ Plastic multiplier lambda --------------------
      call KSCAL2(dfdSig,dSig,FDE,NDI,NTENS)
      call KMAP42(TCe,tn,Cdfdsig,NDI,NTENS)
      call KSCAL2(dfdSig,Cdfdsig,FDR,NDI,NTENS)
      Slamb = max(0.,FDE/(SKp+FDR))
      if (Slamb.EQ.0) goto 100
C ---------------------------- lambda limits ----------------------------
      lambdak = 1e6
      IF (skbar.NE.0.) THEN
        if (cepsp.LT.cepsplat) then
          lambdak = min((1./kappa2-kk)/3./kbar, k0/NINT/abs(kbar))
        else
          lambdak = min((1./kappa4-kk)/3./kbar, k0/NINT/abs(kbar))
        end if
      END IF
*  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
      lambdakbs = 1e6
      if (SKbsB.NE.0) lambdakbs = min(k0/NINT/abs(SKbsB),
     ,            abs((1./CK2-SKbs+Sk)/3./(SKbsB-Skbar)))
*  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
      lamc = 1e6
      lambet = 1e6
      call SCAL2SQ(Alpbar,Alpbar,tempC)
      call SCAL2SQ(Alpha,Alpha,tempA)
      if (tempA.GT.tol) then
        call SCAL2(Alpha/tempA,tnalpB,tempA)
        tempA = 1.+5.*(1.-tempA**2)
      else
        tempA = 1.
      end if
      lamalp = min(1e6,0.8164965809*k0/tempC/NINT/tempA)
      call KSC2SQ(Alpha-Beta,Alpha-Beta,zNorm)
      lama1 = 1e6
      lama2 = 1e6
      lama3 = 1e6
      lama4 = 1e6
C     critical lambda for disappearing segments
      call SCAL2(tnalp1,Alpha1bar,tempA)
      if (tempA*lambda.LT.0.) lamalp1 = -A1Norm/tempA/3.
      call SCAL2(tnalp2,Alpha2bar,tempA)
      if (tempA*lambda.LT.0.) lamalp2 = -A2Norm/tempA/3.
      call SCAL2(tnalp3,Alpha3bar,tempA)
      if (tempA*lambda.LT.0.) lamalp3 = -A3Norm/tempA/3.
      call SCAL2(tnalp4,Alpha4bar,tempA)
      if (tempA*lambda.LT.0.) lamalp4 = -A4Norm/tempA/3.
      lambda = min(lambda,lambdak,lambdakbs,lambdac,
     ,             lama,lama1,lama2,lama3,lama4,lambet)
C ------------------------------ Evolution ----------------------------
      EPSPc = EPSPc + DEPSP
      Sk = Sk + Slamb*SkDot
      SKbs = SKbs + Slamb*SKbsD
C
      do i=1,NTENS
        Alpha(i) = Alpha(i) + Slamb*Alpdot(i)
        Beta(i) = Beta(i) + Slamb*Betdot(i)
        tz(i) = Alpha(i) - Beta(i)
      end do
      call KSC2SQ(tz,tz,zNorm,NDI,NTENS)
      if (zNorm.GT.sqrt23*SKbs) then
        do i=1,NTENS
          Alpha(i) = Beta(i) + tz(i)/zNorm*sqrt23*SKbs
        end do
      end if
      call KMAP42(TCe,dfdSig,tCedf,NDI,NTENS)
      do i=1,NTENS
         Sig(i) = Sig0(i) + dSig(i) - Slamb*tCedf(i)
      end do
C
      END
```