Characterizing sheet metals' constitutive behavior using a crystal plasticity based virtual laboratory

Soheil Solhjoo*[1], Jan Post†[2,3], and Antonis I. Vakis‡[1]

[1] *Computational Mechanical and Materials Engineering, Engineering and Technology Institute Groningen, Faculty of Science and Engineering, University of Groningen, Nijenborgh 4, 9747 AG Groningen, the Netherlands*
[2] *Advanced Production Engineering, Engineering and Technology Institute Groningen, Faculty of Science and Engineering, University of Groningen, Nijenborgh 4, 9747 AG Groningen, the Netherlands*
[3] *Philips, the Netherlands*

Contents

*s.solhjoo@rug.nl, soheilsolhjoo@gmail.com
†jan.post@rug.nl, j.post@philips.com
‡a.vakis@rug.nl

1 Introduction

Finite element method (FEM) is an essential tool in metal forming industry, both in tooling design and process simulation. Besides of the large body of literature on the FEM for solving different states of a desired system (process and/or tooling), such as elasticity, plasticity, heat transfer, contact etc., one thing has remained to be a hot topic in the field, namely the description of materials' constitutive behavior.

It is critical to note that the reliability of the results of any types of modelling, both numerical (such as FEM) and analytical (such as models describing stress-strain curves or yield functions), is limited to and dependent on the reliability of the provided materials data. Conventionally, the required material data is determined by performing mechanical tests such as tensile, compression, indentation and shear tests, resulting in some material's characteristics such as elastic modulus, flow stress, recrystallization temperature and hardness. Determining all of the required material's characteristics is expensive (due to the materials and the highly specialized measuring and sampling equipments) and time consuming (for ordering, preparing and testing); however, direct measurements are not always a way to go: for example, performing a shearing test on a sheet metal with a thickness of $\sim 100\mu$m, where the applied forces and the exact deformations have to be measure with a high accuracy, is impossible, due to the small scale [1].

As a solution, some numerical methods such as FEM can be used for approximating those behaviors. To do so, some other material should be defined for the FEM solver. For example, in order to investigate the sample's behavior in a deep drawing test, the process can be simulated only if the material is defined for its constitutive behavior. One common assumption in FEM is that the material behaves as a non-microstructured material. In other words, the sample is assumed to be constructed of no grains, which is valid only for non-crystalline materials. In the case of crystalline matter, however, the elastic-plastic deformation depend on the direction of loading, because the crystals are mechanically anisotropic [2]; therefore, considering the microstructural effects can increase the accuracy of the material's definition and, consequently, the reliability of the simulations results.

One successful method for describing anisotopic inelasticity of polycrystals is Crystal Plasticity (CP), which is based on the behavior of a single crystal [3]. The constitutive modelling behind the CP method describes the strain hardening assigned to individual deformation systems and grain softening phenomena. The models can be in the form of phenomenological of physics-based laws, and their parameters can be identified for the relevant physical quantities either from experiments [4] or from simulations at smaller scales [5, 6].

Once the constitutive model of a single crystal is established, the co-deformation of a polycrystalline aggregate can be approximated with different methods such as full-field simulations, where each grain is spatially resolved by a number of material points, providing spatial distribution of field values [3]. If the simulated sample is large enough to derive quantities comparable with experimental data, it can be considered as a Representative Volume Element (RVE).

The overall mechanical response of the discretized geometry has to be calculated for the mechanical equilibrium of the governing boundary conditions. To do so, there have been only two methods employed in the field of CP, namely FEM (CPFEM) and spectral methods (CPSM) based on fast Fourier transforms. The FEM approaches can be implemented as user-defined subroutines into commercially available FEM solvers, such as MSC-Marc and Abaqus. Moreover, CPFEM offers the flexibility of applying complex boundary conditions on arbitrarily shaped geometries. The CPSM approaches, however, are more efficient, yet limited to periodic boundary conditions (i.e. no implementation in FEM solvers).

This may rise the question: "why do we need the CPSM approach in the first place, if the CPFEM allows a lot of flexibility of the problem?" The answer lies in possibility of modelling the RVE of interest: for large-scale forming problems, i.e. large RVEs, determining the local material responses using full-field simulations is limited by computational resources. One way to overcome this limitation is to perform some micro-scale CPSM simulation to obtain enough relevant data for defining the corresponding material in a conventional FEM model. In other words, the CPSM can serve as a virtual laboratory for mechanical testing on the RVE of interest, and use that data for describing its constitutive behavior at a larger-scale suitable for the FEM. This report describes a framework to obtain all required material data from a CP method and to use it in a FEM simulation.

2 Sheet metals anisotropy in FEM

The achievement of continuum plasticity is rooted in its ability to reduce complex 3D inelastic phenomena to simpler phenomenological laws defined by: (a) a yield criterion, (b) a plastic flow rule as the normal to a plastic potential, which is often assumed to be the yield criterion in case of metals, and (c) an effective stress-strain law [7]. For example, by assuming the validity of the Tresca yield criterion and isotropic hardening, the metal plasticity can be fully characterized by performing only one experiment of uniaxial tensile test. In order to improve the simulations, however, more complicated models are required to be developed and implemented in the FEM solver to accurately model other phenomena, such as anisotropy [8–10] that is the main topic to be addressed in this report.

2.1 Anisotropy of sheet metals

Due to the anisotropic mechanical responses of crystals, the polycrystalline materials also show anisotropic features, although to some different levels; for example, a polycrystalline material with an annealed microstructure may show limited anisotropic behavior compared to a single crystal. For sheet metals, however, the rolling process induces a significant anisotropy in the mechanical properties, which is in favor of industrial applications. Figure 1 depicts a schematic of a rolled sheet metal with the corresponding orientations.

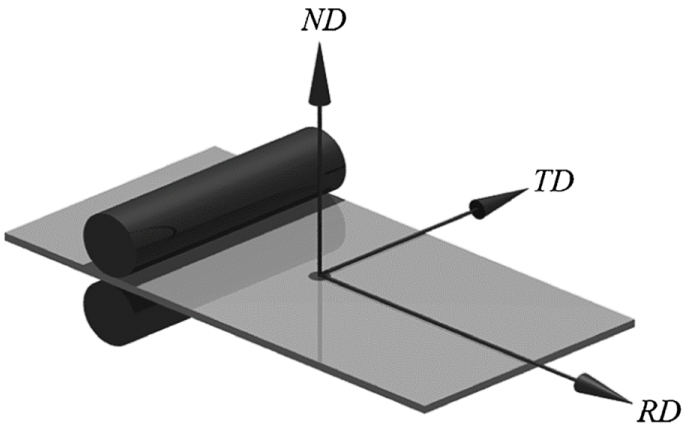


Figure 1: Orthotropy axes of the rolled sheet metals, with RD (rolling direction), TD (transverse direction) and ND (normal direction) [11].

2.1.1 Uniaxial anisotropy

The plastic behavior of the uniaxial samples of a sheet metal depends on the in-plane direction (θ) along which the tensile specimen is cut; see figure 2.

One major method of assessing the uniaxial anisotropic behavior is by a scalar quantity called r-value[1] [12], defined as:

$$r = \epsilon_w/\epsilon_t, \tag{1}$$

where ϵ_w and ϵ_t are the strain in the width (w) and thickness (t), respectively, i.e. $\epsilon_w = \ln(w/w_0)$ and $\epsilon_t = \ln(t/t_0)$, with index 0 indicating the value of the undeformed specimen. Due to the small ratio of thickness to width of tensile specimens, the measurements of these two strain results in different relative errors; therefore, replacing the ϵ_t with $\epsilon_l = \ln(l/l_0)$ (strain in the length direction) would remove this problem. Using the volume constancy condition, i.e. $\epsilon_l + \epsilon_w + \epsilon_t = 0$, the r-value can be rewritten as:

$$r = -\epsilon_w/(\epsilon_l + \epsilon_w). \tag{2}$$

Because the r-value of a specimen cut from a rolled sheet depends on θ, it is noted as r_θ. Figure 3 shows an example of r_θ.

Moreover, the r-value varies with the straining degree; although it depends on the application, it is a common practice to compare the r-values at $\epsilon_l = 0.2$ [11].

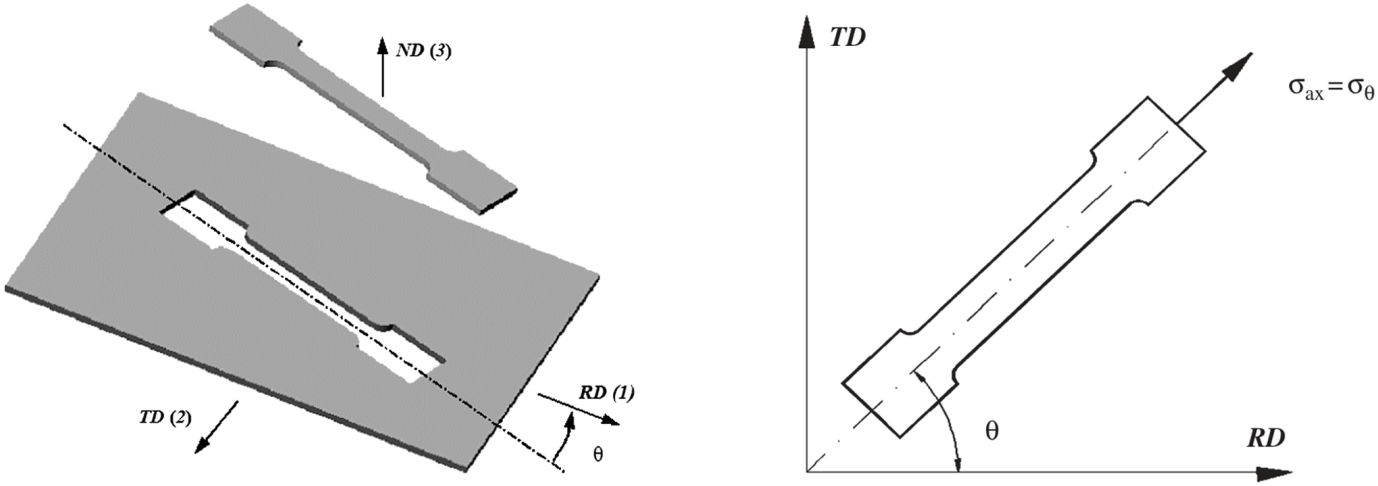
One important role of r-value is to verify the performance of a yield criterion. Denote the directional uniaxial yield stress by σ_θ, the stress tensor components can be written as:

$$\sigma_{11} = \sigma_\theta \cos^2\theta, \tag{3a}$$
$$\sigma_{22} = \sigma_\theta \sin^2\theta, \tag{3b}$$
$$\sigma_{12} = \sigma_\theta \cos\theta \sin\theta; \tag{3c}$$

note that $\sigma_{12} = \sigma_{21}$. By applying a yield function on the stress tenor components, the equivalent stress would be obtained as:

$$Y = \sigma_\theta \cdot F_\theta, \tag{4}$$

[1] This quantity has been called by many different names in the literature: Lankford coefficient, Lankford value, plastic strain ratio, or anisotropy coefficient; however, this quantity is mostly referred to as *r-value*.



(a) (b)

Figure 2: The schematic of uniaxial tensile specimen from a rolled sheet metal [11]: (a) the specimen can be cut at different in-plane angles with respect to the rolling direction (θ), (b) the mechanical response of the tensile specimen varies with θ.

where F_θ is defined from the specific formulation of the worked yield function. Considering the associated flow rule, i.e. defining the plastic flow rule as the normal to a worked yield function, the directional r-value can be written as [11]:

$$r_\theta = \frac{\sigma_{11}\dfrac{\partial\sigma}{\partial\sigma_{11}} + \sigma_{22}\dfrac{\partial\sigma}{\partial\sigma_{22}} + \sigma_{12}\dfrac{\partial\sigma}{\partial\sigma_{12}}}{\sigma_{11}\dfrac{\partial\sigma}{\partial\sigma_{11}} + \sigma_{22}\dfrac{\partial\sigma}{\partial\sigma_{22}}} - 1 = F_\theta\left(\frac{\partial\sigma}{\partial\sigma_{11}} + \frac{\partial\sigma}{\partial\sigma_{22}}\right)^{-1} - 1. \tag{5}$$

Equation 5 defines the r-value as a function of yield function, which can be compared with the experimental values for studying the validity of the yield function. It should be noted that the right side of equation 5 (with F_θ in it) is mostly used for yield functions with rather a simple formulation, while the left one can be implemented for numerical approximations as a solution for yield functions with a complex formulation.

2.1.2 Biaxial anisotropy

In order to identify the parameters of a yield function, a number of experimental tests have to be performed; see figure 4. It should be noted that the straining degree of biaxial tests are rather small (less than 5%).

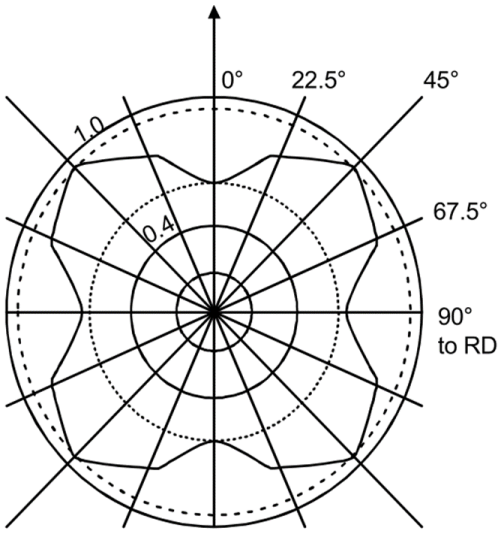


Figure 3: Variation of r-value as function of in-plane direction of an aluminium alloy (AlMg5Mn) [13]. The polar diagram offer direct information on the tendency of the sheet meal to earing.

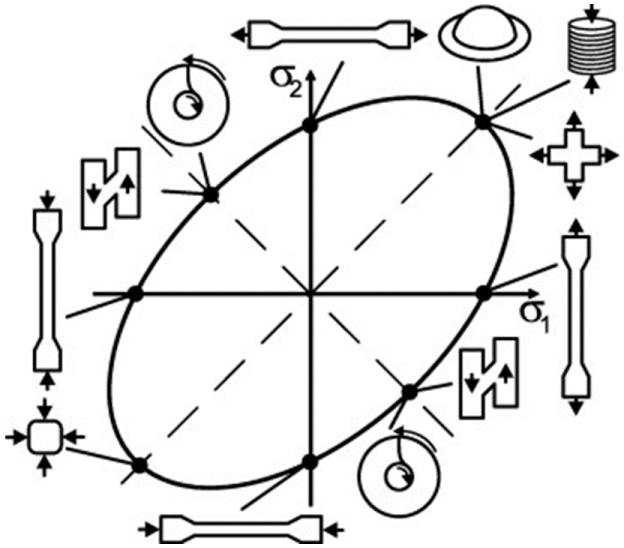


Figure 4: Common testing methods for sheet metal characterization [14].

As was mentioned in section 1, performing experiments are highly demanding; therefore, it is necessary to minimize the number of required experimental tests as low as possible, and derive the most data out of them. One parameter to be used in calibrating some of the yield criteria is biaxial anisotropy coefficient, defined as:

$$r_b = \epsilon_w / \epsilon_l, \tag{6}$$

for the equi-biaxial tensile stress state. If the material is isotropic, $r_b = 1$. This coefficient is linked with the worked yield function via:

$$r_b = F_b \left(\frac{\partial \sigma}{\partial \sigma_{11}} \right)^{-1} - 1. \tag{7}$$

2.2 Yield criteria

A yield criterion expresses a relationship between the components of the stress tensor at the onset of the plastic deformation, i.e. yielding. In a uniaxial case, the yield point can be found from the stress-strain curve of a uniaxial tensile test; however, the problem is more difficult to define a criterion for a multiaxial case. The relationship between the stress state and the yield point (Y) is usually defined in the form of an implicit 'yield function':

$$\phi(\boldsymbol{\sigma}, Y) \leq 0, \tag{8}$$

with $\boldsymbol{\sigma}$ being the stress tensor, and Y is the yield point, commonly defined based on a simple test, e.g. tension, compression or shearing. The yield function must be *closed*, *smooth* and *convex*; the latter is commonly assumed to be valid for metals, which is not necessary in case of nonmetallic materials [15].

The yield function (Eq. 8) can be interpreted as a surface in the stress space, usually called the 'yield surface'. All points located inside of the surface ($\phi < 0$) belong to the elastic domain, and the points of the surface ($\phi = 0$) themselves are related to a plastic state. The domain outside of the yield surface ($\phi > 0$) is physically meaningless.

The yield functions can be divided into two categories regarding the targeted materials: (**a**) isotropic (such as Tresca (the lower bound) [16], twin-shear (the upper bound) [17], and the well-known von-Mises [18, 19] models) and (**b**) anisotropic; see, e.g., [11, 20, 21] for a thorough review on the topic. In the following, three of the yield functions for the anisotropic materials are presented; these are selected for their popularity in modelling sheet metal deformation processes as well as their availability in most of commercial FEM solver, e.g. MSC-Marc. It should be noted that these yield functions are presented for the plane stress condition (if possible), i.e. ($\sigma_{11} \neq 0$, σ_{22}, $\sigma_{12} \neq 0$, and $\sigma_{13} = \sigma_{23} = \sigma_{33} = 0$), which is the case for sheet metal forming processes.

2.2.1 Hill 1948: *Hill48*

One of the earliest and most famous anisotropic yield criteria was proposed by Hill [8, 22] as a quadratic function in the form of:

$$\phi(\boldsymbol{\sigma}, Y) = F\sigma_{22}^2 - G\sigma_{11}^2 + H(\sigma_{11} - \sigma_{22})^2 + 2N\sigma_{12}^2 - 2Y^2, \tag{9}$$

where F, G, H and N are constants specific to the anisotropy state of the material. It can be shown that Eq.9 is related to the uniaxial tests performed at three different directions of $\theta = 0°, 45°$ and $90°$ with [11]:

$$\phi(\boldsymbol{\sigma}, Y) = \sigma_{11} - \left(1 + \left(\frac{\sigma_0}{\sigma_{90}}\right)^2 - \frac{r_0 + r_{90}}{r_{90}(1 + r_0)}\right)\sigma_{11}\sigma_{22} + \left(\frac{\sigma_0}{\sigma_{90}}\right)^2 \sigma_{22}^2 + \frac{r_0 + r_{90}}{r_{90}(1 + r_0)}(2r_{45} + 1)\sigma_{12}^2 - 2Y^2. \tag{10}$$

Moreover, the function F_θ (Eq. 4) can be found as [20]:

$$F_\theta = F\sin^4\theta + G\cos^4\theta + H\cos^2 2\theta + \frac{1}{2}N\sin^2 2\theta, \tag{11}$$

which can be used to predict the directional stress and the r-value from:

$$\sigma_\theta = YF_\theta^{-1}, \tag{12a}$$

$$r_\theta = F_\theta\left(F\sin^2\theta + G\cos^2\theta\right)^{-1} - 1. \tag{12b}$$

Describing the anisotropy of metals by the Hill 1948 yield criterion has the advantage that its basic assumptions are not complicated, its parameters have physical meaning, and can be calibrated with very limited and simple experimental tests and measurements. These justify its wide use in practice; however, due to its simple form, it has some drawbacks as well. For example, its application is limited to materials forming four 'ears' in axisymmetric deep-drawing processes, while such behavior does no hold in practice.

2.2.2 Barlat 1991: *Yld91*

Barlat et al. [9] developed an extension of the phenomenological yield criterion proposed by Hershey [23] and Hosford [24], which was in the form of:

$$\phi(\boldsymbol{\sigma}, Y) = |S_1 - S_2|^m + |S_2 - S_3|^m + |S_3 - S_1|^m - 2Y^m, \tag{13}$$

where $S_{i=1,2,3}$ are the deviator principle stresses, and m is the exponent to be defined based on the crystal structure of the material: $m = 6$ for BCC and $m = 8$ for FCC materials [25].

Barlat et al. introduced shear stress terms in this model (Eq. 13) and proposed a general six-component yield function, which was expressed as follows after a complex number transformation:

$$\phi = (3I_2)^{m/2}\left\{\left|2\cos\left(\frac{2\theta + \pi}{6}\right)\right|^m + \left|2\cos\left(\frac{2\theta + 3\pi}{6}\right)\right|^m + \left|-2\cos\left(\frac{2\theta + 5\pi}{6}\right)\right|^m\right\} - 2Y^m, \tag{14}$$

where

$$\theta = \arccos\left(I_3/I_2^{3/2}\right), \tag{15}$$

with I_2 and I_3 being the second and third invariant of the stress determinant, to be written as:

$$I_2 = +\frac{\left(a\sigma_{22} - c\left(\sigma_{11} - \sigma_{22}\right)\right)^2 + \left(b\sigma_{11} + c\left(\sigma_{11} - \sigma_{22}\right)\right)^2 + \left(a\sigma_{22} + b\sigma_{11}\right)^2}{54} + \frac{(h\sigma_{12})^2}{3}, \tag{16a}$$

$$I_3 = -\frac{\left(a\sigma_{22} - c\left(\sigma_{11} - \sigma_{22}\right)\right)\left(b\sigma_{11} + c\left(\sigma_{11} - \sigma_{22}\right)\right)\left(a\sigma_{22} + b\sigma_{11}\right)}{54} + \frac{\left(h\sigma_{12}\right)^2\left(a\sigma_{22} + b\sigma_{11}\right)}{6}, \tag{16b}$$

where a, b, c and h are weighing coefficients to describe the anisotropy of the material, which can be identified using different methods [11].

Although the Yld91 was originally proposed in the form of Eqs.14 to 16, Yoon et al. [26–28] later showed that Yld91 is a special case of another yield criterion that Barlat et al. developed in 1996 [29] called as Yld96. According to the formulation of Yld96, the yield criterion Yld91 can be written in the form of Eq.13, with $S_{i=1,2,3}$ being the principle values of the the following matrix:

$$\boldsymbol{s} = \begin{bmatrix} s_{11} & s_{12} & 0 \\ s_{12} & s_{22} & 0 \\ 0 & 0 & s_{33} \end{bmatrix}, \tag{17}$$

with

$$s_{11} = \frac{1}{3}\left(c_2\sigma_{11} + c_3(\sigma_{11} - \sigma_{22})\right), \tag{18a}$$

$$s_{22} = \frac{1}{3}\left(c_1\sigma_{22} - c_3(\sigma_{11} - \sigma_{22})\right), \tag{18b}$$

$$s_{12} = c_6\sigma_{12}, \tag{18c}$$

$$s_{33} = -s_{11} - s_{22}, \tag{18d}$$

and

$$S_{1,2} = \frac{s_{11}+s_{22}}{2} \pm \sqrt{\left(\frac{s_{11}+s_{22}}{2}\right)^2 + s_{12}^2}, \tag{19a}$$

$$S_3 = s_{33}. \tag{19b}$$

Comparing this formulation and the original one, it is apparent that the weighting coefficient are renamed from (a, b, c, h) to (c_1, c_2, c_3, c_6). The advantage of this formulation is that its coefficients can be found easier compared to the original one.

2.2.3 Barlat 2005: *Yld2004-18p*

Barlat et al. proposed one of the most advanced yield functions in 2005 [10] based on the linear transformations of the deviatoric stress. This yield function is known as Yld2004-18p for its 18 adjustable parameters, which is defined as:

$$\phi(\boldsymbol{\sigma}, Y) = \sum_{i=1}^{3}\sum_{j=1}^{3}\left|\widetilde{S}_i^{'} - \widetilde{S}_j^{''}\right| - 4Y^m, \tag{20}$$

where m is the same as the one described in Yld91, and $\widetilde{S}_i^{'}$ and $\widetilde{S}_j^{''}$ are the eigenvalues of the transformed stress tensors $\widetilde{\boldsymbol{s}}^{'}$ and $\widetilde{\boldsymbol{s}}^{''}$, respectively. The transformed stresses are functions of the deviatoric stress tensor: $\boldsymbol{s} = \boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma})\boldsymbol{I}$. The transformed stresses are defined in the vector format as $\widetilde{\boldsymbol{s}}^{'} = \boldsymbol{c}^{'}\boldsymbol{s}$ and $\widetilde{\boldsymbol{s}}^{''} = \boldsymbol{c}^{''}\boldsymbol{s}$, which can be written in a full definition of:

$$\begin{Bmatrix} \widetilde{s}_{11}^{'} \\ \widetilde{s}_{22}^{'} \\ \widetilde{s}_{33}^{'} \\ \widetilde{s}_{12}^{'} \end{Bmatrix} = \begin{bmatrix} 0 & -c_{12}^{'} & -c_{13}^{'} & 0 \\ -c_{21}^{'} & 0 & -c_{23}^{'} & 0 \\ -c_{31}^{'} & -c_{32}^{'} & 0 & 0 \\ 0 & 0 & 0 & c_{66}^{'} \end{bmatrix} \begin{Bmatrix} s_{11} \\ s_{22} \\ s_{33} \\ s_{12} \end{Bmatrix}, \tag{21a}$$

$$\begin{Bmatrix} \widetilde{s}_{11}^{''} \\ \widetilde{s}_{22}^{''} \\ \widetilde{s}_{33}^{''} \\ \widetilde{s}_{12}^{''} \end{Bmatrix} = \begin{bmatrix} 0 & -c_{12}^{''} & -c_{13}^{''} & 0 \\ -c_{21}^{''} & 0 & -c_{23}^{''} & 0 \\ -c_{31}^{''} & -c_{32}^{''} & 0 & 0 \\ 0 & 0 & 0 & c_{66}^{''} \end{bmatrix} \begin{Bmatrix} s_{11} \\ s_{22} \\ s_{33} \\ s_{12} \end{Bmatrix}, \tag{21b}$$

with $c_{44}^{'} = c_{44}^{''} = c_{55}^{'} = c_{55}^{''} = 1$ in the case of plane stress. Boogard et al. showed that $c_{12}^{'}$ and $c_{13}^{'}$ can safely be set to unity [30]; however, the original formulation was used in this work.

3 Virtual laboratory for mechanical testing

As was mentioned in earlier sections, reliable simulations elasto-plastic materials requires the experimental data obtained from both uniaxial and biaxial tests, which can be very complicated to perform, specially in the case of small scale samples; therefore, the idea of performing virtual tests can be considered as an interesting alternative approach. Using a virtual lab, larger number of tests can be performed via a much easier procedure compared to the experimental setups; consequently, all required data would be available to calibrate more complicated yield criteria for a better definition of the material's plastic behavior.

The idea of the virtual lab is to describe the material on a smaller scale, i.e. micro (grain) scale for the current work, and calculate macroscopic properties, such as flow stress curve, r-values and yield points, using numerical homogenization[2]. To do so, the first step is to prepare a reliable RVE, which can describe the most important aspects of the microstructure. Moreover, a few limited tests, commonly tensile tests along the RD for sheet metals with one phase, should be performed for calibrating the single crystal plasticity material model. This set of RVE and calibrated material model can be considered as a Digital Twin (DT) of the material . Once the DT is prepared, it can be used as the sample of different virtual mechanical tests: uniaxial and biaxial tests are the relevant ones for describing the anisotropic constitutive behavior of a elasto-plastic sheet metals. The concept of the virtual lab and its process chain is schematically depicted in figure 5.

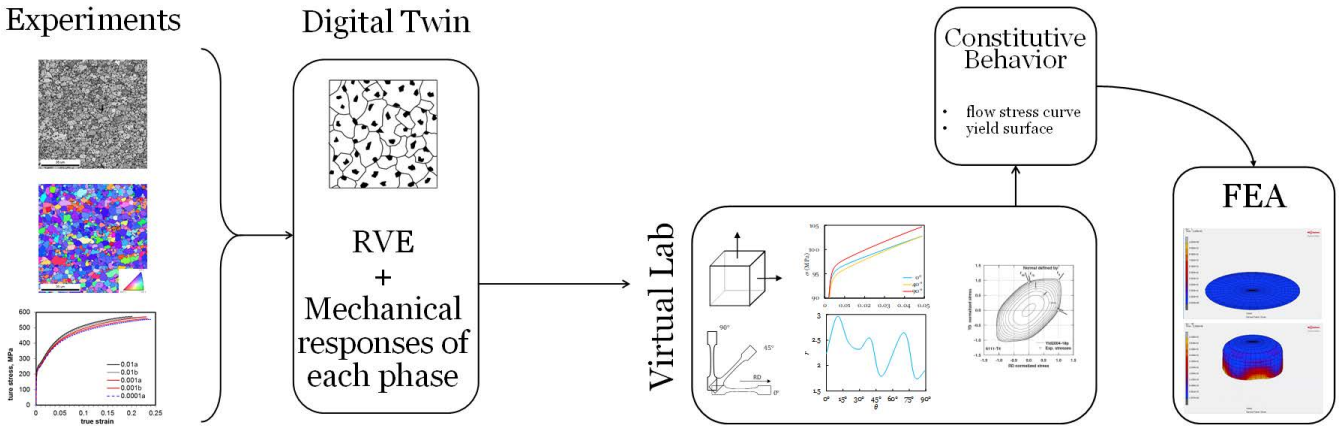


Figure 5: The concept of the virtual lab for mechanical testing. This figure shows the idea of generating Digital Twin (DT) of a material based on some limited experimental tests, and performing other mechanical tests inside of the virtual lab to obtain the data required for defining the mechanical properties of the material, which in turn can be used in FE analyses.

3.1 Uniaxial tests

One important set of mechanical tests is the uniaxial tension, mainly because the material can be deformed for large strains in this test. Moreover, some yield criteria can be calibrated using the results of a number of tensile tests. Furthermore, the r-values can be determined from these tests and be used for evaluation of yield criteria.

In order to perform uniaxial tests, the material is required to be sampled at different in-plane angles θ. Accordingly, this sampling method should be replicated for virtual testing as well. To do so, the common method is to collect the sample from within the RVE. As shown in figure 6, in this process some part of the RVE may be wasted, making it inefficient. To avoid wasting parts of the expensive DT, a new 3-step method has been developed by introducing a *master RVE*; figure 7 shows the procedure of sampling via this method. Following this procedure, no parts of the valuable DT would be thrown away.

3.2 Biaxial tests

Performing controlled biaxial tests are required for calibrating some of yield criteria, such as Yld2004-18p; however, making accurate measurements of the principal strains is not an easy task. Moreover, this experimental procedure is limited to the small straining degrees of less than 5% [11, 32]. Furthermore, investigating all directions of the stress space is not feasible, which can result in missing some unexplored yet critical data points. The idea of the biaxial test module of the virtual lab is to perform biaxial tests at any desired direction, flexible to focus on more important regions (if known). Moreover, the tests

[2] *Homogenization* is the process of finding the relationship between material models at different length scales. For transforming mechanical at the grain scale into macro scale, different schemes are commonly used, such as isostrain, grain cluster or full-field. The readers are referred to [2, 3, 31] for detailed discussion on the topic.

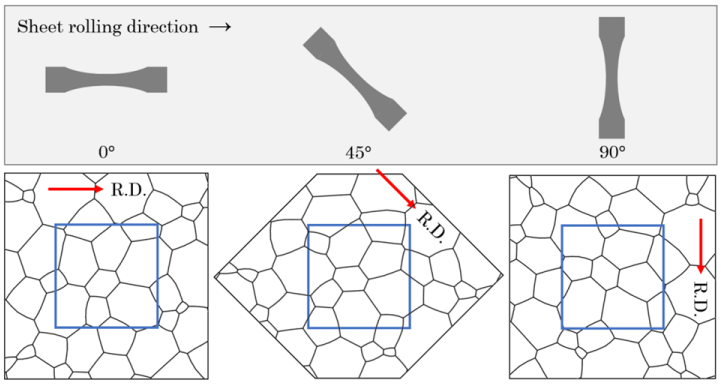


Figure 6: The common process of sampling of an RVE for uniaxial tensile tests, in which the sample is cut from a rotated RVE. This process mimics the process in an experimental setup [1].

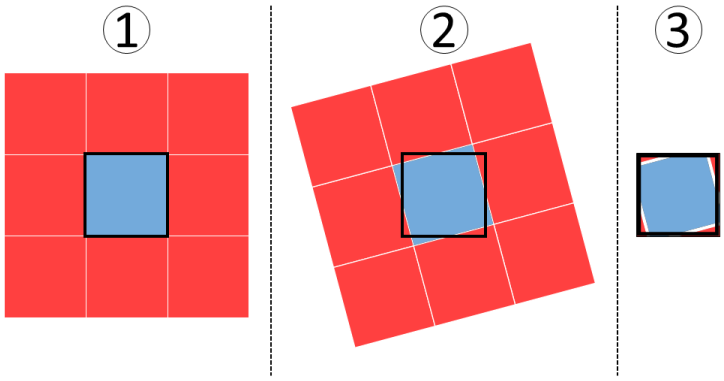


Figure 7: The DT is sampled along different in-plane angles in a 3-step procedure: (1) first, the original RVE is replicated to form a larger microstructure, called *master RVE*, (2) then, the master RVE is rotated at any required angle, and (3) the rotated test RVE, with the shape comparable to the original one, is sampled from the rotated master RVE.

of the virtual lab can be done with $\sigma_{12} \neq 0$. It should be noted that the ratio of $\sigma_{11} : \sigma_{22} : \sigma_{12}$ can be controlled only to the onset of plastic deformation; once the plasticity begins, neither the ratios nor the strain-rates[3] can be controlled.

3.3 Implementation

The virtual lab is designed and developed to deliver certain results from each of its modules, namely the directional flow stress curves (σ_θ) and anisotropy coefficient (r_θ) using the uniaxial testing module, and the stress tensor at the onset of plasticity using the biaxial testing one. To do so, the modules of the virtual mechanical testing laboratory are designed and developed to perform desired tests using a CP-based material model called DAMASK (the Düsseldorf Advanced Material Simulation Kit) [**DAMASK˙website**, 3, 31]. DAMASK can be implemented within a FEM solver, such as Abaqus or Marc, or to be worked using its own spectral solver. The virtual lab is written in MATLAB (The MathWorks, Inc., Natick, MA, USA (release 2018b)) to perform as a pre- and post-processing tool for the internal spectral solver of DAMASK. The code has been tested on a Linux machine. The MATLAB scripts of both modules are available in the appendix. The followings are general overviews of the steps within the design of each of the modules.

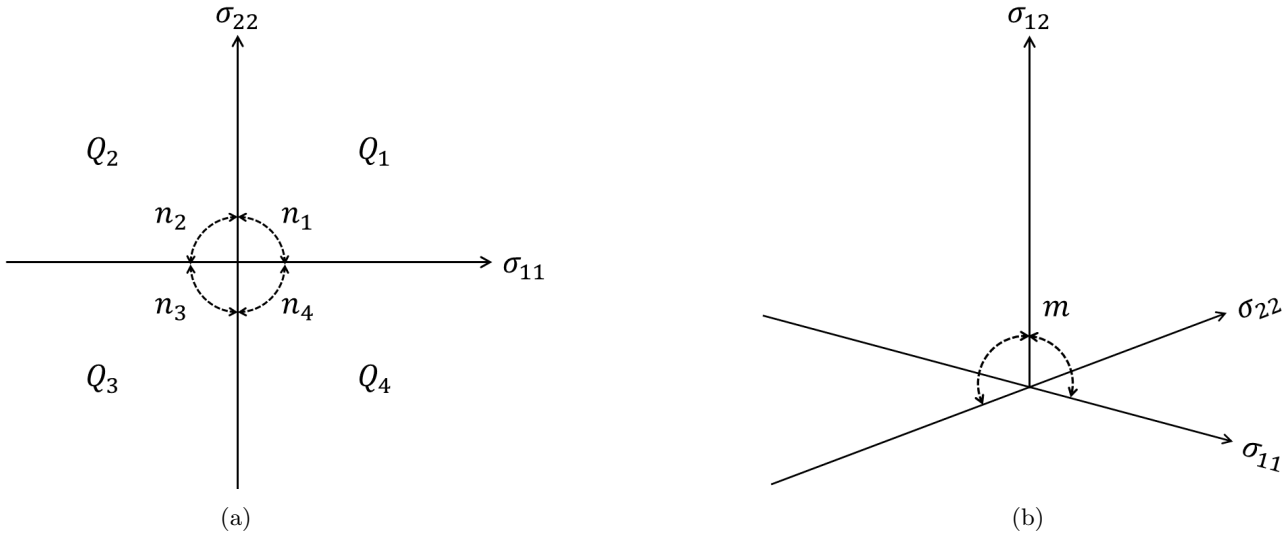
[3]The rate of deformations can be controlled by either velocity gradient (L) or deformation gradient rate ($\dot{F}$). In either case, the logarithmic strain(-rate) is not directly and fully under control.

Uniaxial Testing

- user inputs
  1. Digital Twin (RVE and the calibrated material model (compatible with DAMASK))
  2. The values of strain rate, deformation times, and the number exploring directions ($\theta \in [0, \pi/2]$). *NOTE:* it is advised to use two values of deformation time, mainly one for elastic deformation and the other for plastic one.
- pre-processing
  3. create the master RVE
  4. rotate the master RVE
  5. cut requested the rotated RVEs
  6. prepare the scripts for tests and analyses in DAMASK
- communication with DAMASK
  7. submit jobs
- post-processing
  8. collect the analyzed simulation results
  9. collect the directional flow stress data
  10. calculate the directional r-values
  11. report the results in a single file

Biaxial Testing

- user inputs
  1. Digital Twin (RVE and the calibrated material model (compatible with DAMASK))
  2. The number of exploring directions in the format of $[n_1, n_2, n_3, n_4, m]$, where n_i indicates the number of tests within each quadrant of the $\sigma_{11} - \sigma_{22}$ space, and m assigns the number of tests along the torsional σ_{12} space; see figure 8.
- pre-processing
  3. prepare the scripts for tests and analyses in DAMASK
- communication with DAMASK
  4. submit jobs
- post-processing
  5. collect the analyzed simulation results
  6. identify the initiation of plastic deformation (Y)
  7. collect the yield loci, i.e. the stress states correspond to Y
  8. report the collected yield loci in a single file



(a) (b)

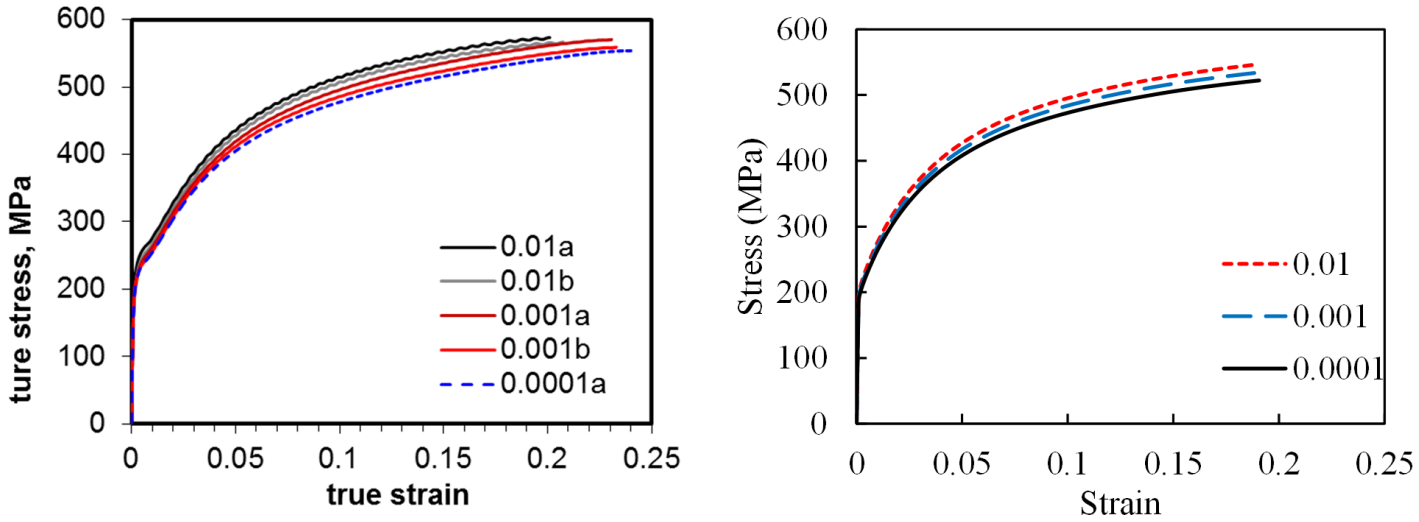
Figure 8: The biaxial module requires for the exploratory directions of the stress state in the format of $[n_1, n_2, n_3, n_4, m]$: the first four numbers indicate the number of tests within each quadrant of $\sigma_{11} - \sigma_{22}$ space (a), and the last one refers to the torsional stress space (b). It should be noted that the spaces would be divided equally according to the set values.

4 Case study: anisotropy of mechanical behavior of annealed 420 steel

A simplified DT of an annealed 420 steel was provided by TU-Delft to be tested with the virtual lab. The DT consisted of an RVE (with $10 \times 10 \times 10$ grids with approximately 500 grains) and the mechanical behavior of the ferrite phase. The carbide was not included in the DT. Moreover, the results of the stress-strain curves obtained from a number of uniaxial tests were provided as the supplementary data.

4.1 Uniaxial testing

The uniaxial tests were performed for 9 different directions with respect to the RD, by calling the uniaxial module of the virtual lab via `VL UT('RVE.geom',9)`; see appendix B. Figure 9 shows the stress-strain curves obtained from both experimental and virtual uniaxial tests at $\theta = 0°$, indicating a close comparison between the two sets of results. Moreover, figure 10 shows the calculated r-values at two different strains of 0.10 and 0.15.



(a) (b)

Figure 9: The stress-strain curves at $\theta = 0°$ for different strain rates obtained from (a) experimental and (b) virtual mechanical testing.

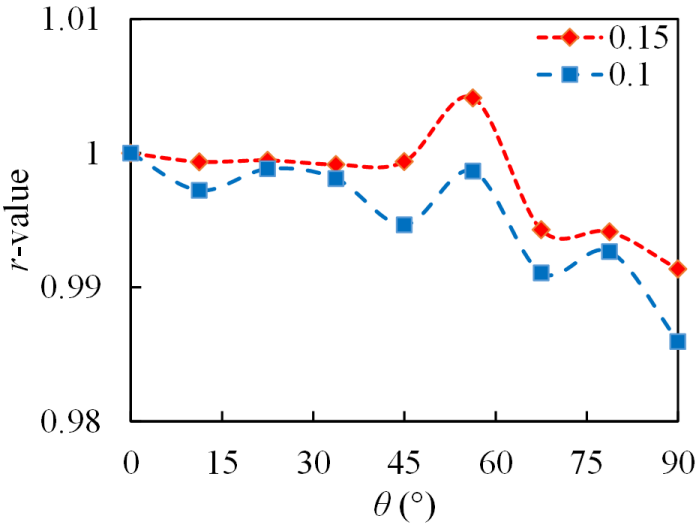


Figure 10: Variation of r-value as function of in-plane direction of the RVE with respect to RD. The two sets shows the calculated r-values at different strains of 0.10 and 0.15.

4.2 Biaxial testing

In total, 65 different biaxial tests were performed on the DT by calling the biaxial module of the virtual lab using the following command: `VL_BT('RVE.geom',[5,1,5,1,3])`; see section 3.3 and appendix C for further explanations. Figure 11 shows the identified yield loci. Using these data, three of the yield criteria, commonly used in metal forming industries (see section 2.2) are calibrated for the given DT.

In order to calibrate the yield criteria using the identified yield loci, the simplex search method of Largarias et al. [33] was used, which is a derivative-free method with no constraint on the variables. This method was called through the `fminsearch` function of MATLAB, and its detailed algorithm can be found on the MathWorks website [34]. The minimization process was performed on a cost function Δ in the form of $\Delta = \Sigma\left(\phi(\boldsymbol{\sigma}, 0) - \phi(\mathbf{0}, Y)\right)^2$, with Σ indicating a sum. In this cost function, the yield loci of the case with $\sigma_{11} > 0$ and $\sigma_{22} = \sigma_{12} = 0$ was selected for the Y value. The approximated values of the constant for each of the selected yield criteria are given in table 1, and figure 12 illustrates the fitted yield surfaces.

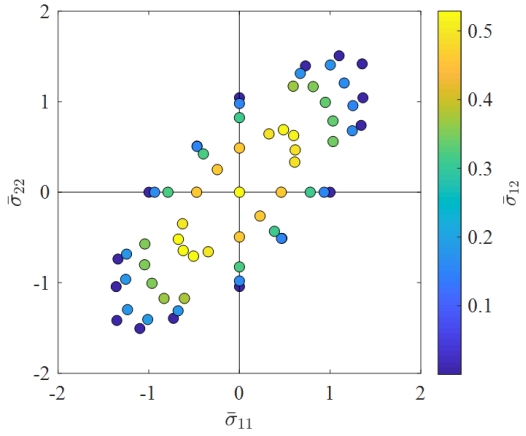
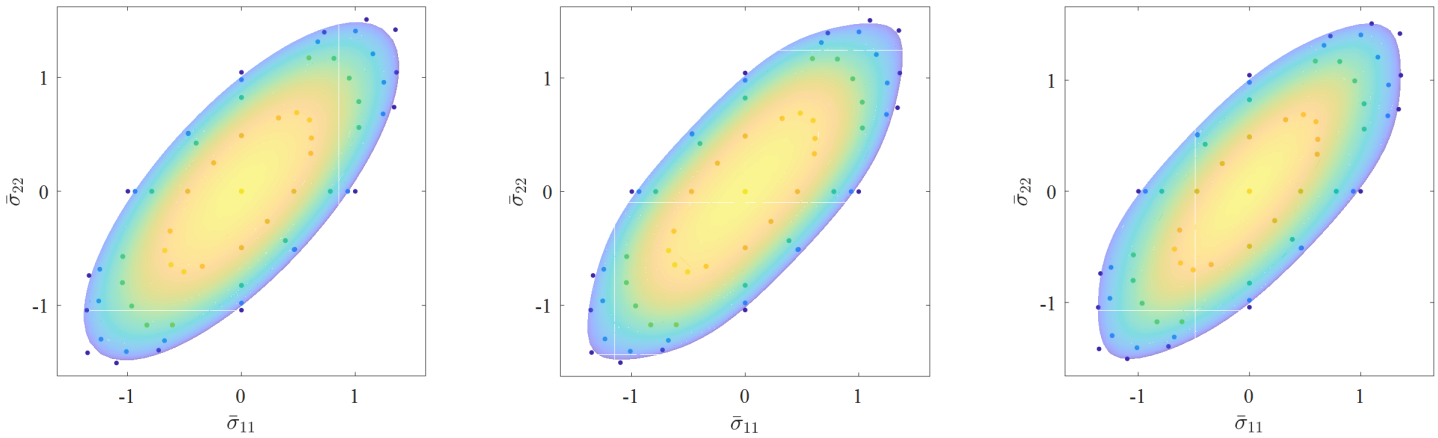


Figure 11: The identified yield loci at the first instance of yielding.



(a) *Hill48* (b) *Yld91* (c) *Yld2004-18p*

Figure 12: The visual representation of the calibrated yield criteria.

Table 1: The optimized constants of *Hill48*, *Yld91* and *Yld2004-18p*.

Hill48	F	G	H	N			
	3.5178	3.8055	-1.5488	3.2415			
Yld91	c_1	c_2	c_3	c_6			
	0.6544	0.8096	1.2510	1.0288			
Yld2004-18p	c'_{12}	c'_{13}	c'_{21}	c'_{23}	c'_{31}	c'_{32}	c'_{66}
	1.6953	0.7552	0.9244	0.3842	0.7418	1.7817	1.6055
	c''_{12}	c''_{13}	c''_{21}	c''_{23}	c''_{31}	c''_{32}	c''_{66}
	0.6261	0.2460	0.3179	0.9565	1.0757	1.5791	0.1042

Appendices

GENERAL DISCLAIMER:

The concept of the virtual lab for mechanical testing
and its implmenetation are developed and prepared by
Soheil Solhjoo, PhD, from University of Groningen (CMME).
The development of this virtual lab and its validations
are mainly based on the ITEA project 16010 called VMAP.

The developed virtual lab can be used for submitting jobs
to DAMASK (the Düsseldorf Advanced Material Simulation Kit)
for performing uniaxial and biaxial mechanical tests
to characterize the constituive behavior of sheet metals.

The performace, accuracy or correctness of the results
are not guaranteed, and the user should take all
responsibilities of running this code on their own.

A The required input files for running the virtual lab

In the appendices, the codes for both modules (uniaxial and biaxial) of the virtual lab are given, and the way of using them are described. For any further description of the formatting regarding DAMASK, the readers are referred to the documentation of DAMASK [35].

In order to test a digital twin using the virtual lab, two different files are required to be provided by the user: (1) an RVE, including the of the position of each grain, and (2) a data file, including the mechanical behavior of each phase. The RVE file, which is called *geometry* file in DAMASK, can have any arbitrary name, such as *RVE.geom*, and it has to provide the following sections:

- `grid`
- `size`
- `origin`
- `homogenization` *option*
- `microstructure #`
- *grain structure*
- `<microstructure>`
- `<texture>`

The data file for the mechanical behavior of the single grains has to be named *phase.data*, and it should be in the format that DAMASK can read. See A.1 and A.2 of the examples of these two files for 420 steel; with these two files, a digital twin of the material is defined. It is important to note that developed virtual lab assumes the rolling and transverse directions are along the x and y axes, respectively. Moreover, wherever required, the virtual lab uses the z axis for performing the rotations.

A.1 Geometry file

```
2610 header
<microstructure>
[Grain1]
crystallite 1
(constituent) phase 1 texture 1 fraction 1.0
[Grain2]
crystallite 1
(constituent) phase 1 texture 2 fraction 1.0
...

<texture>
[Grain1]
(gauss) phi1 55.366  Phi 121.305 phi2 224.862 scatter  0 fraction 1
[Grain2]
(gauss) phi1 261.077 Phi 64.515  phi2 235.893 scatter  0 fraction 1
...

grid a 10 b 10 c 10
size x 0.25 y 0.25 z 0.25
origin x 0.0 y 0.0 z 0.0
homogenization 1
microstructures 282
 98 174  12  12  12  64  84  84  98  98
 98  20  20 223  82  72  79  84  77  98
  2 223 223 223 223 223 170 151 151 200
200 223 223 223 223 223 151 151 151 200
 50  78  65 223 262 110 110 129 129  50
 50  65  65  21  21  31  31  31 207  50
207  55  55  21  21 138 138 138 207 207
...
```

A.2 phase.data

```
[BCC_Ferrite]
plasticity              phenopowerlaw
elasticity              hooke
lattice_structure       bcc
Nslip                   12 # per family
Ntwin                    0 # per family
c11                     380.6e9
c12                     220.2e9
c44                     192.5e9
gdot0_slip              0.001
n_slip                  100
tau0_slip               77e6 # per family
tausat_slip             220e6 # per family
gdot0_twin              0.001
n_twin                  20
tau0_twin               31.0e6 # per family
s_pr                    0 # push-up factor for slip saturation due to twinning
twin_b                  0
twin_c                  0
twin_d                  0
twin_e                  0
h0_slipslip             2500e6
h0_sliptwin             0
h0_twinslip             0
h0_twintwin             0
interaction_slipslip    1 1 1.4 1.4 1.4 1.4
interaction_sliptwin    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
interaction_twinslip    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
interaction_twintwin    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
a_slip                  1.6
atol_resistance         1
```

B Uniaxial Testing Module

The uniaxial tests are designed to be performed using the loading conditions of $\dot{\boldsymbol{F}} = \begin{bmatrix} \dot{F}_{11} & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & * \end{bmatrix}$ and $\boldsymbol{P} = \begin{bmatrix} * & * & * \\ * & 0 & * \\ * & * & 0 \end{bmatrix}$, where $\dot{\boldsymbol{F}}$ and $\boldsymbol{P}$ are the deformation gradient rate and first Piola-Kirchhoff stress tensors, respectively. The following is an example of how the uniaxial test along the x direction can be performed in DAMASK:

```
Fdot 0.01 0 0 0 * 0 0 0 * stress * * * * 0 * * * 0 time 1  incs 10 freq 1
Fdot 0.01 0 0 0 * 0 0 0 * stress * * * * 0 * * * 0 time 20 incs 90 freq 1 dropguessing
```

It should be noted that the loading conditions are set in two separate lines, one for a short amount of time and rather large number of increments to capture the elastic deformation, and the second one to approximate the plastic behavior.

The uniaxial module of the virtual lab can be called via: `VL_UT(RVE,N,r,[t],[n],f)`, where `RVE` is the name of the geometry file, `N` is the number of requested tests, `r` is the strain rate, `[t]` in the format of `[t1,t2,...,tn]` is the deformation time(s), `[n]` in the format of `[n1,n2,...,nn]` is the deformation increment(s), and `f` defines the frequency of the reports from DAMASK. Note that the number of elements in `[t]` and `[n]` has to be equal. Assigning the `RVE` is `mandatory`, but the rest have a default value of: `(N=3,r=0.01,[t]=[1,20],[n]=[10,90],f=1)`.

As a flexibility to the code, the parameters `r` and `f` are allowed to be assigned as `[r]` and `[f]`; however, if any of the two are defined as an array, they has to be with the same size as `[t]` and `[n]`.

The main code of this module is called *VL_UT.m*, which generates a bash file named *VL_UT.sh*. To perform the tests from within a Linux terminal, one can simply use a bash file such as follows:

```
#!/bin/bash
matlab -nojvm -nodesktop -r "VL_UT('RVE.geom'); exit;"
bash UT_VL.sh
```

Once all tests are performed, the results can be found located in the *master_results* directory and in a comma-separated values (CSV) file named *VL_UT.csv*, which has a data structure as shown in table 2.

All specific codes of the uniaxial module are given in the following sections. Moreover, the general codes, which are common between the two modules or have broader applications are given in appendix A.

Table 2: Data structure of *VL_UT.csv*. The values of $\boldsymbol{\sigma^*_{11}}$ are normalized with σ_{11} for the same ϵ_{11}. It should be noted that the values of $\boldsymbol{r}$ for $\epsilon_{11} = 0$ are not a number (type NaN).

	$\theta = 0$	$\boldsymbol{\theta \neq 0}$	$\boldsymbol{\theta}$
$\epsilon_{11}(i=1)$			
$\epsilon_{11}(i=2)$			
$\epsilon_{11}(i=3)$	σ_{11}	$\boldsymbol{\sigma^*_{11}}$	$\boldsymbol{r}$
...			
$\epsilon_{11}(i=n)$			

B.1 VL_UT.m

VL_UT is the starting code of the uniaxial testing module; it generates some other files, including a bash file *VL_UT.sh*, which should be called for submitting the jobs.

```
% MATLAB CODE!
```

B.2 UT_loading.m

UT_LOADING generates and saves the loading file (*UT.load*) according to the assigned values of `r`, `[t]` and `[n]`.

```
% MATLAB CODE!
```

B.3 create_master_geom.m

CREATE_MASTER_GEOM generates a master geometry file, which is a propagation of the original file on a 2D space by creating 8 mirror cells around the original one.

```
% MATLAB CODE!
```

B.4 cut_rotated_grain.m

CUT_ROTATED_GRAIN reads the "geometry" file of DAMASK, and rotate it around the z-axis. The angle of rotation is defined by the variable "rotate". A positive value rotate the system counterclockwise. This code also takes care of the periodicity of the grains along the XY directions. As the output, the code generates a new "geometry" file and saves the rotated data into it, named as *grains_rot.seeds*. Moreover, this code exports the sections ¡microstructure¿ and ¡texture¿ in separate "data" files.

```
% MATLAB CODE!
```

B.5 cut_config_builder.m

CUT_CONFIG_BUILDER identifies the "rotated master geometry" files, prepares a list accordingly, and puts them through two stages: (1) cutting via "cut_rotated_grain.m" and (2) building the required config files. Then, separate folders wil be generated for each test, and the corresponding files will be moved there. The input value of this function will be used for the initial stage of deformation.

```
% MATLAB CODE!
```

B.6 extract_rY.m

EXTRACT_rY: opens the collected results, and read them for their values of e11 and s11, i.e. strain and stress along the tensile direction, respectively. Moreover, it reads e22 and e33, for the calculation of $r = \epsilon_{22}/\epsilon_{33}$. The function saves the results in a file named *VL_UT.csv* in format as described in table 2.

```
% MATLAB CODE!
```

C Biaxial Testing Module

In general, the biaxial tests are controlled by assigning the deformation gradient rate ($\dot{\boldsymbol{F}}$) and first Piola-Kirchhoff stress ($\boldsymbol{P}$) as $\dot{\boldsymbol{F}} = \begin{bmatrix} \dot{F}_{11} & 0 & * \\ \dot{F}_{21} & \dot{F}_{22} & * \\ * & * & * \end{bmatrix}$ and $\boldsymbol{P} = \begin{bmatrix} * & * & 0 \\ * & * & 0 \\ 0 & 0 & 0 \end{bmatrix}$; however, if the tests are supposed to have some restrictive values of $\sigma_{ij} = 0$ (with ij being either 11, 22 or 21), the tensors $\dot{\boldsymbol{F}}$ and $\boldsymbol{P}$ would be updated by assigning their corresponding components to $\dot{F}_{ij} = *$ and $P_{ij} = 0$.

The biaxial module of the virtual lab can be called via: `VL_BT(RVE,[N],e_t,r,N_max,e_l,nv_flag,SF)`, where `RVE` is the name of the geometry file, `[N]` is the number of requested tests in the format of `[n1,n2,n3,n4,m]`, `e_t` is the targeted total strain, `r` is the overall deformation gradient rate, `N_max` sets the maximum number of tests, `e_l` is the user's guess of the strain limiting the deformation in the elastic range, `nv_flag` is a flag with values of for considering the *numerics.config* file (see [36] for further details.), and `SF` is a safety factor for estimating the deformation time: $t = (\texttt{e\_t} * \texttt{r}) * \texttt{SF}$.

For calling the biaxial module, assigning the `RVE` is `mandatory`, but the rest have a default value of: `([N]=[1,0,0,0,0],e_t=0.003,r=0.1,N_max=80,e_l=0.0025,nv_flag=1,SF=1.5)`. There are some further points to be noted:

- `e_t` and `r` work only as approximations, and that is why the safety factor `SF` is required. Moreover, assigning `r` is only for the convergence purposes; otherwise, the models used for describing elastic deformation are mainly rate-independent.
- if the designed number of tests are larger than `N_max`, then `N_max` number of the designed test would be selected randomly.
- `e_t` has to be larger than or equal to `e_l`

The biaxial module should be called from *VL_BT.m*, which calls other functions and generates bash file *VL_BT.sh*, that can be used to submit jobs to DAMASK and perform further post-processing procedures. This can be done by running the following bash file in a Linux terminal:

```
#!/bin/bash
matlab -nojvm -nodesktop -r "VL_BT('RVE.geom',[5,1,5,1,3]); exit;"
bash VL_BT.sh
```

Once the tests are done, the results can be found in the *master_results* directory and in a CSV file named *VL_BT.csv* containing 3 columns of data with a structure as described in table 3.

The following sections present the specific codes of the biaxial module; the general ones can be found in appendix A.

Table 3: Data structure of *VL_BT.csv*. All values are normalized by the first number of the table, except for itself.

σ_{11}	$\boldsymbol{\sigma}^*_{22}$	$\boldsymbol{\sigma}^*_{12}$
$\boldsymbol{\sigma}^*_{11}$		

C.1 VL_BT.m

VL_BT is the runner of the uniaxial testing module; it calls other functions and generates some files, including a bash file *VL_BT.sh*, which should be called for submitting the jobs.

```
% MATLAB CODE!
```

C.2 VL_BT_prepare.m

VL_BT_PREPAER makes the required files ready for further assignments. It calls functions *extract_microtex.m* (see C.3) and *combine_files* (see **??**) to prepare the necessary *material.config* file. Also, it calculates the deformation time and save it in a matlab file *time.mat*.

```
% MATLAB CODE!
```

C.3 extract_microtex.m

EXTRACT_MICROTEX reads the RVE (geometry file of DAMASK), and exports `<microstructure>` and `<texture>` lists in corresponding "*.data" files.

```
% MATLAB CODE!
```

C.4 BW_VL_BT.m

BW_BT_VL prepares all requested $\dot{\boldsymbol{F}}$ and $\boldsymbol{P}$ tensors, and pass them to two other functions (see C.5 and C.6).

```
% MATLAB CODE!
```

C.5 BW_state_builder.m

BW_STATE_BUILDER prepares and writes *state_builder.sh*. The generated bash file contains the list of simulations to be done.

```
% MATLAB CODE!
```

C.6 BW_loader_builder.m

BW_LOADER_BUILDER prepares and writes *loader.sh*, which submit simulations to DAMASK, perform the required post-processing, i.e. calculation of strain and stress, and make a copy of the results in the *master_results* directory.

```
% MATLAB CODE!
```

C.7 extract_YS.m

EXTRACT_YS opens the collected results of DAMASK, and read them for their values of total shear strain and stress. It estimates the instance of changing the deformation regime from elastic to plastic by studying the total shear strain via finding the maximum of its second derivative. Once the yielding is identified the corresponding stress state will be collected as a yield locus. The output is a text file called *VL_BT.csv*. See **??** for the data structure of the output file.

```
% MATLAB CODE!
```

D Common Codes

Some of the used codes are common between the two modules of the virtual lab, and have a broader range of applications. In this appendix, these functions are presented. Moreover, an external MATLAB function called `sort_nat.m` is required for running the virtual lab, which is accessible from the MathWorks website [37].

D.1 combine_files.m

COMBINE_FILES combines text files with an specified extension and export them to a single new file.

```
function combine_files(extension,export)
list = list_builder(['*.' extension]);
j = 1;
for i=1:size(list,1)
    file_name = [list{i},'.',extension];

    fid = fopen(file_name);
    tline = fgetl(fid);
    l = 1;
    while ischar(tline)
        data(l,1) = cellstr(tline);
        tline = fgetl(fid);
        l = l + 1;
    end
    fclose(fid);

    data_list(j,1) = cellstr(['<' list{i} '>']);
    data_list(j+1:j+size(data,1),1) = data;

    j = j + size(data,1) + 1;
    clear data
end
%% Export
fid = fopen(export,'w');
for i = 1:size(data_list,1)
    fprintf(fid,'%s\n',data_list{i});
end
fclose(fid);
end
```

D.2 list_builder.m

LIST_BUILDER generates a list of files with the specified extension.

```
function [names] = list_builder(files_extension)
files = dir(files_extension);
names = {files.name};
names = names';
% Continue if no file is identified
if isempty(names)
    return
end
% Corret the answer if only one file is found
just_one = 0;
if size(names,1)==1
    just_one = 1;
end
names = split(names,'.');
if just_one
    names(end) = [];
end
```

```
% Prepare the ouput
names = names(:,1);
names_temp = str2double(names);
if ~any(isnan(names_temp))
    names = sort(names_temp);
    names(isnan(names(:)))=[];
end
end
```

D.3 ind_finder.m

IND_FINDER finds the indices of specific labels in the header of DAMASK results.

```
function ind_is = ind_finder(header,required_labels)
% Find the columns for the required labels
[~,ind_which] = ismember(header,required_labels);
ind_is = find(ind_which~=0);
% Make sure that the identified columns are sorted correctly, compared
% to the defined "required_labels"
ind_which(ind_which==0) = [];
ind = sortrows([ind_which',ind_is'],1);
ind_is = ind(:,2)';
end
```

E Fitting procedure

The process of fitting the yield criteria on the identified yield loci is described in section 4.2. This appendix contains the MATLAB codes that are used for this purpose.

At the beginning, the *VL_BT.csv* file should be read, which can be done via the following commands:

```
y0 = csvread('VL_BT.csv');
% correct the first element -> 1
y0(1) = 1;
% distribute the values into properly named variables
s11 = y0(:,1);
s22 = y0(:,2);
s12 = y0(:,3);
% select the value of Y0 (i.e. Y in yield criteria)
Y0 = y0(1);
```

E.1 Hill84.m

HILL_48 fits the Hill48 yield criterion (see section 2.2.1) on a set of identified yield loci.

```
% MATLAB CODE!
```

E.2 Yld91.m

YLD91 can be used to calibrate the Yld91 yield criterion (details in section 2.2.2) for a given set of yield loci. The function needs a value of m as well for its input.

```
% MATLAB CODE!
```

E.3 Yld2004_18p.m

YLD2004_18p calibrates the Yld2004-18p yield criterion (described in section 2.2.3) using a set of yield loci. A value for m should be passed in the input argument of this function. The first two output variables of the function are called c_1 and c_2 each as an array with 7 element, which are corresponding to $\boldsymbol{c'}$ and $\boldsymbol{c''}$, respectively.

```
% MATLAB CODE!
```

References

[1] M. Kamps, "Microstructure based crystal plasticity modelling of macroscale material characteristics", MA thesis (2018).

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

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

[4] C. Zambaldi, Y. Yang, T. R. Bieler, and D. Raabe, "Orientation informed nanoindentation of α-titanium: indentation pileup in hexagonal metals deforming by prismatic slip", Journal of Materials Research **27**, 356–367 (2011).

[5] A. Koester, A. Ma, and A. Hartmaier, "Atomistically informed crystal plasticity model for body-centered cubic iron", Acta Materialia **60**, 3894–3901 (2012).

[6] D. Cereceda, M. Diehl, F. Roters, P. Shanthraj, D. Raabe, J. M. Perlado, and J. Marian, "Linking atomistic, kinetic monte carlo and crystal plasticity simulations of single-crystal tungsten strength", GAMM-Mitteilungen **38**, 213–227 (2015).

[7] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, and M. A. Bessa, "Deep learning predicts path-dependent plasticity", Proceedings of the National Academy of Sciences **116**, 26414–26420 (2019).

[8] R. Hill, "A theory of the yielding and plastic flow of anisotropic metals", Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **193**, 281–297 (1948).

[9] F. Barlat, D. J. Lege, and J. C. Brem, "A six-component yield function for anisotropic materials", International Journal of Plasticity **7**, 693–712 (1991).

[10] F. Barlat, H. Aretz, J. Yoon, M. Karabin, J. Brem, and R. Dick, "Linear transfomation-based anisotropic yield functions", International Journal of Plasticity **21**, 1009–1039 (2005).

[11] D. Banabic, in *Sheet metal forming processes* (Springer Berlin Heidelberg, 2010), pp. 27–140.

[12] W. T. Lankford, S. C. Snyder, and J. A. Bausher, "New criteria for predicting the press performance of deep drawing sheets", Transactions of the American Society of Metals **42**, 1197–1232 (1950).

[13] K. Siegert, *Drawing of automotive sheet metal parts (talat lecture 3705)*, 1996.

[14] Q. Yin, B. Zillmann, S. Suttner, G. Gerstein, M. Biasutti, A. E. Tekkaya, M. F.-X. Wagner, M. Merklein, M. Schaper, T. Halle, and A. Brosius, "An experimental and numerical investigation of different shear test configurations for sheet metal characterization", International Journal of Solids and Structures **51**, 1066–1074 (2014).

[15] A. W. Ganczarski and J. J. Skrzypek, in *Mechanics of anisotropic materials* (Springer International Publishing, 2015), pp. 133–157.

[16] `https://www.wikiwand.com/en/Yield_surface#/Tresca_yield_surface`.

[17] M.-h. Yu, "Twin shear stress yield criterion", International Journal of Mechanical Sciences **25**, 71–74 (1983).

[18] R. v. Mises, "Mechanik der festen körper im plastisch- deformablen zustand", Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse **1913**, 582–592 (1913).

[19] `https://www.wikiwand.com/en/Von_Mises_yield_criterion`.

[20] D. Banabic, ed., *Multiscale modelling in sheet metal forming* (Springer International Publishing, 2016).

[21] M.-H. Yu, *Unified strength theory and its applications* (Springer Singapore, 2018).

[22] `https://www.wikiwand.com/en/Hill_yield_criterion#/Quadratic_Hill_yield_criterion`.

[23] A. HERSHEY, "THE PLASTICITY OF AN ISOTROPIC AGGREGATE OF ANISOTROPIC FACE-CENTERED CUBIC CRYSTALS", JOURNAL OF APPLIED MECHANICS-TRANSACTIONS OF THE ASME **21**, 241–249 (1954).

[24] W. F. Hosford, "A generalized isotropic yield criterion", Journal of Applied Mechanics **39**, 607–609 (1972).

[25] R. W. Logan and W. F. Hosford, "Upper-bound anisotropic yield locus calculations assuming 111-pencil glide", International Journal of Mechanical Sciences **22**, 419–430 (1980).

[26] J. Yoon, D. Yang, and K. Chung, "Elasto-plastic finite element method based on incremental deformation theory and continuum based shell elements for planar anisotropic sheet materials", Computer Methods in Applied Mechanics and Engineering **174**, 23–56 (1999).

[27] J. Yoon, "A general elasto-plastic finite element formulation based on incremental deformation theory for planar anisotropy and its application to sheet metal forming", International Journal of Plasticity **15**, 35–67 (1999).

[28] J. Yoon, F. Barlat, K. Chung, F. Pourboghrat, and D. Yang, "Earing predictions based on asymmetric nonquadratic yield function", International Journal of Plasticity **16**, 1075–1104 (2000).

[29] F. Barlat, Y. Maeda, K. Chung, M. Yanagawa, J. Brem, Y. Hayashida, D. Lege, K. Matsui, S. Murtha, S. Hattori, R. Becker, and S. Makosey, "Yield function development for aluminum alloy sheets", Journal of the Mechanics and Physics of Solids **45**, 1727–1763 (1997).

[30] T. van den Boogaard, J. Havinga, A. Belin, and F. Barlat, "Parameter reduction for the yld2004-18p yield criterion", International Journal of Material Forming **9**, 175–178 (2015).

[31] F. Roters, P. Eisenlohr, C. Kords, D. Tjahjanto, M. Diehl, and D. Raabe, "DAMASK: the düsseldorf advanced MAterial simulation kit for studying crystal plasticity using an FE based or a spectral numerical solver", Procedia IUTAM **3**, 3–10 (2012).

[32] K. Pöhlandt, D. Banabic, and K. Lange, "Equi-biaxial anisotropy coefficient used to describe the plastic behavior of sheet metal", in Proc. 5th esaform conference, krakow (2002), pp. 723–727.

[33] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder–mead simplex method in low dimensions", SIAM Journal on Optimization **9**, 112–147 (1998).

[34] `https://www.mathworks.com/help/matlab/math/optimizing-nonlinear-functions.html#bsgpq6p-11`.

[35] `https://damask.mpie.de/bin/view/Documentation/WebHome`.

[36] `https://damask.mpie.de/Documentation/NumericsConfig`.

[37] `https://www.mathworks.com/matlabcentral/fileexchange/10959-sort_nat-natural-order-sort`.