Documentation for MatFlow workflow RVE_extrusion.

immediate

July 9, 2024

Contents

1 Abstract

Crystal plasticity offers predictions for material failure at the microscopic scale but proper validation of parameters is essential for correct predictions. Here a method for parameter calculation and then validation is presented. It goes along with this study [ref]. Where an in-situ high resolution digital image correlation (HRDIC) experiment as a direct comparison with a CP model generated with open source simulation software MatFlow [1] using an electron back scatter detraction (EBSD) map, comparing slip activation order and quantity, strain localisation, position and magnitude. We found literature values were irrelevant, values found by macroscopic validation gave a local strain match averaging with ±25% and macroscopic strain match ±8%, and values found from microscopic validation gave —fill in later —-. Future improvements need to be done relating subsurface microstructure and uniquely defining the saturation limits for CRSS values.

2 Introduction

2.1 Crystal plasticity summary

Text in Quotation mark are 'technical terms' and text in italics are *non technical terms used for explaining*.

Crystal plasticity is the study of how crystals permanently deform. Understanding how metals and other crystalline materials deform is important for calculating formality limits, crack initiation criteria and yielding criteria. Crystal plasticity simulations can be used to investigate a wide range of parameters, such as different load modes, load direction and load magnitude on many different microstructural parameters such as different grain size, shape and texture. One of the main benefits of simulations is, it is cheaper and quicker than similar experiments. To test experimentally many different parameters requires a lot of material which is expensive and sometimes infeasible to obtain.

DAMASK [2] is a crystal plasticity fast Fourier transform solver (CPFFT), it works by applying directional properties to single crystals and extending the stress-strain response from one grain to the next grain. For each increment of strain (in strain controlled simulations) two errors must be minimised for the convergent at each increment. The stress boundary condition which is the mean stress from each grain stress match the mean over all stress. The divergence find out what divergences mean. The solver in DAMASK can look at strain variation across a single grain because a grain has many 'material points' *like pixels* and at each material point the equations are solved. This must be done iteratively because plasticity is non-linear. The increments then repeat until the desired total strain has been reached.

Material science is a multi-level problem and the solvers used in this work are to investigate length scales of around 10um - 1mm. The law can be changed to look at continuum slip, discrete slip and stress concentrators, that is it can be used to look at smaller or bigger length scales.

MatFlow [1] is a computational frame work written in python. That links many different softwares together. In this report MatFlow links DefDap [3], Formable [4] and DAMASK, into an easy to use '*.ymal*' (text) file, where the *.ymal* file describes a 'workflow'. Within a workflow a number of tasks are defined; these tasks are modular and can easily be swapped and changed for different tasks. It's main benefits are in repeatability and ease of use. Without MatFlow you (the user) would have to interacted with each software individually, this makes it difficult to go back and change something and it is easy to loose information along the way, a clear example is that in different software the XYZ axis are in different directions, (python - X to the right, Y down, Z into page/ DAMASK - X to the right, Y up, Z out of page) additionally you need to be trained in all these different software before you can start doing any useful work, slowing down your progress.

The workflow used in this report is called $RVE\_extrusion$. Here we give an example for titanium alloy Ti-6Al-4V, but this frame work can be used for any crystalline material.

3 Method

3.1 Phenomenological law

The work hardening equation used in this report has been taken from DAMASK [2]. The equations explains how slip in a all the slip systems for a single crystal becomes more difficult. It is a phenomenological constitutive law, meaning it is not physics based and is defined in equation 1:

$$\dot{\xi} = h_0^{s-s}(1 + c_1(f_{tw}^{tot})^{c_2})(1 + h_{int}^{\alpha}) \sum_{\alpha'=1}^{N_s} |\dot{\gamma}^{\alpha'}| \left| 1 - \frac{\xi^{\alpha'}}{\xi_{\infty}^{\alpha'}} \right|^{a} sgn\left(1 - \frac{\xi^{\alpha'}}{\xi_{\infty}^{\alpha'}}\right) h^{\alpha\alpha'} + \sum_{\beta'=1}^{N_{tw}} \dot{\gamma}^{\beta'} h^{\alpha\beta'} \tag{1}$$

In equation 1: h_0^{s-s} is the slip-slip interaction matrix; c_1 and c_2 are both computational fitting parameters; f_{tw}^{tot} is the total twin volume fraction; h_{int}^{a} is slip twin interaction matrix; N_s is the total number of slip systems; α refers to specific slip system; $\dot{\gamma}^{\alpha'}$ is the shear strain rate on a slip system; $\xi^{\alpha'}$ is the current CRSS and $\xi_{\infty}^{\alpha'}$ is the final CRSS; $h^{\alpha\alpha'}$ is the slip system latent hardening matrix; N_{tw} is the number of twin system; β is a specific slip system; $\dot{\gamma}^{\beta'}$ is the is the shear strain rate on a twin system; $h^{\alpha\beta'}$ is the latent hardening between the slip and twin systems;sgn refers to the sign function.

In this equation many of the terms are related to twinning. In the case of titanium alloy Ti-6Al-4V (Ti64) twinning can be assumed to not occur, so the equations 1 and simplifies to equation 2 and because $f_{tw}^{tot} = 0$, $h_{int}^{\alpha} = 0$, $N_{tw} = 0$. This phenomenological constitutive law works by increasing/decreases (depending on the $sng()$ function) the slip system resistances (CRSS) along a curve that fits with the curvature of a in a way that then tends to some final value. The influence this has on the single crystal plastic stress-strain response is shown in Figure 1 and 2.

$$\dot{\xi} = h_0^{s-s} \sum_{\alpha'=1}^{N_s} |\dot{\gamma}^{\alpha'}| \left| 1 - \frac{\xi^{\alpha'}}{\xi_{\infty}^{\alpha'}} \right|^{a} sgn\left(1 - \frac{\xi^{\alpha'}}{\xi_{\infty}^{\alpha'}}\right) h^{\alpha\alpha'} \tag{2}$$

The equation in 2 can be schematically shown in Figure: 1 and Figure 2.

$$\dot{\gamma}^{\alpha} = (1 - f_{tw}^{tot})\dot{\gamma}_0^{\alpha} \left| \frac{\tau^{\alpha}}{\chi^{\alpha}} \right|^{n} sgn(\tau^{\alpha}) \tag{3}$$

Here in equation 3, the flow rule can be seen. This equation describes how the plastic shear rate develops and $\dot{\gamma}_0^{\alpha}$ describes the initial plastic shear rate, τ^{α} describes the shear stress and n is stress exponent. This equation involves the interaction that twins have on slip. So again it can be simplified to equation 4.

$$\dot{\gamma}^{\alpha} = \dot{\gamma}_0^{\alpha} \left| \frac{\tau^{\alpha}}{\chi^{\alpha}} \right|^{n} sgn(\tau^{\alpha}) \tag{4}$$

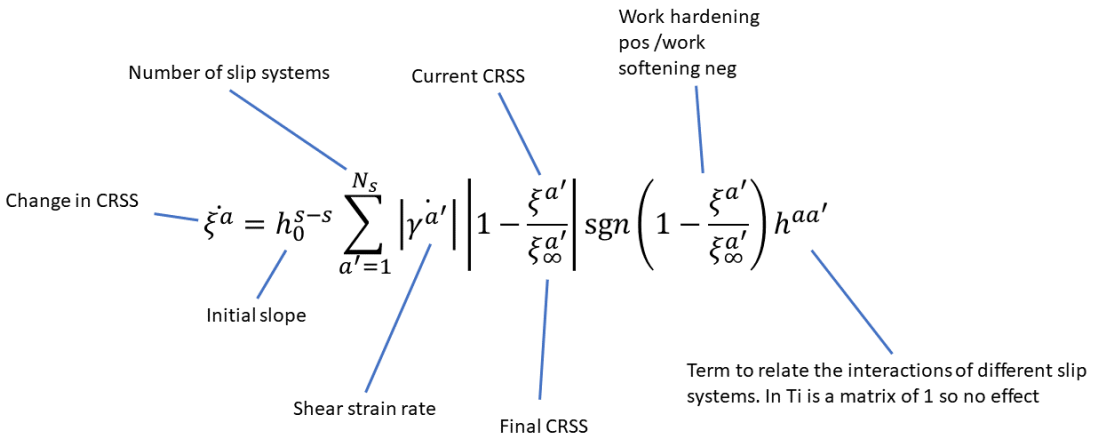

Figure 1: Annotation of the work hardening law as given in equation 2.

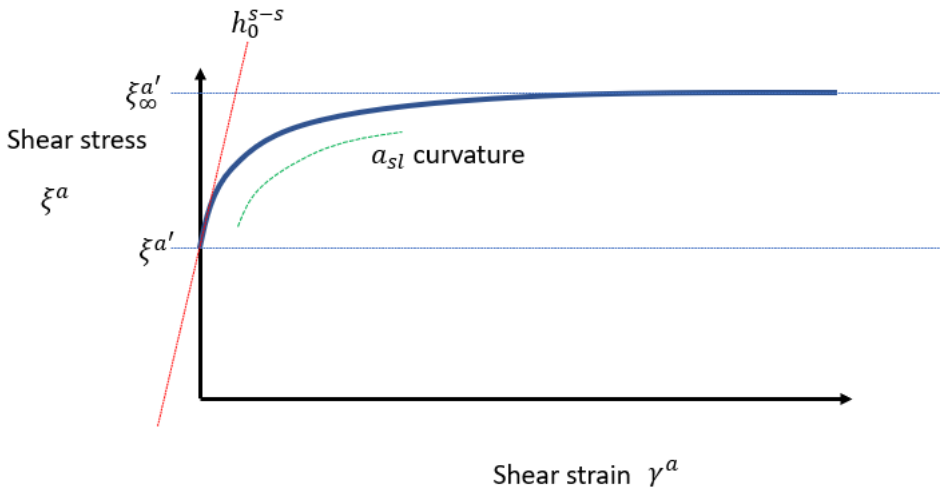

Figure 2: Simple explanation of how the monotonic hardening law influences the plastic stress-strain response for one slip system in a single crystal.

3.2 Preprocessing

When deforming the volume element, the unindexed phases create issues within the simulation, as there are no assigned material parameters and no deformation is possible. Therefore the unindexed phase needs to be removed before the model can be ran.

3.2.1 Cropping

The EBSD map must be square/rectangular, if you have image to the edge of a sample the edges of the EBSD map must be cropped. This should be completed in external software such as Aztec crystal or MTEX [5].

Make sure to crop a larger area than your region of interest.

Periodic boundary conditions effect the edge points in the simulation. You need to add additional material points equal to either 5% of your ROI or 5 grains (which ever is smaller) for all in plane directions.

3.2.2 Nearest neighbour method

One possible method of removing unindexed phase is the nearest neighbour method.

Each pixel has surrounding neighbours or nearest neighbours, all non edge pixels have 8 nearest neighbours. There are two possibilities, either all 8 nearest neighbours do or do-not have the same orientation. If all 8 nearest neighbours do have the same orientation, the unindexed pixel then takes the orientation of the nearest neighbours. If all 8 nearest neighbours do not have the same orientation, the code checks to see if 7 nearest neighbours have the same orientation. At which point the process is repeated for 7 nearest neighbours down to 1 nearest neighbours. This can be done in AZtec crystal or MTEX.

3.3 Processing

In section 7 the code of this workflow can be seen.

3.3.1 resources

These simulations require large memory to run, it is import to therefor assign higher memory cores to complete the simulation.

3.3.2 *load_microstructure_EBSD*

The root to folder path is placed (make sure forward slash and no spaces). Then the file name in CPR format is placed next to $filename$, make sure the file name has no spaces.

You can scale the spatial resolution of the simulation by the $scale\_factor$, greater than 1 will result in higher spatial resolution. Mainly useful if you have large EBSD maps that need shrinking.

You can 'flip' the image which means a 180degree rotation in plane. E.g. if you took the EBSD map upside down relative to the HRDIC data.

To create the microstructure, the workflow uses the software DefDAP [3] to spatially map orientations from an EBSD map into a 2D array within DefDAP. An algorithm is used to group similarly orientated pixels into grains. The term similarly orientated is defined by *boundary_tol* and is measured in degrees, e.g. 2degree. The average orientation of the pixel within a grain is calculated and turned into a 2D array. If comparing to HRDIC analysis make sure these values are the same. An EBSD map example can be seen in figure 3.

Minimum grain size for a grain measured in pixels. Any grain that has fewer pixels than this value (e.g. 50) will be dissolved into the surrounding grains.

3.3.3 *generate_volume_element_extrusion*

The first stage is to extrude the 2D array into a 3D model with dimensions (X $\times$ Y $\times$ Z). Where the length in the Z-axis is equal to *depth*. The volume element has properties that are defined within the model and are those of Ti64 these can be seen in Table 1 and are asigned to the material through the *phase_label*. Finding parameters for your material can be done by looking at section 4.

3D models have benefits compared to 2D models as the stress state can be expressed in 3D, and can include out of plane deformation but there are limitations, such as this 2D extruded microstructure is not physically representative of 3D microstructure. Some authors have tried to address subsurface effects [6]. This method does not currently address this issue (work is being done but is not complete), on average subsurface effects the predictions somewhere between by 10-25%.

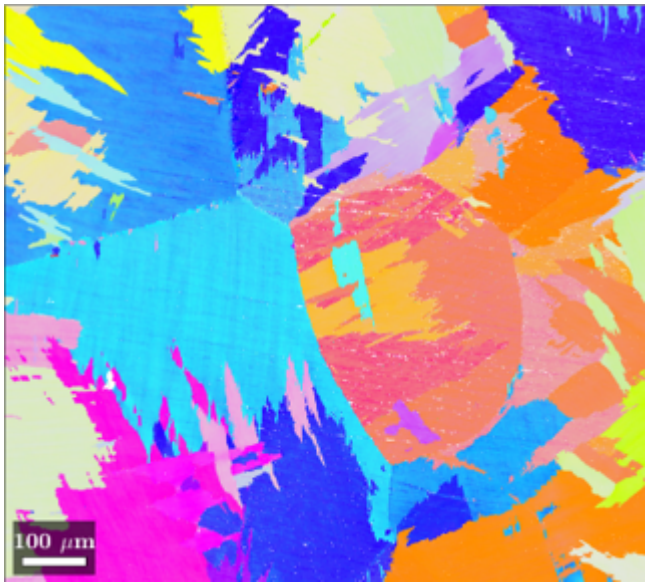

Figure 3: EBSD map of a Ti64 sample.

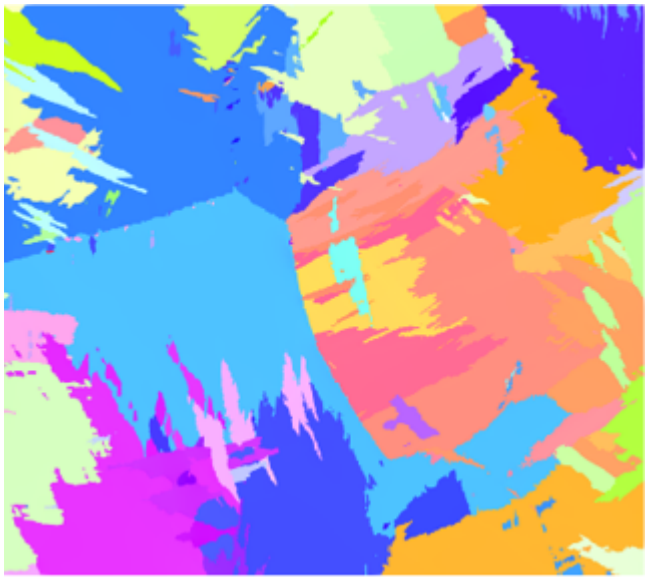

Figure 4: A 2D volume element of the EBSD map.

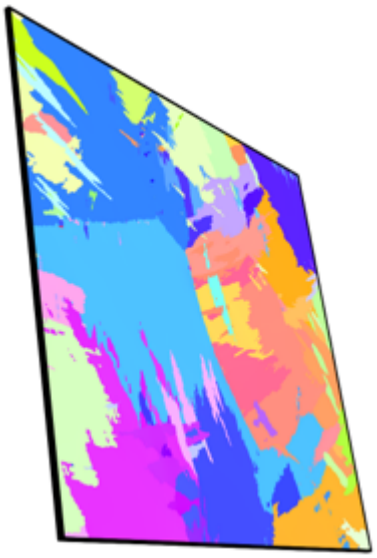

Figure 5: 3D volume element extruded and buffer layer added.

3.3.4 *modify_volume_element*

Periodic boundary conditions means surface voxels interact with the opposite side. In-plane isotropic *buffer_zones* are added to try and smooth out this effect. There properties are represented with the *Ti_iso* phase_label are equal to the bulk properties, e.g. Youngs modulus and shear modulus.

To allow for out of plane deformation, the out of plane buffer zone have properties of a very compliant material, to represent air. We are interested in out of plane deformation as when pairing with HRDIC, experimentally we measure the free surface and we want to reproduce that [7].

3.3.5 *visualise_volume_element*

The volume element is visualised by creating a file compatible with the visualisation toolkit (VTK) [8] using the software DAMASK [2]. This *.vtk* file can be viewed in external software such as ParaView [9]. This is an optional step mainly used for display the simulation.

The visualisation is done with another ymal file.

3.3.6 *simulate_volume_element_loading*

A load case is generated, for example plane strain, which can be seen in equation 5. In the experimental work, uniaxial deformation is applied. In the simulated work, uniaxial deformation could be applied, but with large simulation and the addition of the buffer zones uniaxial deformation can cause convergence issues. Instead plane strain is applied where deformation is in the X axis and no deformation is in the Z (*direction* : xz). The buffer zones being very compliant still allow for out of plane deformation for the Ti64 phase, while helping with convergence.

$$\mathbb{E} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & 0 \\ \varepsilon_{21} & \varepsilon_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5}$$

By using the Formable [4] software, an open source python package to provide formability analysis. Where positive strain represents tension and negative represents compression. In the case of a strain loading regime, the simulation will be split into small steps of strain increment *total_time* with each step having displacement equal to *target_deg_grad_rate*. Where *num_increments* is a numerical parameter, you can increase this if you have difficulties with convergences. This can be visualised in figure 6.

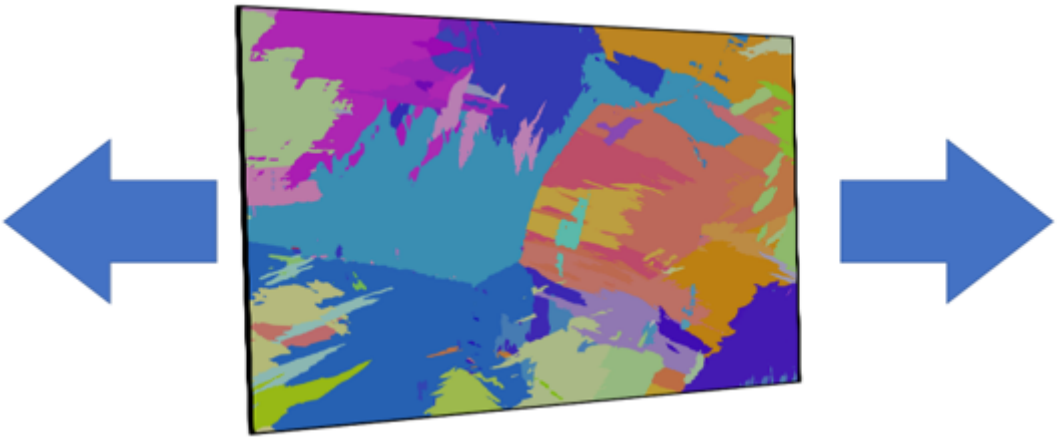

Figure 6: Plane strain applied load.

3.4 Post-processing

After the simulation has finish you must zip the file on the cluster $matflowzip * file\_name*$. You can then download the zipped file locally. I recommend using a large external hard-drive (1-2TB) as simulations outputs are around 5GB for one simulation.

The post-processing analysis is completed in python, and the jupyter notebooks are available at – github –.

3.5 Crystal plasticity parameters

An example set of parameters used for Ti64 with a lamellar microstructure are shown in Table: 1 The values have been taken from macroscopic stress-strain curve fitting.

Phenomenological crystal elasticity - α Ti64				
Parameter	Value	Units	Meaning	References
C_{11}	162.4	GPa	Stiffness	[10]
C_{12}	92	GPa	Stiffness	[10]
C_{13}	69	GPa	Stiffness	[10]
C_{33}	180.7	GPa	Stiffness	[10]
C_{44}	49.7	GPa	Stiffness	[10]
Phenomenological crystal plasticity - α Ti64				
Parameter	Value	Units	Meaning	References
ξ^{1-3}	390	MPa	Current basal CRSS	[10]
ξ_{∞}^{1-3}	400	MPa	Finial basal CRSS	ξ^{1-3}+10MPa
ξ^{4-6}	468	MPa	Current prismatic CRSS	[10]
ξ_{∞}^{4-6}	478	MPa	Finial prismatic CRSS	ξ^{4-6}+10MPa
ξ^{6-12}	663	MPa	Current pyramidal CRSS	[10]
ξ_{∞}^{6-12}	673	MPa	Finial pyramidal CRSS	ξ^{6-12}+10MPa
N_s	12	N/A	Number of slip systems	[11]
α	1-12	N/A	Slip system being considered	[11]
h_0^{s-s}	190	MPa	Initial work hardening rate	[12]
$\dot{\xi}$	variable	MPa	current RSS	N/A
$\dot{\gamma}^{\alpha'}$	variable	N/A	Shear strain	N/A
$\dot{\gamma}_0^{\alpha}$	0.001	s^{-1}	Initial plastic shear rate	[10]
τ^{α}	variable	MPa	Shear stress	N/A
n	15	N/A	Stress exponent	[10]

Table 1: Crystal plasticity parameters for the phenomenological work hardening law

4 Choosing crystal plasticity parameters.

4.1 Elastic

The single crystal elastic constants for the stiffness tensor can be taken from the literature for your alloy. These parameters do not change with microstructure or texture, and only change significantly with large changes in chemical composition. Therefore similar alloys from the literature are relevant.

If non are available, experiments will need to be done, one method is available here [13]. Many other methods exist in the literature.

It is possible to obtain CRSS ratios from the literature [14]. This is useful for reducing the level uncertainty.

4.2 Plastic

4.2.1 Literature values

The plastic parameters in the literature for your material can be useful to get the correct order of magnitude, **BUT** parameters such as CRSS change with processing route and so therefore exact value change between billets and therefore literature values are NOT relevant.

4.2.2 Macroscopic calibration

The next step is to fit your simulated data to experimental stress strain curves using the MatFlow workflow *single_crystal_parameter_fitting*. Document for this workflow can be found –elsewhere– . Briefly, an iterative process of running a simulation and a tensile test, then compare the true-stress true-strain curves. If they don't match change the parameters until they do match. This can be shown in figure7. If you do this for a range of orientations (e.g. RD and 45degrees) then that will help to validate your model.

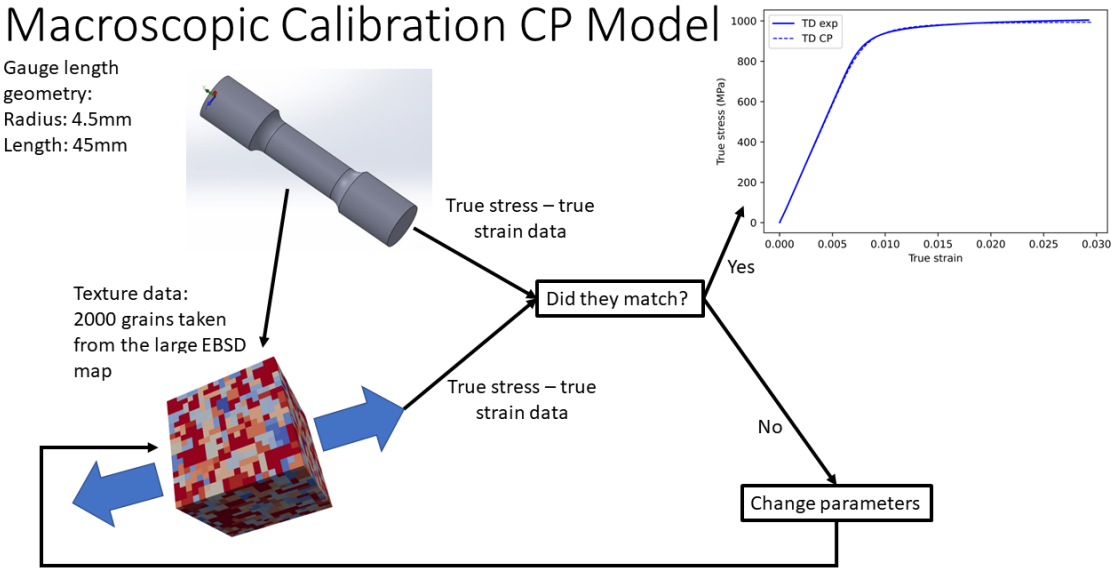

Figure 7: Macroscopic calibration using tensile true stress-stress data.

The parameter a_sl is the inverse of the strain rate dependence, in Ti-6Al-4V the strain rate dependence is 0.01 and therefore a_sl is equal to 100. This is related to microstructure/texture/alloy?

The parameter n_sl is from –not sure–.

Assumptions can then be made assuming the ratio between CRSS values for different slip systems [13] i.e. the yielding criteria. For Ti-6Al-4V, relative to basal: Basal=1, Prismatic =1.2 and pyramidal =1.7. The upper limit for the yielding CRSS values is half of the yield stress for the weakest system e.g. Yield Stress = $850MPa$, Basal < $425MPa$.

The pheno power law in section 3.1, has 10 variables even when excluding twinning. Stress strain curves only contain 2 variables therefore solutions are non-unique. Even using all the assumptions given in section 4.2.2 which reduces the variables to 5 more information is needed. HRDIC offers a method for obtaining more information.

4.2.3 Microscopic validation

Using in-situ HRDIC you can measure when slip start to occur relative to an applied stress. Then using slip trace analysis you can measure what slip system has been activated. You can then find the Schmid factor and times it by the applied stress as seen in equation 6 to approximate the CRSS values. Although, CRSS is actually equal to Schmid factor times resolved shear stress it should give an approximate CRSS values.

$$CRSS = Schmidfactor \times Engineeringstress \tag{6}$$

Currently there is no method implemented to measure the hardening - this is an important step and more work needs to be done to complete it.

4.3 Buffer Zone

4.3.1 Air

The Air phase should have stiffness ×1000 less than the bulk material and very weak strength but allow for some plasticity [7].

4.3.2 Iso_phase

The isotropic phase should only have elastic parameters. For your material you should look for bulk response values for Young's modulus and Poisson's ratio. If you are having trouble with this phase causing unstable stress state at the interface, just remove the phase from the simulation.

5 Pairing with High resolution Digital image correlation

Juptyer notebooks are avaible here [ref]

6 Summary

7 Code

```
001resources:
002  any:
003    scheduler: sge
004    scheduler_args:
005      shebang_args: --login
006      options: ["-l mem512"]
007
008
009tasks:
010- schema: load_microstructure_EBSD
011  inputs:
012    root_path: /mnt/iusers01/jf01/mbgm5pc3/EBSD_scans/year1_HRDIC_2
013    scaling_factor: 1
014    EBSD:
015      filename: A3_zone3
016      flip_vert: false   # optional
017      boundary_tol: 2  # optional
018      min_grain_size: 50   # optional
019
020- schema: generate_volume_element_extrusion
021  inputs:
022    depth: 6
023    image_axes: [y, x]
024    homog_label: SX
025    phase_label: Ti
026
027- schema: visualise_VE_VTK
028
029- schema: modify_VE_add_buffer_zones
030  inputs:
031    buffer_sizes: [4, 4, 4, 4, 2, 2]   # size of buffer on each face [-x, +x, -y, +y, -z, +z
032    phase_ids: [1, 1, 1, 1, 2, 2]   # phase of each buffer. Relative, so 1 is the first new
033    phase_labels: [Ti_iso, Air]       # labels of the new phases
034    homog_label: SX
035    order: [z, y, x]         # order to add the zones, default [x, y, z]
036
037- schema: visualise_VE_VTK
038
039- schema: simulate_VE_loading_damask
040  resources:
041    main:
```

```
042      num_cores: 10
043    output_file_parser:
044      num_cores: 10
045  inputs:
046    load_case::plane_strain:
047      total_time: 250
048      num_increments: 250
049      target_def_grad_rate: 1.0e-4
050      direction: xz
051    homogenization:
052      SX:
053        mechanical:
054          type: pass
055        N_constituents: 1
056    damask_phases:
057      Ti:
058        lattice: hP
059        c/a: 1.587
060        mechanical:
061          output: [F, F_p, P, O]
062          elastic:
063            type: Hooke
064            C_11: 160.0e9
065            C_12: 90.0e9
066            C_13: 66.0e9
067            C_33: 181.7e9
068            C_44: 46.5e9
069          plastic:
070            type: phenopowerlaw
071            output: [gamma_sl]
072            N_sl: [3, 3, 0, 12]    # basal, prism, -, 1. pyr<c+a>
073            dot_gamma_0_sl: 0.001
074            n_sl: 20
075            a_sl: 100
076            xi_0_sl: [400.e6, 420.e+6, 0.0, 612.e+6]
077            xi_inf_sl: [420.e+6, 440.e+7, 0.0, 622.e+6]
078            h_0_sl-sl: 200.e+6
079            h_sl-sl: [+1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, -1.0, -1.0, -1.0, -1.0,
080              -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, +1.0, 1.0, 1.0, 1.0,
081              1.0, 1.0, 1.0, 1.0, 1.0, 1.0, +1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
082              1.0, 1.0, +1.0, 1.0, 1.0, 1.0, 1.0, 1.0]      # unused entries are indicated b
083      Ti_iso:
084        lattice: cF
085        mechanical:
086          output: [F, P]
087          elastic:
088            type: Hooke
089            C_11: 175.5e9
090            C_12: 90.4e9
091            C_44: 42.55e9   # isotropic value: 0.5*(C_11-C_12)
092          plastic:
093            type: none
094      Air:
095        lattice: cF
096        mechanical:
097          output: [F, P]
```

```
098          elastic:
099            type: Hooke
100            C_11: 1e8
101            C_12: 1e6
102            C_44: 49.5e6   # isotropic value: 0.5*(C_11-C_12)
103          plastic:
104            type: isotropic
105            output: [xi, gamma]
106            xi_0: 0.3e6
107            xi_inf: 0.6e6
108            dot_gamma_0: 0.001
109            n: 5
110            M: 3
111            h_0: 1e6
112            a: 2
113            dilatation: true
114    damask_post_processing:
115    - name: add_stress_Cauchy
116      args: {P: P, F: F}
117      opts: {}
118    - name: add_strain
119      args: {F: F, t: U, m: 0}
120      opts: {}
121    VE_response_data:
122        # Use for average quantities in a single phase, say average stress/strain at each st
123      phase_data:
124      - field_name: sigma
125        phase_name: Ti
126        out_name: vol_avg_stress
127        transforms: [mean_along_axes: 1]
128      - field_name: epsilon_U^0(F)
129        phase_name: Ti
130        out_name: vol_avg_strain
131        transforms: [mean_along_axes: 1]
132        # Use to extract spatial data from (probably a subset of increments)
133
134#step =    1, 2, 3,  4,    5,  6,  7,  8,  9,  10, 11,  12,  13,  14        #use step 5 and
135#A3_zone1 [0, 2, 7,  26,   33, 43, 51, 56, 63, 70, 84,  114, 155, 199]
136#A3_zone2 [1, 5, 10, 30,   36, 44, 50, 56, 67, 84, 105, 139, 187, 243]
137#A3_zone3 [0, 0, 7, 29,    37, 44, 49, 57, 68, 77, 93,  130, 181, 241]
138#B2_zone1 [1, 3, 7, 14,    33, 41, 50, 57, 65, 79, 105, 133, 165, 196]
139      field_data:
140      - field_name: epsilon_U^0(F)
141        increments:
142        - values: [ 37, 44, 49, 57, 68, 77, 93,  130, 181, 241]
143      - field_name: sigma
144        increments:
145        - values: [37, 44, 49, 57, 68, 77, 93,  130, 181, 241]
146      - field_name: O
147        increments:
148        - values: [0]
149      - field_name: grain     # Grain mapping
150        increments:
151        - values: [0]
152      - field_name: phase     # Phase mapping
153        increments:
```

```
      - values: [0]
      # Use to extract grain averaged data
    grain_data:
    - field_name: epsilon_U^0(F)
      increments:
      - values: [ 37, 44, 49, 57, 68, 77, 93,  130, 181, 241]
    - field_name: sigma
      increments:
      - values: [ 37, 44, 49, 57, 68, 77, 93,  130, 181, 241]
    - field_name: gamma_sl
      increments:
      - values: [ 37, 44, 49, 57, 68, 77, 93,  130, 181, 241]
    - field_name: O
      increments:
      - values: [0]
  damask_numerics:
    grid:
      itmin: 2
      itmax: 100
      derivative: FWBW_difference
```

8 Bibliography

References

[1] A. Plowman, *MatFlow*, 2021. [Online]. Available: https://pypi.org/project/matflow/.

[2] F. Roters, M. Diehl, P. Shanthraj, *et al.*, "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*, vol. 158, no. April 2018, pp. 420–478, 2019, ISSN: 09270256. DOI: 10.1016/j.commatsci.2018.04.030.

[3] M. Atkinson, R. Thomas, A. Harte, P. Crowther, and J. Quinta da Fonseca, *DefDAP: Deformation Data Analysis in Python*, 2021. [Online]. Available: https://zenodo.org/record/4697260/export/json#.YO68kehKg2w.

[4] A. Plowman, *Formable*, Manchester, 2021. [Online]. Available: https://pypi.org/project/formable/.

[5] R. Hielscher, R. Kilian, D. Mainprice, F. Bachmann, and F. Bartel, *MTEX*, Chemnitz, 2021. [Online]. Available: https://mtex-toolbox.github.io/index.html.

[6] A. Nicolas, A. W. Mello, Y. Sun, D. R. Johnson, and M. D. Sangid, "Reconstruction methods and analysis of subsurface uncertainty for anisotropic microstructures," *Materials Science and Engineering: A*, vol. 760, pp. 76–87, Jul. 2019, ISSN: 09215093. DOI: 10.1016/j.msea.2019.05.089.

[7] T. Maiti and P. Eisenlohr, "Fourier-based spectral method solution to finite strain crystal plasticity with free surfaces," *Scripta Materialia*, vol. 145, pp. 37–40, 2018, ISSN: 13596462. DOI: 10.1016/j.scriptamat.2017.09.047. [Online]. Available: https://doi.org/10.1016/j.scriptamat.2017.09.047.

[8] R. Perkins, "File formats on the Internet," *Computers and Geosciences*, vol. 21, no. 6, pp. 775–777, 1995, ISSN: 00983004. DOI: 10.1016/0098-3004(95)00008-V.

[9] *ParaView*. [Online]. Available: https://www.paraview.org/overview/.

[10] F. Bridier, D. L. McDowell, P. Villechaise, and J. Mendez, "Crystal plasticity modeling of slip activity in Ti-6Al-4V under high cycle fatigue loading," *International Journal of Plasticity*, vol. 25, no. 6, pp. 1066–1082, 2009, ISSN: 07496419. DOI: 10.1016/j.ijplas.2008.08.004. [Online]. Available: http://dx.doi.org/10.1016/j.ijplas.2008.08.004.

[11] *DAMASK documentation 1.1.3. Hexagonal (hex).* [Online]. Available: https://damask.mpie.de/Documentation/Hex.

[12] M. Kasemer, R. Quey, and P. Dawson, "The influence of mechanical constraints introduced by β annealed microstructures on the yield strength and ductility of Ti-6Al-4V," *Journal of the Mechanics and Physics of Solids*, vol. 103, pp. 179–198, 2017, ISSN: 00225096. DOI: 10.1016/j.jmps.2017.03.013.

[13] P. Dawson, D. E. Boyce, J. S. Park, E. Wielewski, and M. P. Miller, "Determining the strengths of HCP slip systems using harmonic analyses of lattice strain distributions," *Acta Materialia*, vol. 144, pp. 92–106, 2018, ISSN: 13596454. DOI: 10.1016/j.actamat.2017.10.032.

[14] P. R. Dawson, D. E. Boyce, J. S. Park, E. Wielewski, and M. P. Miller, "Determining the strengths of HCP slip systems using harmonic analyses of lattice strain distributions," *Acta Materialia*, vol. 144, pp. 92–106, Feb. 2018, ISSN: 13596454. DOI: 10.1016/j.actamat.2017.10.032.