Robust methodology for the EBSD local misorientation analysis of surface cold work

Ivan Bogachev[a, *], Kevin M. Knowles[a] and Grant J. Gibson[b]

[a] University of Cambridge, Department of Materials Science and Metallurgy, 27 Charles Babbage Road, Cambridge, CB3 0FS, UK

[b] Rolls-Royce plc, PO Box 31, Derby, DE24 8BJ, UK

Abstract

The amount of cold work induced by a surface hardening technique and the depth to which it is produced within a metallic material are both important parameters within the field of surface engineering. In this paper a methodology of establishing reliable estimates of the depth and magnitude of cold work in surface hardened nickel-based superalloy single crystals from a dataset (map) of electron backscattered diffraction images through the analysis of local misorientations is described in detail. The impact of varying a number of acquisition parameters within the scanning electron microscope and the impact of the various post-acquisition analysis parameters on the outcome of the analysis are both described and discussed in detail. The Python script used to perform this analysis is published in full. The principles and processes underlying this methodology, as well as the published script, can be readily adapted for the analysis of datasets of electron backscattered diffraction images from other surface hardening techniques and other surface-hardened materials.

1. Introduction

Electron backscattered diffraction (EBSD) is a highly versatile scanning electron microscopy (SEM) technique, with many applications in the field of materials science [1]. During an EBSD scan, which is typically performed with a sample tilt of ~70°, characteristic Kikuchi [2], or electron backscattered, diffraction patterns (EBSPs) are recorded at each step of the scan. The patterns are then automatically indexed to determine the orientation of the sample material crystal matrix at each beam position

relative to a set of external samples axes [1,3,4,5]. Resolutions on a nanometre scale are achievable due to the relatively small interaction volume of the electron beam with the sample.

One use of EBSD which has been gaining increased prominence in recent years is in the study of cold work in metallic materials [6–9]. Cold work is the collective term for the microstructural changes induced in a sample by plastic deformation at sufficiently low temperatures to prevent the "recovery" of the material. A key component of cold work is an increased density of dislocations in the cold worked area. Some of these dislocations, so-called "geometrically necessary dislocations" (GNDs) form networks with a net non-zero Burgers vector, rotating the material locally to accommodate plastic strain resulting from the deformation [10]. Crystal matrices in neighbouring parts of a cold worked area will therefore be misorientated relative to each other. These local misorientations (LMs) can be quantified by EBSD, with greater LM values indicating greater densities of GNDs, and therefore greater "cold work".

Cold working sees extensive use in various engineering applications as a way of increasing the hardness of a metallic material. In particular, surface hardening techniques are frequently employed to increase the fatigue life of components in a variety of applications [7,11–13]. These techniques function by deforming the component surface. For example, in mechanical shot peening (MSP), the surface of a sample is bombarded by small hard shot. This process generates a surface layer of high dislocation density which hardens the material, thereby helping to inhibit the propagation of surface cracks during fatigue loading [14–16].

The density of dislocations and the depth to which an elevated density of dislocations extends beneath the surface are key parameters which determine the amount and extent of the strengthening effect provided by surface hardening. The lateral distribution of the cold worked layer across a sample surface is also important, as any inhomogeneities in the cold worked layer may lead to regions of weakness: underworked areas would see less protection, while overworked areas may act as crack nucleation sites. The ability of LM analysis in EBSD to create maps of cold work across a sample surface up to a nanometre scale therefore makes it an ideal technique for characterising surface cold work, allowing the observation of local features which are indistinguishable in other commonly used techniques for characterising cold work, such as hardness testing and X-ray diffraction [17,18].

A methodology for LM analysis of cold work, which has recently been used extensively by the present authors in several studies on surface hardened samples of a single crystal nickel alloy, is described in this paper, including a Python script written to implement this methodology. While the methodology is not in itself entirely novel, to the best of the authors' knowledge, its detailed description here is one of the few currently available in open access in which every stage of the analysis is described in a unified and continuous form. In particular, the Python script, which is given in full in the Appendix, allows the user greater flexibility and control over the output than is offered by commercially available software, such as the Oxford Instruments AZtec package.

Details of the experimental methods used to gather the EBSD data are given in Section 2. The various stages of the LM analysis of the EBSD data are described in Section 3, including the derivation of the two key numerical cold work quantities: depth and magnitude. The sensitivity of the method to changes in various acquisition and analysis parameters is considered in Section 4. A brief summary of the outcome of this work is given in Section 5.

2. Experimental methods

2.1 Sample preparation

The samples used in the present study were manufactured by Rolls-Royce plc from the nickel-based single crystal superalloy, CMSX-4. Nickel-based superalloys are a diverse group of alloys, used in both single crystal and polycrystalline form, characterised by high strength almost up to their melting points, excellent oxidation resistance, excellent corrosion resistance and excellent creep resistance [19,20]. Single crystal superalloys are commonly employed in the production of turbine blades for gas turbines. Surface hardening, including MSP, is increasingly used to improve the fatigue resistance of such components. It is of interest to industry to assess the depth, intensity and uniformity of the resultant surface cold work layers.

A number of samples were conventionally cast as single crystals, machined by low stress grinding into bar shapes, then briefly heat treated at 1100 °C for 1 h in an Ar atmosphere to alleviate the surface effects of the machining process. MSP was then performed by Curtiss-Wright Surface Technologies. Different MSP parameters were used for each sample to create a range of cold work depths and densities. A full description of the MSP parameters used to peen these samples and a full description of their as-peened states may be found in previous publications by the authors [21,22].

Metallographic specimens were prepared by cutting cross-sections perpendicular to the cold worked surfaces of the samples, using either a Struers Accutom-5 or a Secutom-10 precision cutting machine, with a low maximum feed rate of 0.025 mm/s to minimise machining damage. The cross-sections were mounted in conductive resin, ground with progressively increasing grades of grinding paper up to grit size 2500, then polished using 6, 3, 1 and 0.25 µm diamond suspensions, finishing with a ~15–20 min 0.06 µm colloidal silica polish. Following polishing, the samples were sonicated in soapy water, then in acetone and finally in ethanol, after which they were dried in a hot cupboard at ~70 °C, generally overnight.

2.2. Data acquisition

EBSD data was acquired on a Zeiss GeminiSEM 300, equipped with a field-emission gun and an Oxford Instruments Symmetry detector. A voltage of 20 kV, a working distance of ~20 mm, an aperture of

120 µm and a sample tilt of 70° were used as the standard microscope settings. The effects of using different microscope settings on the results of the LM analysis of EBSD data are discussed in Section 4.2.

The spatial resolution of the EBSD scans, also known as the scan step size, was set equal to 0.5 µm/pixel. This step size is a compromise between the ability to resolve fine details of the cold work structure and the total scan time. The step size directly affects the results of the LM analysis. This is because, for a smaller step size, the nearest neighbour misorientations are assessed over smaller distances. Since misorientations are likely to be lower the less distance there is between pixels [10], local misorientation would on average be lower for smaller step sizes. Results of LM analysis should therefore only be compared between scans of equal step size. The scan step size is an input parameter in the script and should be set by a user in code line 327, as given in the Appendix.

The scan area should include a large section of non-cold worked material to measure accurately the cold work depth, which is defined as the depth to which the LM is elevated in comparison with the non-cold worked far-field material. For the MSP-treated samples, a scan depth of ~300 µm was used in this study. The scan depth should be adjusted for the surface hardening techniques under consideration. For example, deep cold rolling typically causes deeper layers of cold work and so therefore requires a greater scan depth to be used [23].

The Oxford Instruments AZtec 4.1 acquisition software was used to record and index the EBSD patterns automatically, and then to derive the orientation of the material in each pixel of the scan relative to a fixed external set of sample axes. Precise indexing is required to obtain the best results in the subsequent LM analysis. In the AZtec 4.1 acquisition software, the goodness of fit of the automatic indexing can be estimated using the mean angular deviation parameter (MAD). This parameter is the average difference of the measured angles between Kikuchi bands in the recorded EBSP and the predicted angles between these bands in a simulated EBSP based on the proposed best-fit crystal lattice orientation. Lower MAD values signify that the proposed material orientation is closer to the experimentally measured orientation, and therefore indicate a more precise indexing and/or better quality EBSP. In general, scans with MADs of ≤ 0.5° in the far-field (non-cold worked) portion of the scan were found to yield better results in LM analysis. The best quality of indexing, with the lowest MADs, was found to be achievable in the AZtec 4.1 acquisition software using the “Refined Accuracy” indexing mode with the “Refine Solutions” option enabled at the Optimise Solver stage.

After acquiring the orientational data, LM analysis was then performed on this data using a self-written script in the Python programming language. The functioning of this script is described in the next section, with reference to specific code lines in the script, which, as previously mentioned, is given in full in the Appendix.

3. Local misorientation analysis

3.1. Calculating misorientation between two pixels

When produced as an output from the AZtec 4.1 acquisition software, the orientation data is given in the form of the Bunge Euler angles, (φ_1, Φ, φ_2) [24,25], for each pixel in the scan. To find misorientations between the orientations of two given pixels defined in this way, the Euler angles are first converted into unit quaternion vectors, as described by Morawiec and Pospiech [26].

In the script in the Appendix, the quaternion for a given set of Euler angles is calculated in code lines 8–15. A .txt file containing the Euler angles (φ_1, Φ, φ_2) for each pixel in an EBSD dataset is converted by the script into an array of quaternions in code lines 55–79. Any entry with Euler angles exactly equal to (0, 0, 0) is counted as unindexed, because this is how unindexed pixels are recorded by the Aztec 4.1 acquisition software. The corresponding quaternion is set in the script to consist of NaN ('Not a Number') values. NaN is a value assigned in computing to variables which, while taking part in the computational algorithm, do not have a definite numerical value, and therefore will not contribute to any subsequent calculation. It is important to note that this function is written specifically for the output produced by the AZtec 4.1 acquisition software, and must be modified if the output of a different acquisition software is to be analysed.

Once quaternion vectors have been calculated, the misorientation angle between two pixels is calculated as twice the arccosine of the dot product of the corresponding normalised quaternion vectors [27]. It is important to account for the possibility that the acquisition software may index the orientation of different pixels with distinct, but symmetrically equivalent, indices. A simple example of this is taking a [010] direction to lie parallel to the face of the sample instead of a [100] direction; these two directions are equivalent in a cubic material. Such indexing leads to artificially high misorientations between pixels, and a number of such artifacts were found in the raw output data produced by the AZtec 4.1 acquisition software.

In the script, if a misorientation greater than 62.8° (the greatest physical rotation between two cubic unit cells [28]) is found, it is corrected by premultiplying one of the quaternions involved, in turn, by quaternions corresponding to one of the symmetry operations of the holosymmetric cubic point group, $m\bar{3}m$ [29], and recalculating the misorientation. The smallest misorientation thus obtained (sometimes known as the "disorientation") is taken to be the true misorientation between the two pixels. The function to calculate misorientations between two given pixels, accounting for the possibility of symmetrically equivalent orientations, is given in code lines 17–53.

Misorientations between individual adjacent pixels can be used to identify the presence of grain boundaries when the misorientation angle between neighbouring pixels is greater than a threshold value, termed the grain misorientation angle. However, in order to use local misorientation as a

measure of cold work, a LM parameter quantifying the magnitude of the misorientation in a given area must be defined.

3.2. Calculating and mapping local misorientation

3.2.1 Defining the kernel average neighbour misorientation (KANM) parameter

There are several frequently used LM parameters, all of which provide a measure, in units of degrees (°), of the level of misorientation present in a particular area. Higher values of LM parameters indicate a higher density of geometrically necessary dislocations, and therefore higher cold work, in that area. A distinction can be made between grain-based and kernel-based methods [10]. In grain-based methods, a single LM value is calculated for an entire grain, in order to allow comparison of the deformation levels of different grains. However, the spatial resolution of grain-based methods is limited to the size of the grain, and, by definition, grain-based methods cannot be used for single crystal samples.

In kernel-based methods the average misorientation is calculated for a $k \times k$ subset of pixels – the kernel – around a given pixel in the centre (Figure 1). k is the kernel size, and is equal to $(2m + 1)$, where m is an integer ≥ 1, with smaller kernel sizes giving greater spatial resolution.

A number of kernel-based LM parameters can be defined. In particular, the kernel average neighbour misorientation (KANM) was chosen for the present methodology. KANM is defined as the average of the misorientations between each pair of neighbouring pixels within the kernel, as illustrated graphically in Figure 1. The choice of this particular LM parameter was based on its previous use in the literature on LM analysis of superalloys [30, 31]. The use of other similar kernel-based LM parameters is possible, but it is known that the actual choice of parameters has only a limited effect on the outcome of LM analysis [10].

3.2.2. Calculating and mapping KANM

For a dataset of quaternions, a kernel is defined in turn around every pixel, and a KANM value is calculated for each one (ignoring any unindexed pixels) (code lines 81–156). The smallest possible kernel size of 3 was used to obtain the most localised misorientation measurements. A filtering operation is performed, setting the value of any pixel with less than three indexed nearest neighbours to NaN, in order to remove stray pixels from the sample surface.

One important note to make is that as the scan step size approaches the scale of tens of nanometres, cold work information will start to be lost, and misorientations between neighbouring pixels will approach the precision of the EBSD orientation measurement. If such a fine step size is desired, it is advised to use a larger kernel size, with the LM parameter modified to only take into account

misorientations between a central pixel and pixels at the perimeter of the kernel. Such a LM parameter is known as the kernel average central misorientation (KACM) [10]. For a kernel size of 3, KANM and KACM are identical.

Defects such as pores or recrystallised grain boundaries were excluded from the calculation by setting a maximum misorientation angle, the kernel exclusion angle (ω_k), with any misorientations above this ignored. This kernel exclusion angle was set to 2°, a choice that was informed by the literature on the LM analysis of other nickel-based superalloys. Values of 1, 2, 3 and 4° [6,9,30,32] have all been reported, of which 2° has been used most frequently. In the present study this value was found to afford an optimal middle ground between removing the effects of any material defects, such as pores, and the sampling of real cold work-induced misorientation. Different values of k and ω_k can be set by the user (code lines 328 and 329). The effects of other values of these parameters on the analysis of the present sample are discussed in Section 4.

After an array of KANM values is obtained, it can be plotted (function given in code lines 158–191) to create a map of the cold work within the given area of a sample. Examples of such maps are shown in Figures 1b and 2a. The colour of a pixel on such a map indicates the local misorientation according to a given colour scale. Pixels with colours corresponding to larger numbers can therefore be seen to have higher levels of cold work. These KANM maps are powerful tools for revealing features of the cold worked layer, such as areas of concentrated deformation, underworked regions, slip bands or recrystallised grains [22]. The range of the colour scale can be changed (this is set using the parameter "yrange" in code line 335) to change the contrast of the image in order to reveal features with different extents of cold work. A colour scale range of 0–2° was most commonly used by the authors, to match ω_k = 2°.

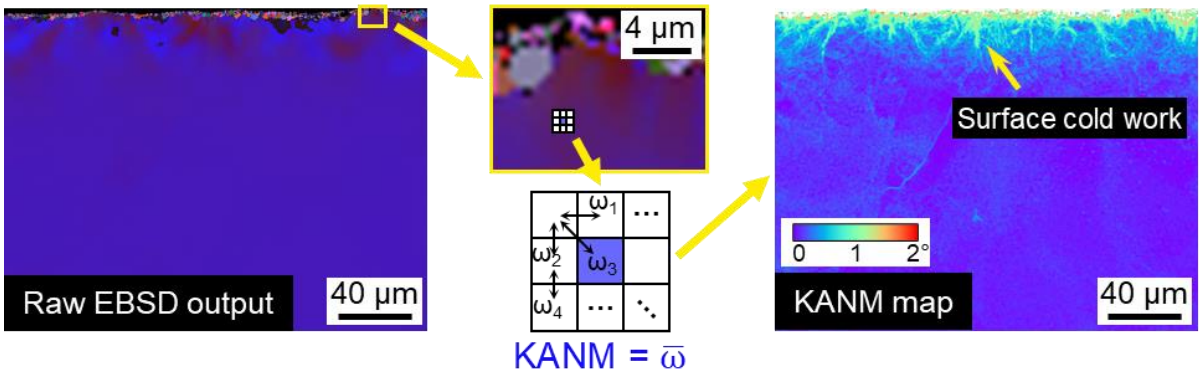

***Figure 1**: A visual representation of the process of obtaining KANM maps from raw EBSD orientational data. For each pixel in the raw EBSD data (left), a square grid of k × k pixels is taken (the kernel) and the angles between each pair of neighbouring pixels (ω_i) found. The KANM value is the average of all of these angles. Plotting KANM values produces a map of cold work, with higher KANM values indicated by the yellow-red end of the colour scale.*

3.3. Quantifying sample-wide cold work: depth and magnitude

In order to characterise a cold worked layer in a particular sample and to compare it with cold worked layers produced with different surface hardening conditions, two quantitative measures were defined in the course of developing this methodology: cold work depth and cold work magnitude. To derive these two measures, another key quantity has to be defined: the baseline misorientation, which is discussed in detail below.

3.3.1. Baseline misorientation and cold work depth

Conceptually, the cold work depth can be defined as the depth from the surface up to which the level of cold work, as measured by the local misorientation, is higher than some far-field baseline level of misorientation. This baseline represents the typical misorientation within the non-cold worked material in the sample, in units of degrees (°). Ideally, it would represent the fundamental variability of orientation of the crystal matrix, which is present even in undeformed single crystals. In practice, noise coming from factors limiting the precision of the measurement of the material orientation by EBSD, such as the resolution of the EBSP and the accuracy of the indexing, will tend to raise this baseline further. A number of other practical points also have to be considered when applying this methodology.

To obtain the cold work depth from a KANM map, a depth profile of the map has to first be created. The construction of the depth profile and the computation of the cold work depth is accomplished by the script in the function given in the code lines 193–245, while the calculated cold work depth value is printed out and the depth profile is plotted graphically using the function in code lines 247–296. To create the depth profile, the mean and the standard deviation of KANM values in each scan line along the depth axis are calculated (code lines 220–221). Plotting these graphically allows the user to see how the average cold work changes with depth (Figure 2). The position of zero depth is defined as the point where at least half of the pixels in a scan line have non-NaN KANM values. Note that it is here assumed that the cold worked surface is at the top of the input KANM map, with the in-depth axis taken as lying from the top of the map downwards.

Next, the baseline misorientation value must be calculated. By analogy with a prior methodology proposed by Child *et al*. for LM analysis using a grain-based LM parameter [6], the baseline misorientation was defined as the average of the (mean + 1 standard deviation) values of a far-field segment of the profile. The length of this segment is taken as 1/6 of the depth of the map (about 50 µm in a map of depth 300 µm). This can also be customised in the script by setting an alternate value of the function variable “base” (in units of µm) in code line 341.

When developing the initial methodology, the original intention was to take the baseline segment from the last part of the depth profile, furthest from the cold worked surface. However, it was found that many KANM maps tend to contain far-field noise artefacts. An example of this phenomenon is

shown in Figure 2. This noise led to a rise in the mean scan line KANM values at the end of the profile, raising the level of the baseline misorientation locally as a consequence (Figure 2b).

Instead, therefore, it was decided to use the 1/6 smallest scan line (mean + 1 standard deviation) values in the profile as the baseline segment, thereby mitigating the effects of any far-field noise on the baseline. Baselines calculated using the smallest 1/6 values are, by definition, less than or equal to baselines calculated using the end of the profile for the baseline segment; this has the general effect of increasing the measured cold work depths (Figure 2c). The effect of changing the length of the baseline segment, and the relationship of the baseline with the measured cold work depth, are discussed further in Section 4.

Locations where the mean of a scan line becomes equal to the baseline value may not be unique in a particular scan, since the depth profile can undulate, with local fluctuations of the scan line means above and below the baseline. High KANM values associated with far-field noise can also produce local perturbations of the scan line mean values about the baseline. It might therefore become necessary to choose one particular depth value to be defined as the actual cold work depth from a number of potential candidate positions. All the potential depth values are identified and recorded by the script (code lines 230–237). If there is more than one of these then the median, or, if there is an even number of such positions, the value immediately to the right of the median of the candidate list, is defined by the script as the cold work depth value (code lines 238–241). Any depth values associated with noise will skew this median position, and if any such values occur, the user will have to choose the correct depth position manually. To help identify these erroneous depth values and, if necessary, to choose the correct depth position manually, the script enables all the candidate depth positions to be displayed when plotting depth profiles (code lines 284–286), as is demonstrated in Figure 2c. The user can then pick out the correct value of the depth and define it as the output cold work depth by setting the parameter “pos” to the appropriate value (code line 339).

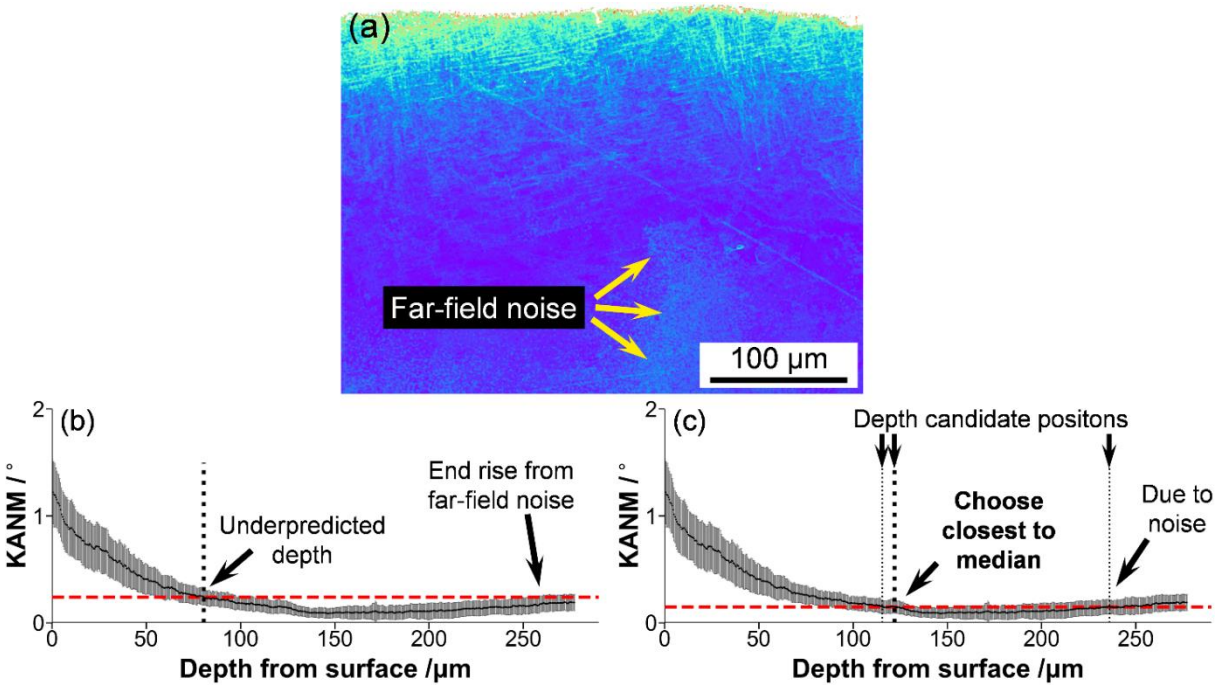

***Figure 2**: (a) Example KANM map of a sample surface hardened using MSP, which displays some far-field noise, and (b, c) the depth profile resulting from this KANM map. Cold work depth positions are indicated by vertical dotted lines. In (b) the baseline misorientation is calculated using the last 50 µm of the depth profile, and is therefore affected by an end rise in mean KANM values, which is caused by the far-field noise seen in (a). In (c), the lowest (mean + standard deviation) values, equivalent to 50 µm in cumulative length, are used to calculate the baseline, thereby removing the effects of far-field noise.*

Baseline values in (b) and (c), represented by dashed red lines, are 0.24 and 0.15°, respectively. In (c) there are three positions where the scan line means cross the baseline, all of which are shown on the profile by vertical lines. The furthest value on the right, significantly removed from the other two, clearly arises due to the far-field noise. Amongst the other two candidate values, the second was picked according to the rule that the median, or, the closest value to the right of it, is taken as the cold work depth, after positions caused by far-field noise are excluded. Cold work depths are 80.5 and 122 µm in (b) and (c), respectively, the greater value in (c) being a direct consequence of the lower baseline.

3.3.2. Cold work magnitude

"Cold work magnitude" is a term which was coined during the development of the present methodology to describe the intensity, or density, of the cold worked layer. As a comparative measure, cold work magnitude can be evaluated qualitatively by eye from a side-by-side comparison of the KANM maps of two or more samples. The sample with higher KANM values at a given depth can be

concluded to have received harsher cold working than a sample with lower KANM values in the same area.

To provide a quantitative reflection of these comparisons, a numerical value for the magnitude was defined, proportional to the average increase, per depth, of KANM values within the cold worked layer above the baseline in a particular EBSD scan. This value is found, in units of °/µm, by summing all the (mean – baseline) KANM values within the cold worked layer, and dividing this by the cold work depth:

$$CW\ magnitude = \frac{\sum_{pixels\ in\ CW\ layer} mean\ KANM\ in\ given\ scan\ line - baseline\ value}{CW\ depth} \quad (1)$$

Magnitude values are calculated by the script (code lines 298–319) for each of the candidate depth positions, as defined in Section 3.3.1, and the magnitude value corresponding to the depth chosen as the defined cold work depth should then be taken as the best estimate or 'correct' magnitude of the cold work.

These numerical magnitudes should only be compared between scans with the same step size. The above definition of the magnitude makes its value directly proportional to the step size of the scan, since the more pixels there are per given length of the sample, the more (mean – baseline) values contribute to the sum. An average increase in KANM per pixel in the cold worked layer could be obtained by multiplying the magnitude as defined previously by the scan step size. However, this would still not make the magnitude independent of the step size, because of the dependence of LM values on the by each pixel in the scan (see Section 2.2). Because of this intrinsic dependency on scan step size, it was decided to leave cold work magnitudes in the logically appealing units of °/µm, and use them as relative quantities, only to be compared between scans with equal step size.

It should also be noted that numerical magnitudes were found to favour samples with shallower cold work. This is because cold work in surface hardened samples is invariably by far the highest at the hardened surface, especially as the surface of a sample tends to contain defects which further increase KANM values in this region. The shallower the cold worked layer, the greater a proportion of it is made up by this near-surface high-deformation region, and therefore the numerical cold work magnitude is greater. Despite all the limitations of this quantity, numerical cold work magnitude is a useful guide when quantifying trends in the density of a cold worked layer produced by altering surface hardening conditions [33].

Recognition of these various aspects of the analysis and how they are treated concludes the description of the methodology developed for the LM analysis of surface cold worked samples. The next section is a discussion of the sensitivity of this methodology to changes in various parameters, including both data collection conditions and post-processing variables. In particular, the relationship of the baseline misorientation with cold work depth and magnitude is discussed.

4. Methodology sensitivity

This section is a demonstration of how KANM maps and the cold work depths and magnitudes derived from them are affected by variations in the acquisition conditions, the quality of the resulting EBSD scan, and the post-acquisition analysis parameters. The dependence of cold work depth and magnitude on the map baseline misorientation is discussed in Section 4.1. The effects of varying acquisition parameters on the LM analysis is shown in Section 4.2, the effects of changing indexing mode is shown in Section 4.3, and finally the effects of post-acquisition parameters are discussed in Section 4.4.

4.1. Effect of baseline on cold work depth and magnitude

Variation of the baseline misorientation value due to various factors occurring at different steps of the methodology is of primary importance. As has already been seen in the previous section, this is because the baseline directly affects the estimate of the cold work depth. It also impacts, though less strongly, the cold work magnitude.

The variation of cold work depth and magnitude in a number of KANM maps of four different MSP samples is shown in Figure 3. Each set of data was obtained from the same sample for a given line, all using the same scanning electron microscope, similar acquisition conditions and the same LM analysis parameters.

It can be seen from this figure that higher baseline misorientation values tend in most cases to lead to lower cold work depths. This is because higher baselines set higher thresholds for a misorientation value to be considered “elevated”, and therefore to constitute part of the cold worked layer. The effect on cold work magnitude is less significant and systematic. This is because the cold work magnitude, as defined in Equation 1, depends in opposite senses on both cold work depth and the baseline value, so that the relative strengths of their variations determine whether cold work magnitude is higher or lower in a particular map.

This demonstrates a crucial point: estimates of cold work depths measured using the present LM analysis methodology in any two samples are only comparable when the baselines of the KANM maps are similar. Such a comparison should first of all be achieved by using the same LM analysis parameters in each map to be compared. On the other hand, some process inputs can affect the orientational resolution achieved in the far-field region of an EBSD dataset, and this may lead to fluctuations of the baseline value between different maps. Such conditions include the microscope setup, the resolution of the EBSD detector, the precision of the EBSP indexing and any microscope drift during acquisition. Any factor leading to less precise indexing of the EBSPs will cause random errors in calculating local misorientations to increase, consequently causing higher KANM baselines.

To work around the issue of far-field fluctuations impacting the baseline value in individual maps, even when using the same microscope setup, the present authors typically recorded sets of several KANM maps for each sample being studied, and used an average depth obtained across these individual maps. Such average depths were then considered comparable across a number of samples if the average baselines of each sample set were similar to each other.

In subsequent sections, the effects of several process parameters are presented.

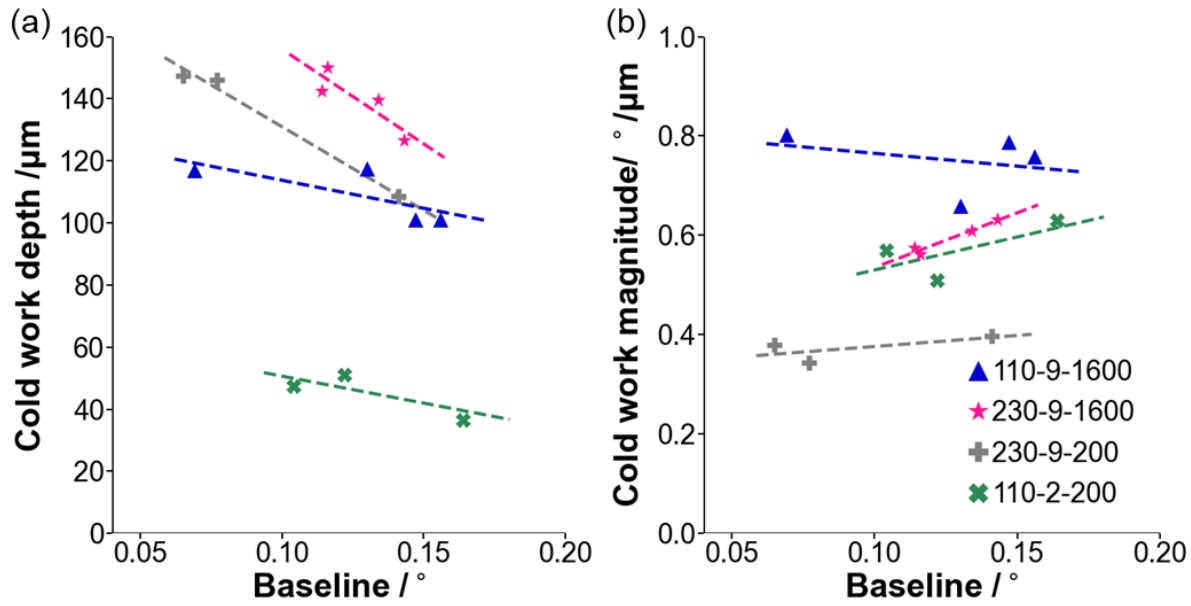

***Figure 3**: Sets of (a) cold work depths, (b) cold work magnitudes, plotted against the baseline KANM values in the corresponding KANM maps for various samples. Cold work depths vary within a range of ±20 % when calculated from maps with different baselines, with lower baselines usually leading to higher depth values. Average cold work depths derived from these sets are comparable to each other because the average baselines of these ranges are similar. Cold work magnitudes vary less than cold work depths, within a range of ±10 %, and do not have a strict pattern of increasing or decreasing with baseline.*

4.2. Acquisition parameters

The sensitivity of the methodology to three key acquisition parameters: sample working distance, EBSD detector position and EBSP exposure time, were considered. The effects of varying these parameters on baselines in KANM maps of three different samples are shown in Figure 4. While

variations in baseline between the different scans are seen in each graph, none of the acquisition parameters are seen to have a significant systematic effect on the baseline.

A wide range of microscope working distances was used, but the baseline variation does not appear to follow any clearly identifiable pattern (Figure 4a). Both the detector position (Figure 4b) and the exposure time (Figure 4c) display cut-off values of –8 mm and 2 ms, respectively, below which the baseline value increases noticeably, but otherwise the baseline variation appeared random with changes in these parameters. It is therefore evident that the present methodology is robust to small variations in acquisition parameters in a particular microscope setup.

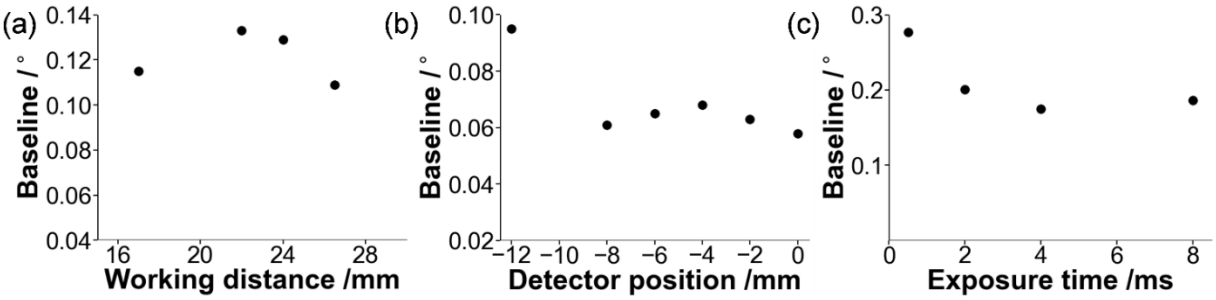

Figure 4: *Baseline KANM values plotted against (a) working distance, (b) detector position and (c) exposure time. Each series of data was acquired from a different sample. Within each series, EBSD scans were acquired from the same area in a particular sample. In (a) the baseline neither increases nor decreases systematically as a function of the working distance. In (b) and (c), values of detector position and exposure time below –8 mm 2 ms, respectively, lead to noticeably higher baseline values, but otherwise baselines vary randomly. The standard values used for each parameter throughout the work were 20 mm, –5 mm and 4 ms, respectively.*

4.3 Precision of indexing

As was mentioned in Section 2.2, it is important to ensure the most accurate indexing setting is used in the acquisition software. In the Oxford Instruments AZtec 4.1 acquisition software used in this study, the use of the “Refined Accuracy” indexing mode, with the “Refine Solutions” option enabled is recommended by the present authors for EBSD analysis. A comparison of KANM maps derived from EBSD data acquired without these options enabled, and with these options enabled, is shown in Figure 5. Without these options being enabled, the baseline is 0.14° and fine microstructural features are partially obscured by random noise (Figure 5a). With “Refined Accuracy” and “Refine Solution” both enabled, the far-field section of the map contains less noise, the effect of which is to produce both a clearer definition of fine microstructural features and a lower baseline of 0.06° (Figure 5b). It is

therefore advisable, whenever possible, to have these, or equivalent, options enabled when performing LM analysis.

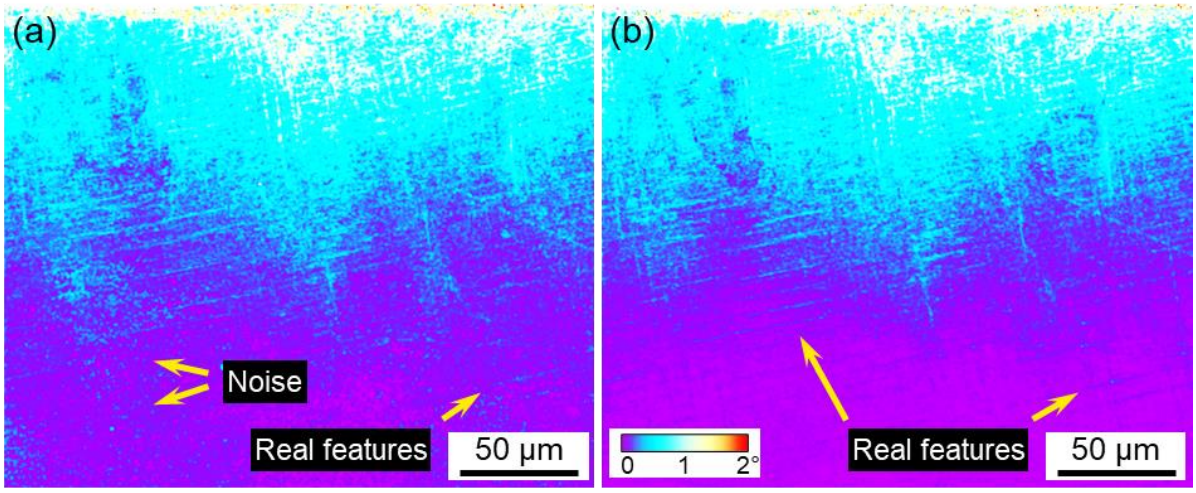

***Figure 5**: KANM maps from the same area of a sample, with (a) "Refined Accuracy" and "Refine Solution" options NOT enabled during acquisition, or (b) enabled. The far-field region of (b) is much more free of random noise, leading to a lower baseline and better resolution of microstructural features. Contrast has been increased in these maps to show the faint in-depth cold work features.*

4.4. LM analysis parameters

The main input analysis parameters, as described in Section 3, are the kernel size, the kernel exclusion angle, and the length of the baseline segment. Throughout this methodology, standard values of these parameters to be used were chosen, the choices shaped by logical reasoning and informed by previous studies using LM analysis. However, these "standard" values are by no means unique, and others can conceivably be used instead. The effects of varying these parameters on KANM maps, cold work depths and magnitudes are demonstrated below.

4.4.1. Kernel size

The smallest kernel size possible, k = 3, was used in order to obtain KANM maps at the finest scale. Kernel sizes can only be odd numbers in order to obtain a square grid centred on a single pixel. KANM maps produced from a single EBSD scan with different kernel sizes, from 3 to 11, are compared in Figure 6. Larger kernel sizes are seen to produce more blurred KANM maps, in which finer microstructural features, such as slip bands, become progressively more difficult to distinguish. This is because a larger kernel averages misorientation over a larger area, leading to a loss of local resolution.

The variations of the baseline, cold work depth and magnitude between the maps in Figure 6 with kernel size are plotted in Figures 7a, b and c, respectively. The baseline values are seen to fall as the kernel size increases. This occurs because the baseline is defined as the "average of the (mean + 1 standard deviation) values of a far-field segment of the profile" (Section 3.3.1). As the size of the kernel increases, kernels of neighbouring pixels will overlap to a greater and greater extent. Since the KANM parameter only considers nearest neighbour misorientations, this will lead to KANM values of neighbouring pixels becoming more and more similar, decreasing the standard deviation within a scan line, and therefore decreasing the baseline.

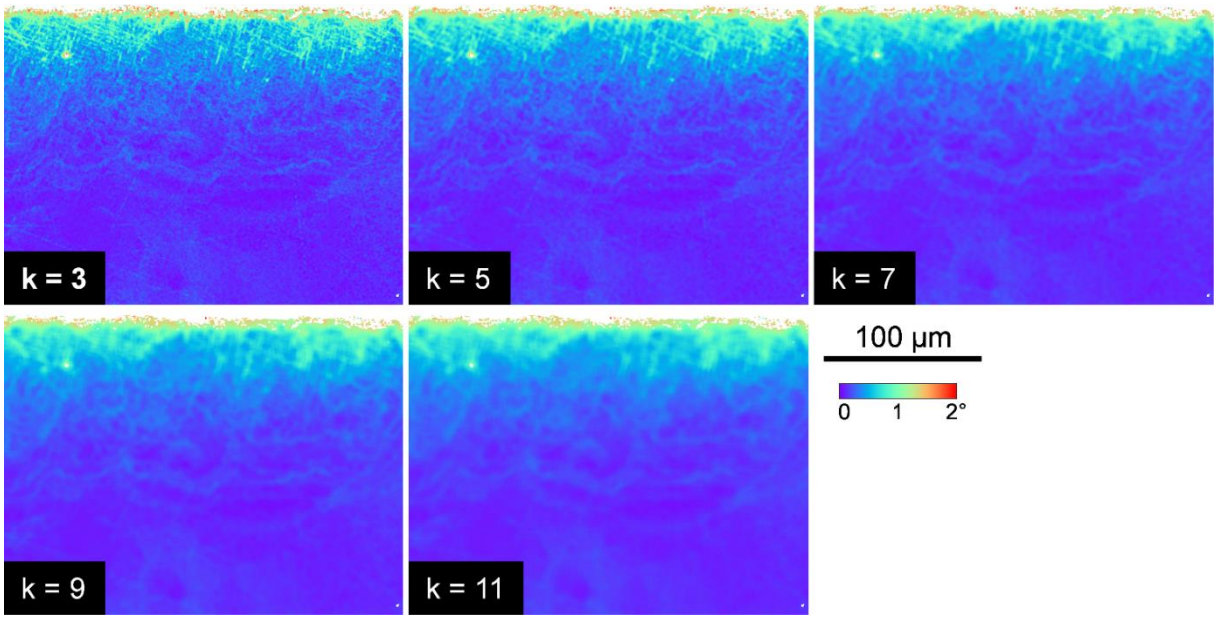

Figure 6: *KANM maps from a single EBSD scan, obtained using progressively larger values of kernel size, k. As the kernel size increases, the maps become more blurred, because with each larger kernel size averaging is occurring over more nearest neighbour misorientations.*

As the baseline becomes smaller, the position of cold work depth, which depends only on the baseline and the scan line means, shifts deeper. Therefore greater depths of cold work are recorded for larger kernel sizes, with an increase of approximately 7 % seen in Figure 7b between k = 3 and 11. Cold work magnitude, affected by both the baseline and the depth of cold work, stays fairly constant varying only within a range of ~2 % in Figure 7c. Overall, the kernel size has a fairly minor effect on the outcome of the KANM analysis, the greatest difference being a blurring of the KANM maps. For this reason alone, the smallest kernel size is preferable.

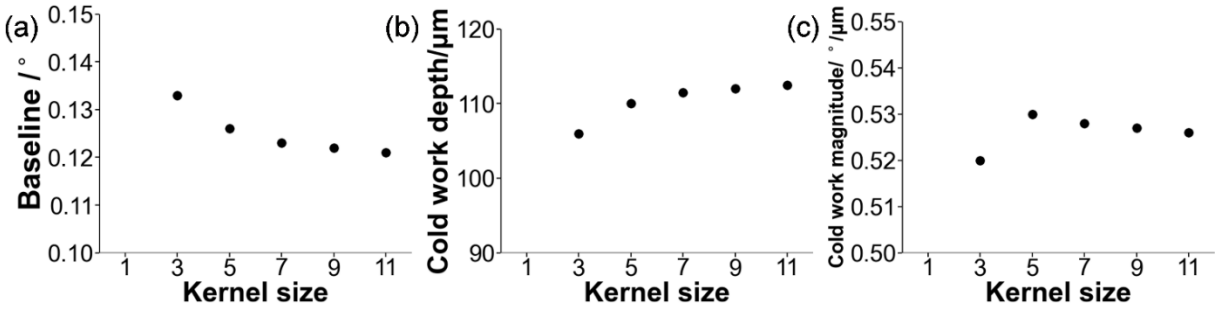

***Figure 7**: (a) baseline KANM, (b) cold work depth and (c) cold work magnitude, obtained from the KANM maps in Figure 6, versus kernel size. The baseline decreases as the kernel size is increased, which leads to greater depth estimates. Cold work magnitude displays an overall increase, but only of ~1–2 %.*

4.4.2. Kernel exclusion angle

A kernel exclusion angle ω_k of 2° was used throughout this methodology. This seemed the most appropriate value for the samples studied, based on existing literature on superalloys, as has been discussed in Section 3.2.2.

KANM maps obtained from a single EBSD scan using various values of ω_k are shown in Figure 8. Corresponding depth profiles are given in Figure 9. In comparison with ω_k = 2°, it is clear that with the kernel exclusion angle set to 1°, much of the cold work structure becomes less distinct, and the scan line mean KANM values decrease significantly up to depths of ~35 µm from the surface. This indicates that a significant amount of the cold work in the cold worked layer corresponds to misorientations of 1–2°.

Increasing the exclusion angle above 2° primarily affects the top ~5 µm of material. Indeed, changing ω_k between 5 and 63° only makes a difference to the distribution of KANM values in this area. The topmost part of a sample is prone to contain microstructural damage such as scratches, pores, recrystallised grains and grain boundaries. It is to exclude the effect of such features that the kernel exclusion angle is used in the first place. As the exclusion angle is increased, more of the misorientations associated with such features contribute to the KANM values, and this produces a rapid rise in KANM values here.

Outside this topmost layer, increasing ω_k up to 5° results in a very few isolated areas displaying higher KANM values, as shown in the inset in Figure 8. This does not alter the observable cold work structure in a significant way. Some bulk defects, such as pores or dirt on the sample surface, also become visible on KANM maps and cause spikes in the depth profiles, as can be seen in Figure 9.

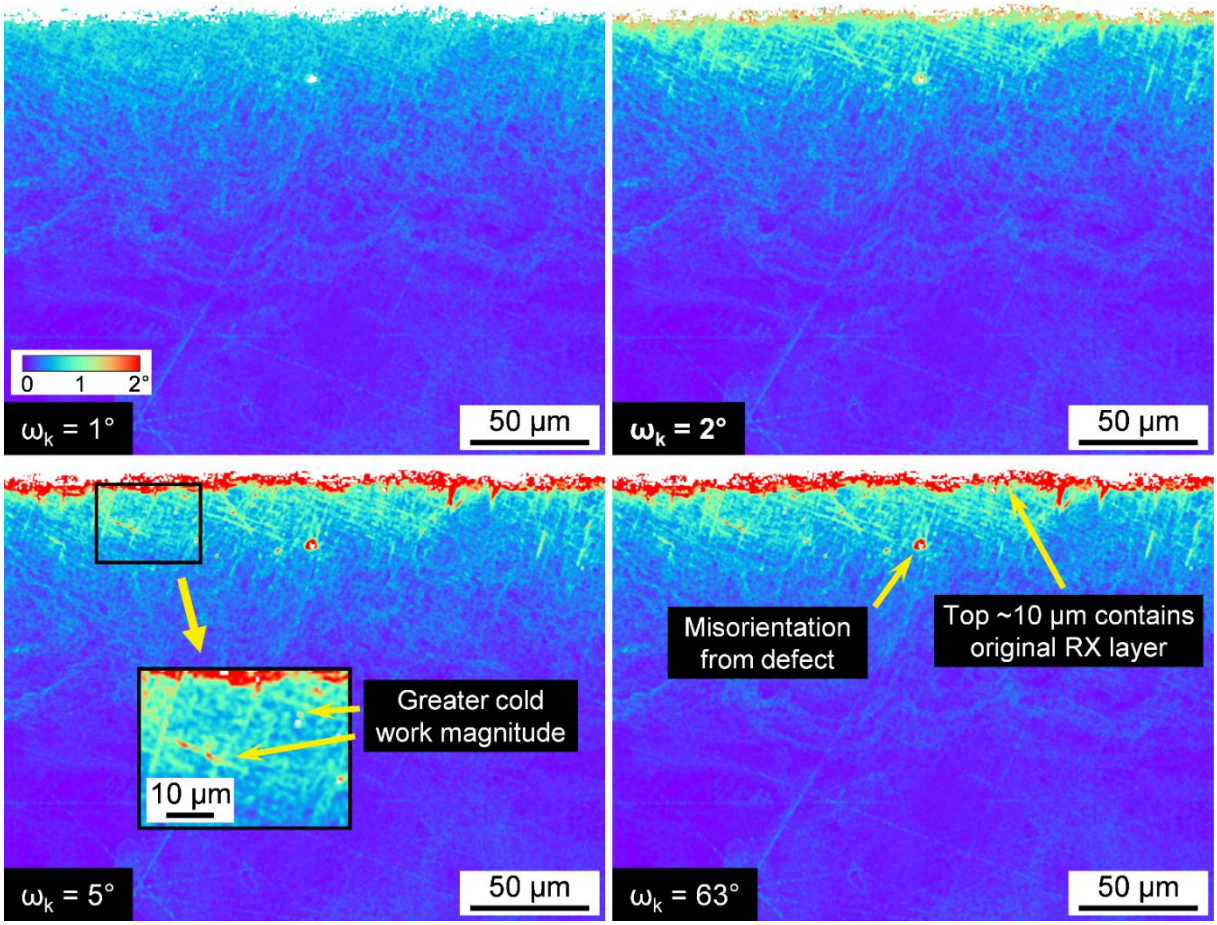

Figure 8: *KANM maps obtained from a single EBSD scan using different kernel exclusion angles. Below 2°, which was used as the standard value, the KANM map displays fewer details of the cold work structure. At higher angles, most of the difference in comparison with ω_k = 2° is in the topmost ~5 µm of material, which contains the most defects which may contribute to KANM. Beyond this layer, only small sections of slip bands and some defects display higher KANM values at higher ω_k. The upper limit of the colour scale is 2° in all maps, so that pixels with KANM values above this are assigned the same saturated red colour.*

The baseline misorientation is not affected by the exclusion angle, and is equal to 0.133° in all cases. Cold work depth rises by ~2% from ω_k rising from 1° to 63°. It is only affected because the position of zero depth, defined as the first scan line containing at least 50 % non-NaN KANM values, shifts upward as the kernel exclusion angle increases. This is because at lower exclusion angles, more pixels near to the sample surface are given NaN KANM values due to their kernels containing only misorientations above ω_k.

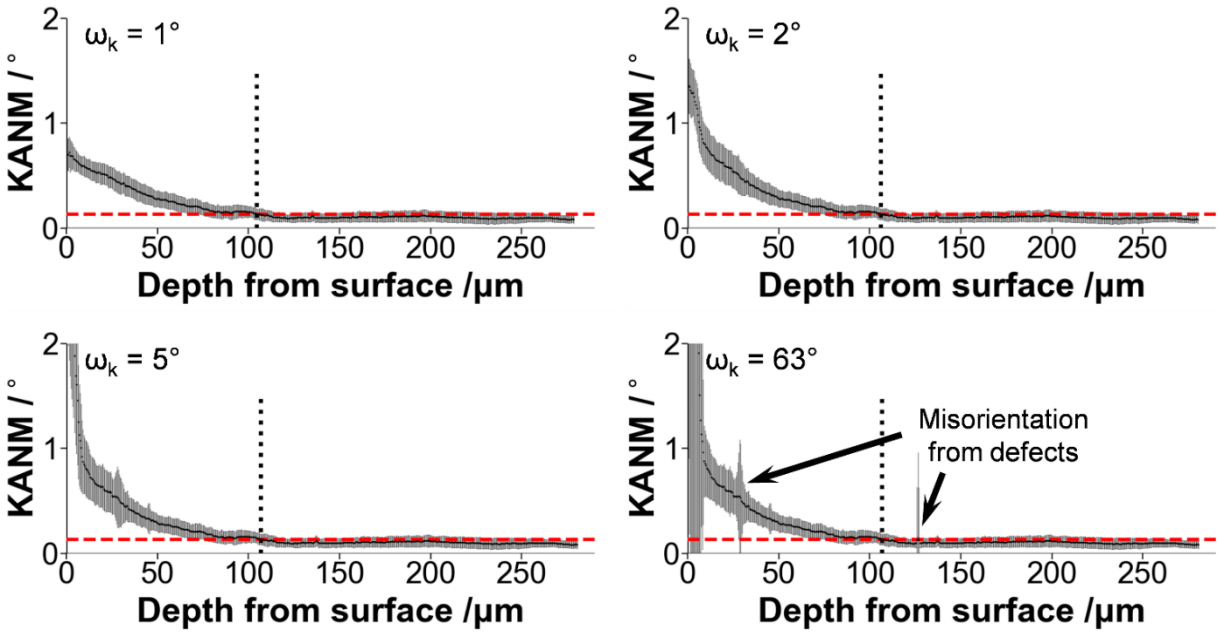

***Figure 9**: Depth profiles of the KANM maps in Figure 8 for different values of the kernel exclusion angle. Cold work depths are indicated by dotted vertical lines, and baselines are shown by dashed red lines. At ω_k = 1°, the mean KANM values up to a depth of ~35 µm are lower compared with ω_k = 2°. When ω_k is further increased above 2°, only the top ~10 µm of the profile is significantly affected, except for spikes associated with defects in the bulk material.*

The cold work magnitude is affected more strongly by changes in ω_k than the cold work depth. The variation of the cold work magnitude with ω_k in the EBSD scan in Figure 8 is plotted in Figure 10. The dependence of the cold work magnitude on the kernel exclusion angle is very well approximated by a logarithmic function, because the magnitude increases with increasing exclusion angle, but at a much reduced rate at higher angles. This is because the higher the kernel exclusion angle, the less chance there is of such a misorientation being present, so that higher kernel exclusion angles add fewer and fewer new misorientations to contribute to KANM values. Overall, the magnitude increases by a factor of about ×2.5 between ω_k = 1° and ω_k = 63°. Above ω_k = 2°, however, this increase is driven largely by higher misorientations at or near the surface, and does not provide further insight into the distribution of the bulk cold work. This is clearly seen in the maps in Figure 8. Therefore 2° does indeed appear to be an optimal value to choose for the kernel exclusion angle ω_k for this material. In samples with a different crystal structure or different processing, other values may have to be selected by the user. It should also be noted that in polycrystalline samples, where grain boundaries extend throughout the bulk of the material, the effects of changing the kernel exclusion angle would be more significant for the analysis.

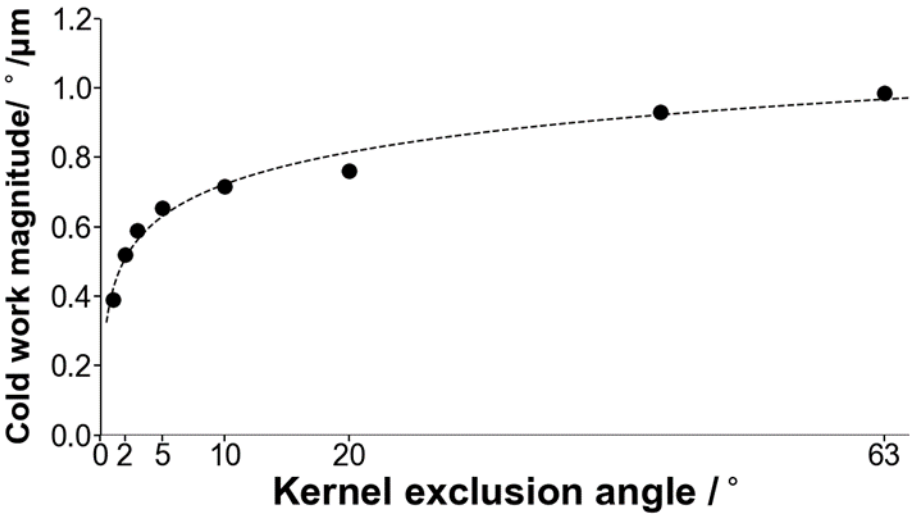

***Figure 10**: Cold work magnitude as a function of kernel exclusion angle. This magnitude increases with ω_k, but at a slower rate at higher exclusion angles. A best fit line is plotted in black, given by the equation magnitude = 1.13ln(ω_k) + 0.42.*

4.4.3. Baseline segment length

The baseline misorientation is calculated as the average of a certain number of the smallest scan line (mean + standard deviation) values. For convenience, the number of points is converted to an equivalent segment length by multiplying it by the step size of the scan. In this way, for a baseline segment length of 50 µm, which was the standard value used for maps with a depth of 300 µm and a step size of 0.5 µm/pixel, the baseline is the average of the smallest 100 (mean + standard deviation) values. The use of the least values in a scan was adopted to mitigate the effects of far-field noise (see Section 3.3.1). With this method, however, the baseline becomes directly dependent on the length of the baseline segment, since longer segments will always mean that higher KANM values contribute to the baseline. This is shown in Figure 11, where baseline KANM values and cold work depths are plotted against different baseline segment lengths for a single EBSD scan.

With the rise in the baseline, the estimate of the depth of cold work undergoes a consequent fall. Between segment lengths of 10 and 50 µm, a drop from 113 to 106 µm occurs (~7 %), but from 50 to 150 µm, there is a decrease of only 2 µm, or approximately 2 %. When the segment length is set to 200 µm, there is a more rapid drop to 90 µm cold work depth. This especially large drop occurs because there are now several positions where the scan line means become less than the baseline, and the median of these is taken as the cold work depth. This can be seen in Figure 12, where the candidate cold work depth positions for a segment length of 200 µm are plotted, together with the depth as defined for a segment length of 50 µm.

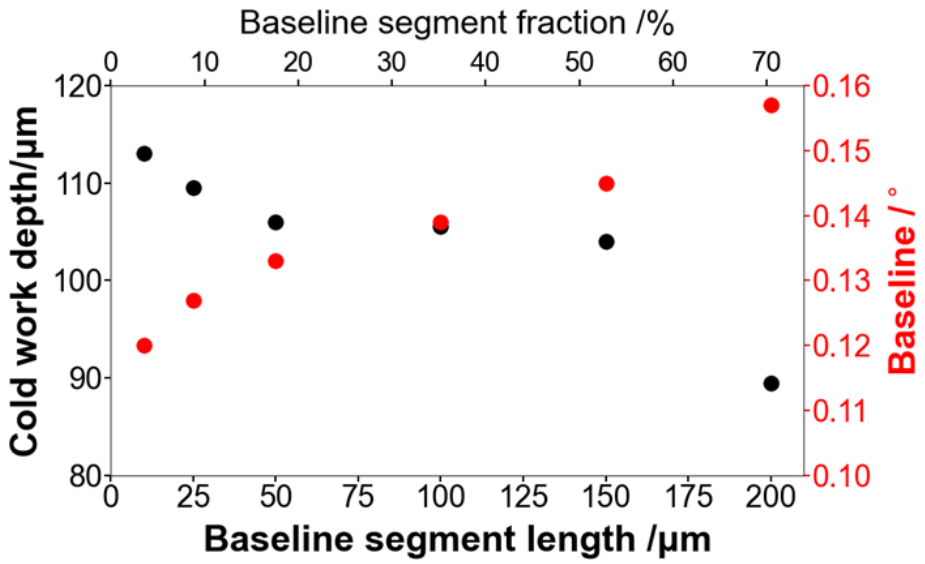

***Figure 11**: Cold work depth and baseline KANM values as a function of the baseline segment length for a single EBSD scan. The baseline segment length is plotted both in µm and as a fraction of the depth of the KANM map. Larger segment lengths lead to higher baseline misorientations and therefore to lower cold work depth measurements.*

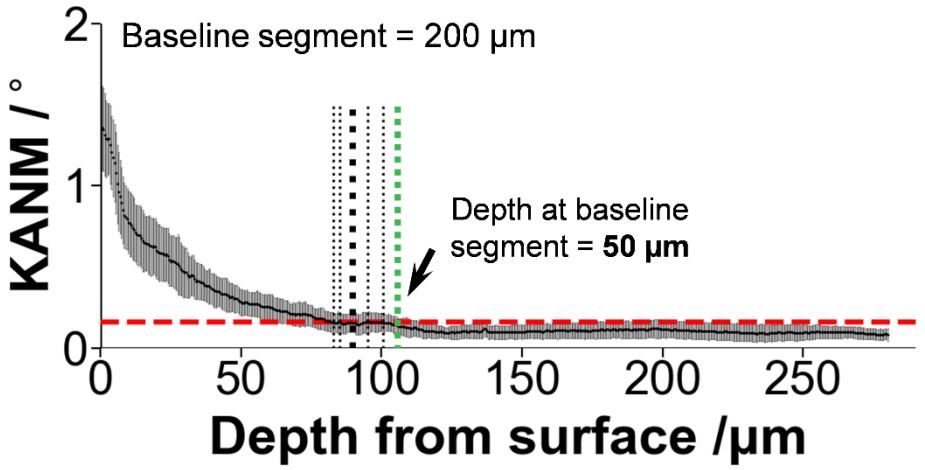

***Figure 12**: Depth profile of the KANM map used to derive the data plotted in Figure 11, with the baseline calculated from a baseline segment of 200 µm. The baseline is shown as a horizontal red line. Due to the waviness of the depth profile, there are five depths where the scan line mean crosses below the baseline. The median of the candidate depth positions (bold) is defined as the cold work depth, following the methodology described in Section 3.3.1. On the other hand, a single cold work depth value is obtained using a segment length of 50 µm, which had resulted in a lower baseline. This position is shown as a vertical green line on the plot. It is within 5 µm of the deepest of the five candidate positions derived from a segment length of 200 µm, but the actual recorded difference in cold work depth is 16 µm.*

In summary, while there is quite a rapid drop in cold work depth from the value seen at a very small segment length of 10 µm (~3 % of the map depth), as long as there is no far-field noise, the depth stabilises around 50 µm and remains little affected until very high baseline segment lengths (200 µm, or ~70 % map depth) are reached. It can be concluded that 50 µm (~17 % of the map depth) is a reasonable segment length to use, and that the cold work depth is relatively robust with respect to minor deviations in the baseline segment away from this value, even if the exact dependence may vary from scan to scan.

5. Conclusions

A methodology of establishing estimates of the depth of cold work and its magnitude in surface hardened nickel-based superalloy single crystals from a dataset of electron backscattered diffraction images through the analysis of local misorientations has been described in detail in this paper. The methodology is shown to be robust to changes in a number of acquisition and analysis parameters, the effects of which on the outcome of the analysis have been described and discussed. The principles and processes underlying this robust methodology can be readily adapted by the user for other surface hardening techniques and other surface-hardened materials and is independent of the scanning electron microscope software package used to gather the data.

CRediT authorship contribution statement

Ivan Bogachev: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Software, Visualization, Writing – original draft, Writing – review & editing. **Kevin Knowles:** Conceptualization, Formal analysis, Supervision, Project administration, Funding acquisition, Writing – original draft, Writing – review & editing. **Grant J. Gibson:** Project administration, Supervision, Funding acquisition, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was financed by Rolls-Royce plc under RR studentship G100397. The authors would like to thank St Catharine's College, Cambridge for providing additional financial support.

Appendix: Python script

The full Python script used to perform the local misorientation analysis is given below, with each line of code numbered. Minor explanatory comments are given throughout to accompany the relevant functions. It should be noted that the script has not been optimised for efficiency.

```
1  # The modules necessary to run the script are imported below.
2  import numpy as np
3  import math
4  import matplotlib.pyplot as plt
5  import warnings
6  from matplotlib_scalebar.scalebar import ScaleBar
7
8  # The inputs of the function get_q() are the three Bunge Euler angles ϕ1, Φ and ϕ2, in radians.
9  # The output is the quaternion vector obtained from these angles, in numpy array format.
10 def get_q(phi1, PHI, phi2):
11     q0 = np.cos(PHI/2)*np.cos((phi1+phi2)/2)
12     q1 = np.sin(PHI/2)*np.cos((phi1-phi2)/2)
13     q2 = np.sin(PHI/2)*np.sin((phi1-phi2)/2)
14     q3 = np.cos(PHI/2)*np.sin((phi1+phi2)/2)
15     return np.array([q0, q1, q2, q3])
16
17 # Rots_list is a list of the quaternions representing each of the 48 cubic symmetry operations.
18 Rots_list = np.array([[1,0,0,0],[-1,0,0,0],[0,1,0,0],[0,-1,0,0],[0,0,1,0],[0,0,-1,0],[0,0,0,1],[0,0,0,-1],
19     [1/np.sqrt(2),0,0,1/np.sqrt(2)],[1/np.sqrt(2),0,0,-1/np.sqrt(2)],[-1/np.sqrt(2),0,0,1/np.sqrt(2)],
20     [-1/np.sqrt(2),0,0,-1/np.sqrt(2)],[1/np.sqrt(2),1/np.sqrt(2),0,0],[1/np.sqrt(2),-1/np.sqrt(2),0,0],
21     [-1/np.sqrt(2),1/np.sqrt(2),0,0],[-1/np.sqrt(2),-1/np.sqrt(2),0,0],[1/np.sqrt(2),0,1/np.sqrt(2),0],
22     [1/np.sqrt(2),0,-1/np.sqrt(2),0],[-1/np.sqrt(2),0,1/np.sqrt(2),0],[-1/np.sqrt(2),0,-1/np.sqrt(2),0],
23     [0,1/np.sqrt(2),1/np.sqrt(2),0],[0,1/np.sqrt(2),-1/np.sqrt(2),0],[0,-1/np.sqrt(2),1/np.sqrt(2),0],
24     [0,-1/np.sqrt(2),-1/np.sqrt(2),0],[0,1/np.sqrt(2),0,1/np.sqrt(2)],[0,1/np.sqrt(2),0,-1/np.sqrt(2)],
25     [0,-1/np.sqrt(2),0,1/np.sqrt(2)],[0,-1/np.sqrt(2),0,-1/np.sqrt(2)],[0,0,1/np.sqrt(2),1/np.sqrt(2)],
26     [0,0,1/np.sqrt(2),-1/np.sqrt(2)],[0,0,-1/np.sqrt(2),1/np.sqrt(2)],[0,0,-1/np.sqrt(2),-1/np.sqrt(2)],
27     [1/2,1/2,1/2,1/2],[-1/2,-1/2,-1/2,-1/2],[-1/2,1/2,1/2,1/2],[1/2,-1/2,-1/2,-1/2],
28     [1/2,-1/2,1/2,1/2],[-1/2,1/2,-1/2,-1/2],[1/2,1/2,-1/2,1/2],[-1/2,-1/2,1/2,-1/2],
29     [1/2,1/2,1/2,-1/2],[-1/2,-1/2,-1/2,1/2],[-1/2,-1/2,1/2,1/2],[1/2,1/2,-1/2,-1/2],
30     [-1/2,1/2,-1/2,1/2],[1/2,-1/2,1/2,-1/2],[-1/2,1/2,1/2,-1/2],[1/2,-1/2,-1/2,1/2]])
31
32 # The input for quat_mult() are two quaternions, Q and q, in numpy array format.
33 # The output is the quaternion which results from their multiplication, in numpy array format.
34 def quat_mult(Q,q):
35     r1 = Q[0]*q[0]-Q[1]*q[1]-Q[2]*q[2]-Q[3]*q[3]
36     r2 = Q[0]*q[1]+Q[1]*q[0]+Q[2]*q[3]-Q[3]*q[2]
37     r3 = Q[2]*q[0]+Q[0]*q[2]+Q[3]*q[1]-Q[1]*q[3]
38     r4 = Q[0]*q[3]+Q[3]*q[0]+Q[1]*q[2]-Q[2]*q[1]
39     return np.array([r1,r2,r3,r4])
40
41 # The input for Omega(), Q1 and Q2, are two quaternions in numpy array format.
42 # The output, omega, is the misorientation angle between the input quaternions, in degrees.
43 def Omega(Q1, Q2):
```

```
        omega = (180/np.pi)*2*np.arccos(np.dot(Q1,Q2)/np.sqrt(np.dot(Q1,Q1)*np.dot(Q2,Q2)))
        if omega >=62.8:
            omega_vars = []
            for Q in Rots_list:
                Q2_new = quat_mult(Q,Q2)
                omega_vars.append((180/np.pi)*2*np.arccos(np.dot(Q1,Q2_new)/np.sqrt(np.dot(Q1,
                Q1)*np.dot(Q2_new,Q2_new))))
            omega_vars = np.array(omega_vars)
            omega = np.amin(omega_vars)
            return omega

    # The function loadEBSD() takes as input the following values: filename, in string format,
    # which is the path to the .txt file containing the raw EBSD data; Xmax and Ymax,
    # which are, respectively, the width and height, in pixels, of the EBSD dataset;
    # sep is the separator character between individual entries in the .txt file, either a comma
    # (which is set as the standard value) or a tab space. count (integer) is the number of lines in
    # the header of the .txt file.
    # The output, QUAT, is a numpy array of the quaternions of each pixel in the EBSD dataset.
    # Any non-indexed pixels will be equal to np.nan, which is a NaN (not a number) value.
    # Note that this function is only written to be valid for the output of the Oxford Instruments
    # AZtec 4.1 acquisition software.
    def loadEBSD(filename, Xmax, Ymax, sep = ',', count = 3):
        file = open(filename, "r")
        lines = file.readlines()
        linelist = []
        for l in lines[count:]:
            linelist.append(l.split(sep))
        file.close()
        QUAT = np.full((Xmax,Ymax,4), np.nan)
        for l in linelist:
            if l[2] != "0" and int(l[0])<Xmax and int(l[1])<Ymax:
            Phi1 = float(l[2])
            pHI = float(l[3])
            Phi2 = float(l[4])
            QUAT[int(l[0]), int(l[1])] = get_q(Phi1,pHI,Phi2)
        return QUAT

    # The input variables for the function get_KANM() are an array of quaternions
    # (as produced by loadEBSD()), the kernel size k (integer) and the kernel exclusion
    # angle k_exc (in degrees). threshold controls the fraction of indexed pixels in a
    # pixel row or column, below which that row or column is deleted. This is set standardly
    # to 1% indexed pixels, but can be changed (if no deletion is desired, threshold can be
    # set to -1). Filter is a Boolean variable which controls whether filtering to remove
    # isolated indexed pixels is performed (True) or not (False).
    # The output is a numpy array of KANM values, in degrees.
    def get_KANM(quat, k, k_exc, threshold = 0.01, Filter = True):
        Xmax = quat.shape[0]
        Ymax = quat.shape[1]
        KANM = np.full((Xmax,Ymax),np.nan)
        r = k//2
        for i in range(Xmax):
```

```
        for j in range(Ymax):
            if math.isnan(quat[i,j,0]) != True:
                kernel = np.full((k,k,4), np.nan)
                for p in range(i-r, i+r+1):
                    for o in range(j-r, j+r+1):
                        if p>=0 and o>= 0:
                            try:
                                kernel[p-i+r, o-j+r] = quat[p,o]
                            except IndexError:
                                pass
                omegas = []
                for g in range(k-1):
                    for h in range(k):
                        Q0 = kernel[g,h]
                        omegas.append(Omega(Q0, kernel[g+1,h]))
                        try:
                            omegas.append(Omega(Q0, kernel[g,h+1]))
                            omegas.append(Omega(Q0, kernel[g+1,h+1]))
                        except IndexError:
                            pass
                        if h != 0:
                            omegas.append(Omega(Q0, kernel[g+1,h-1]))
                for h in range(k-1):
                    Qk = kernel[k-1,h]
                    omegas.append(Omega(Qk, kernel[k-1,h+1]))
                omegas = np.array(omegas)
                omegas = omegas[~np.isnan(omegas)]
                omegas[omegas > k_exc] = np.nan
                with warnings.catch_warnings():
                    warnings.simplefilter("ignore", category=RuntimeWarning)
                    KANM[i,j] = np.nanmean(omegas)
    if Filter == True:
        t = 1
        for i in range(t, Xmax-t):
            for j in range(t, Ymax-t):
                if math.isnan(KANM[i,j]) != True:
                    nancount = 0
                    for p in range(i-t, i+t+1):
                        for o in range(j-t, j+t+1):
                            if math.isnan(KANM[p,o]) == True:
                                nancount += 1
                    if nancount >=6:
                        KANM[i,j] = np.nan
    nan_ys = []
    for j in range(Ymax):
        line = KANM[:,j]
        vals = line[~np.isnan(line)]
        if vals.size/line.size < threshold:
            nan_ys.append(j)
    nan_ys = np.array(nan_ys, dtype = int)
    nan_xs = []
```

```
        for i in range(Xmax):
            line = KANM[i,:]
            vals = line[~np.isnan(line)]
            if vals.size/line.size < threshold:
                nan_xs.append(i)
        nan_xs = np.array(nan_xs, dtype = int)
        if nan_xs.size>0:
            KANM = np.delete(KANM, nan_xs, 0)
        if nan_ys.size>0:
            KANM = np.delete(KANM, nan_ys, 1)
        return KANM

# The function plot_map() takes a 2D numpy array of misorientation data (MO),
# and plots a colour map of it. The other inputs are as follows. yrange: a tuple of
# the lower and upper ranges of the colour scale, in degrees. step: the step size of
# the input array, in µm/pixel. saveloc: the pathway, in string format, to a location
# where the image is to be saved; if not given in string format, the image is not saved.
# figsize: a tuple of the dimensions of the image to be saved, in inches.
# dpi: the resolution of the image (dots per inch). CBAR: a tuple containing the font
# sizes of, respectively, the tick marks and the label of the colour bar. If CBAR = False,
# no colour bar is plotted. scbar: a tuple with, respectively, the size of the scale bar,
# in µm, and the font size. If scbar = False, no scale bar is plotted. cmap: the colour
# scheme used for the map, in string format, from one of the pre-existing matplotlib
# colour schemes.
def plot_map(MO, yrange, step = 0.5, saveloc = 0, figsize = (21,16), dpi = 50, CBAR = (70,80),
scbar=(100,120), cmap = 'rainbow'):
    plt.figure(figsize=figsize, dpi = dpi)
    plt.imshow(np.transpose(MO), cmap = cmap, vmin = yrange[0], vmax = yrange[1])
    plt.axis('off')
    if scbar != False:
        scalebar = ScaleBar(step, 'um', location = 4, pad = 1.5, border_pad = 1, scale_loc = 'top',
        font_properties = {'size':scbar[1]}, fixed_value = scbar[0])
        plt.gca().add_artist(scalebar)
    if CBAR != False:
        ax = plt.gca()
        from mpl_toolkits.axes_grid1.inset_locator import inset_axes
        axins = inset_axes(ax, width="25%", height="7.5%", loc="lower left", borderpad=15)
        axins.xaxis.set_ticks_position("bottom")
        bar = plt.colorbar(cax = axins, orientation = "horizontal", ticks = np.linspace(yrange[0],
        yrange[1], 3))
        bar.ax.tick_params(labelsize=CBAR[0])
        bar.set_label('KANM /$^\circ$', size=CBAR[1])
    plt.subplots_adjust(left=0, bottom=0, right=1, top=1)
    if type(saveloc) == str:
        plt.savefig(saveloc)
    plt.show()

# Inputs for get_depth() are: MO1: the numpy array of KANM values.
# step: the step size of the scan, in µm/pixel. base (float, in µm): length of the
# baseline segment, in µm, to be used to define the baseline. If base = False,
# the baseline segment length is taken as 1/6 of the size of the in-depth axis (Y).
```

```
    # pos: an integer value which chooses, from the list of the candidate depth
    # positions, which one will be defined as the cold work depth. If pos = False,
    # then the depth value nearest to the median is selected.
    # The outputs of the function are as follows. defdepth: the depth of cold work
    # (in µm), as selected using the argument pos. depths: numpy array of all the
    # depths where the mean of the scan lines crosses below the baseline, all such
    # positions being candidates for the depth of cold work.
    # baseval: the value of the baseline, in degrees.
    # xaxis, meanMO, errMO: numpy arrays containing, respectively, the depth axis
    # (in µm); the mean KANM values of the scan lines; and the error
    # (standard deviation) in the KANM values in each scan line, all to be used for
    # plotting depth profiles.
    def get_depth(MO1, step, base = False, pos = False):
        MO = np.copy(MO1)
        (Xmax, Ymax) = MO.shape
        nan_ys = []
        for j in range(Ymax):
            line = MO[:,j]
            vals = line[~np.isnan(line)]
            if vals.size/line.size < 0.5:
                nan_ys.append(j)
        nan_ys = np.array(nan_ys, dtype = int)
        MO = np.delete(MO, nan_ys, 1)
        meanMO = np.nanmean(MO, axis = 0)
        errMO = np.nanstd(MO, axis = 0)
        (Xmax, Ymax) = MO.shape
        if base == False:
            base = Ymax/6*step
        N = int(base/step)
        basevals = np.copy(meanMO) + np.copy(errMO)
        basevals.sort()
        baseval = np.mean(basevals[0:N])
        xaxis = np.linspace(0,Ymax-1,Ymax)*step
        try:
            greater = np.where(meanMO>baseval)[0]
            poses = []
            for i in range(greater.size-1):
                if greater[i+1] != greater[i] + 1:
                    poses.append(greater[i]+1)
            poses.append(greater[-1]+1)
            depths = np.array(poses)*step
            if pos == False:
                defdepth = depths[depths.size//2]
            else:
                defdepth = depths[pos]
        except:
            defdepth = np.nan
            depths = np.array([])
        return defdepth, depths, baseval, xaxis, meanMO, errMO

    # The function plot_depth is used to plot the depth profile of a KANM map,
```

```
248     # and to print out the cold work depth, along with the list of candidate depth
249     # positions (in µm) and the baseline cold work (in degrees), as found using get_depth().
250     # Inputs: MO is the array of KANM values as produced by get_KANM.
251     # yrange is a tuple of the boundaries of the y (KANM) axis, in degrees e.g. (0,2) would
252     # set the y-axis to go from 0 to 2°. Similarly, xrange is a tuple of the limits of the x (depth)
253     # axis, in µm. step, base and pos are the same as in get_depth(), and are passed
254     # directly to it in this function. saveloc is the path to the location where the image is to
255     # be saved, in string format (if input as a number, the image will not be saved).
256     # fontsizes is a tuple of the font sizes of the axes labels and tick mark labels, respectively.
257     # figsize is a tuple of the dimension of the image. dpi is the resolution of the
258     # image, in dots per inch. plot_all is an argument which, if set to True, will make
259     # the function plot all the candidate cold work depths on one image, in case
260     # there is more than one such depth value.
261     def plot_depth(MO, yrange, xrange, step = 0.5, base = False, pos = False, saveloc = 0,
262     fontsizes = (85,75), figsize = (21,10.4), dpi = 50, plot_all = False):
263         means = get_depth(MO, step = step, base = base, pos = pos)
264         plt.figure(figsize=figsize, dpi = dpi)
265         plt.errorbar(means[3], means[4], yerr = means[5], color = 'k', marker = '.', ls = '', capsize=2,
266         elinewidth = 1)
267         plt.xlabel('Depth from surface /\u03BCm', size = fontsizes[0], fontweight = 'bold',
268         labelpad = 20)
269         plt.ylabel('KANM /$^\circ$', size = fontsizes[0], fontweight = 'bold')
270         plt.box(False)
271         plt.axvline(0, linewidth = 4, color = 'k')
272         plt.axhline(0, linewidth = 4, color = 'k')
273         plt.ylim(yrange[0], yrange[1])
274         plt.xlim(xrange[0], xrange[1])
275         plt.xticks(np.arange(xrange[0], xrange[1]+0.1, round(xrange[1]/6*21/figsize[0]/10,0)*10),
276         size =    fontsizes[1])
277         plt.yticks(np.arange(yrange[0], yrange[1]+0.01, round(yrange[1]/2*10.4/figsize[1]*2,0)/2),
278         size      = fontsizes[1])
279         ax = plt.gca()
280         ax.tick_params(width=3, length = 15, direction = 'out')
281         plt.axhline(y = means[2], color='r', linestyle='--', linewidth = 8)
282         if math.isnan(means[0]) != True:
283             plt.axvline(x = means[0], color='k', linestyle=':', linewidth = 10, ymax = 0.75)
284             if plot_all == True:
285                 for d in means[1]:
286                     plt.axvline(x = d, color='k', linestyle=':', linewidth = 10, ymax = 0.75)
287         if figsize == (21, 10.4):
288             plt.subplots_adjust(top=0.963,bottom=0.25,left=0.097,right=1)
289         else:
290             plt.tight_layout()
291         print("Defined depth: ", means[0], "\u03BCm"))
292         print("All depth candidates: ", means[1], "\u03BCm")
293         print("Baseline: ", round(means[2],3), “deg”)
294         if type(saveloc) == str:
295             plt.savefig(saveloc)
296         plt.show()
297
298     # For the function get_mag(), which calculates the cold work magnitude in
```

```
# a KANM map, the inputs are: the array of KANM values, the step size of
# the KANM map (in µm/pixel), the baseline segment length (base, in µm) and
# the position of the depth value in the list of candidate depths to be chosen as
# the cold work depth. base and pos are the same as in the function get_depth(),
# and, if set to False, will make, respectively, the base equal 1/6 of the map depth,
# and pos be the median position in the list of candidate depths. For convenience,
# the magnitudes, in degrees per µm, resulting from using each of the depth
# candidate positions are printed and returned in a list as the output of the function.
def get_mag(MO, step = 0.5, base = False, pos = False):
    means = get_depth(MO, step = step, base = base, pos = pos)
    M = means[4]
    B = means[2]
    Ds = means[1]
    mags = []
    for i in range(len(Ds)):
        D = Ds[i]
        mag_vals = M[:int(D/step)-1] - B
        mags.append(np.round(np.sum(mag_vals)/D, decimals = 3))
    print("All depth candidates: ", means[1], "\u03BCm")
    print("Corresponding magnitudes: ", mags, "deg/\u03BCm")
    return mags

# All the necessary variables are defined here: the pathway to the EBSD data (filename),
# the map dimensions, xmax and ymax (in pixels), the step size, step (in µm/pixel),
# the kernel size and the kernel exclusion angle (in degrees). The KANM map is computed
# and plotted, the cold work depth and magnitude are calculated and printed out.
filename = "C:/Test_EBSD.txt"
xmax, ymax = (760,580)
step = 0.5
k = 3
k_exc = 2
plt.rcParams.update({'font.sans-serif':'Arial'})

quat = loadEBSD(filename = filename, Xmax = xmax, Ymax = ymax)
KANM = get_KANM(quat, k = k, k_exc = k_exc)
saveloc_map = "C:/ Test_EBSD_KANM_map.tif"
plot_map(KANM, yrange = (0,2), dpi=50, step=step, saveloc=saveloc_map, CBAR=False,
figsize=(21,16), scbar=(100,120))
plt.close()

pos = False
saveloc_depth = "C:/ Test_EBSD_KANM_depth.tif"
plot_depth(KANM, (0,2), (0,290), step=step, figsize=(21,14), pos=pos, base=False,
saveloc=saveloc_depth)
plt.close()
get_mag(KANM, step=step, pos = pos)
```

References

[1] A.J. Schwartz, M. Kumar, B.L. Adams, D.P. Field, eds., Electron Backscatter Diffraction in Materials Science, 2nd ed., Springer Science+Business Media LLC, 2009.

[2] S. Kikuchi, Diffraction of cathode rays by mica, Jpn. J. Phys. 5 (1928) 83–96.

[3] S.I. Wright, B.L. Adams, Automatic analysis of electron backscatter diffraction patterns, Met. Trans. A 23 (1992) 759–767.

[4] J.I. Goldstein, D.E. Newbury, J.R. Michael, N.W.M. Ritchie, J.H.J. Scott, D.C. Joy, Scanning Electron Microscopy and X-Ray Microanalysis, 4th ed., Springer Science+Business Media LLC, 2018.

[5] B.L. Adams, S.I. Wright, K. Kunze, Orientation imaging: the emergence of a new microscopy, Met. Trans. A 24 (1993) 819–831.

[6] D.J. Child, G.D. West, R.C. Thomson, Assessment of surface hardening effects from shot peening on a Ni-based alloy using electron backscatter diffraction techniques, Acta Mater. 59 (2011) 4825–4834.

[7] G.J. Gibson, K.M. Perkins, S. Gray, A.J. Leggett, Influence of shot peening on high-temperature corrosion and corrosion-fatigue of nickel based superalloy 720Li, Mater. High Temp. 33 (2016) 225–233.

[8] D. Kumar, S. Idapalapati, W. Wei, Microstructural response and strain hardening in deep cold rolled nickel-based superalloy for aerospace application, Procedia CIRP 71 (2018) 374–379.

[9] B. Nagarajan, D. Kumar, Z. Fan, S. Castagne, Effect of deep cold rolling on mechanical properties and microstructure of nickel-based superalloys, Mater. Sci. Eng. A 728 (2018) 196–207.

[10] S.I. Wright, M.M. Nowell, D.P. Field, A review of strain analysis using electron backscatter diffraction, Microsc. Microanal. 17 (2011) 316–329.

[11] S. Bagherifard, Enhancing the structural performance of lightweight metals by shot peening, Adv. Eng. Mater. 21 (2019) 1801140.

[12] F. Klocke, S. Mader, Fundamentals of the deep rolling of compressor blades for turbo aircraft engines, Steel Res. Int. 76 (2005) 229–235.

[13] C.C. Wong, A. Hartawan, W.K. Teo, Deep cold rolling of features on aero-engine components, Procedia CIRP 13 (2014) 350–354.

[14] A. Evans, S.B. Kim, J. Shackleton, G. Bruno, M. Preuss, P.J. Withers, Relaxation of residual stress in shot peened Udimet 720Li under high temperature isothermal fatigue, Int. J. Fatigue 27 (2005) 1530–1534.

[15] J. Hoffmeister, V. Schulze, A. Wanner, R. Hessert, G. Koenig, Thermal relaxation of residual stresses induced by shot peening in IN718, Proceedings of the 10th International Conference on Shot Peening (2008) 157–162.

[16] H.L. Cockings, B.J. Cockings, W. Harrison, M. Dowd, K.M. Perkins, M.T. Whittaker, G.J. Gibson, The effect of near-surface plastic deformation on the hot corrosion and high temperature corrosion-fatigue response of a nickel-based superalloy, J. Alloys Compd. 832 (2020) 154889.

[17] F.O. Sonmez, A. Demir, Analytical relations between hardness and strain for cold formed parts, J. Mater. Process. Tech. 186 (2007) 163–173.

[18] P.S. Prevéy, X-ray diffraction residual stress techniques, Metals Handbook, Volume 10, 380–392, American Society for Metals, Metals Park, Ohio, 1986.

[19] B. Geddes, H. Leon, X. Huang, Superalloys: Alloying and Performance, AMS International, 2010.

[20] C.T. Sims, N.S. Stoloff, W.C. Hagel, eds., Superalloys II, Wiley, 1987.

[21] I. Bogachev, K.M. Knowles and G.J. Gibson, Electron backscattered diffraction analysis of cold work in a shot peened single crystal nickel superalloy, Materialia 14 (2020) 100860.

[22] I. Bogachev, K.M. Knowles and G.J. Gibson, High temperature behaviour of a mechanically surface hardened single crystal nickel-based superalloy, Materialia 23 (2022) 101438.

[23] I. Bogachev, K.M. Knowles, G.J. Gibson, Deep cold rolling of single crystal nickel-based superalloy CMSX-4, Materialia 20 (2021) 101240.

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

[25] S. Lainé, The role of twinning in the plastic deformation of alpha phase titanium, PhD thesis, University of Cambridge, 2017.

[26] A. Morawiec, J. Pospiech, Some information on quaternions useful in texture calculations, Texture. Microstruct. 10 (1989) 211–216.

[27] K. Kunze, S.I. Wright, B.L. Adams, D.J. Dingley, Advances in automatic EBSP single orientation measurements, Texture. Microstruct. 20 (1993) 41–54.

[28] J.K. Mackenzie, M.J. Thomson, Some statistics associated with the random disorientation of cubes, Biometrika 44 (1957) 205–210.

[29] J.K. Mackenzie, Second paper on statistics associated with the random disorientation of cubes, Biometrika 45 (1958) 229–240.

[30] O.M.D.M. Messe, S. Stekovic, M.C. Hardy, C.M.F. Rae, Characterization of plastic deformation induced by shot-peening in a Ni-base superalloy, JOM 66 (2014) 2502–2515.

[31] A.S. Gill, A. Telang, C. Ye, S.R. Mannava, D. Qian, V.K. Vasudevan, Localized plastic deformation and hardening in laser shock peened Inconel alloy 718SPF, Mater. Charact. 142 (2018) 15–26.

[32] B.J. Foss, S. Gray, M.C. Hardy, S. Stekovic, D.S. McPhail, B.A. Shollock, Analysis of shot-peening and residual stress relaxation in the nickel-based superalloy RR1000, Acta Mater. 61 (2013) 2548–2559.

[33] I. Bogachev, Influence of surface hardening on single crystal nickel-based superalloys, PhD thesis, University of Cambridge, 2022.