

PAPER • OPEN ACCESS

Calculating dislocation displacements on the surface of a volume

To cite this article: B Bromage and E Tarleton 2018 *Modelling Simul. Mater. Sci. Eng.* **26** 085007

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices
to create your essential collection of books in STEM research.

Start exploring the **collection** - **download the first chapter of
every title for free.**

Calculating dislocation displacements on the surface of a volume

B Bromage and E Tarleton 

Department of Materials, University of Oxford, Parks Road, OX1 3PH, United Kingdom

E-mail: edmund.tarleton@materials.ox.ac.uk

Received 1 May 2018, revised 5 September 2018

Accepted for publication 25 September 2018

Published 29 October 2018



CrossMark

Abstract

In this paper we propose a new method for calculating the displacement field produced by an arbitrary dislocation structure on the surface of a finite convex volume. The algorithm is designed for use in 3D nodal based discrete dislocation dynamics codes, when coupling with the finite element method using the superposition principle. Starting from Barnett's solution for the displacement field generated by a triangular dislocation loop, we outline a method for evaluating the displacement field of an arbitrary dislocation structure. The algorithm can handle both internal and exited segments and does not require storing any history information. The displacement field is evaluated segment wise, in an analogous way to the stress field, allowing for easy parallelisation.

Keywords: dislocation displacement, superposition, discrete dislocation plasticity, EasyDD

(Some figures may appear in colour only in the online journal)

1. Introduction

Dislocation motion is the primary facilitator of plastic deformation in crystalline materials. In order to better understand plastic deformation at the mesoscale, discrete dislocation dynamics (DDD) codes have been developed, where dislocations are discretised into a series of straight line segments. This paper will focus on nodal DDD models [1], where each segment is of arbitrary orientation and connects two nodes, the positions of which are updated at each time



Original content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

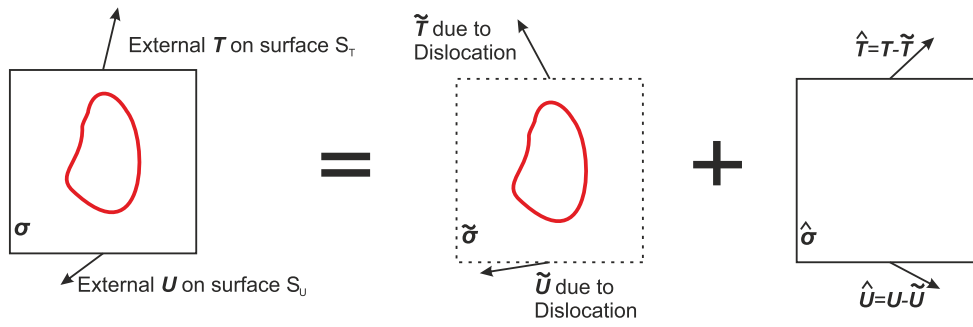


Figure 1. The superposition principle. The boundary values of tractions, \mathbf{T} , and displacements, \mathbf{U} , applied to the surfaces, S_T and S_U respectively. These are enforced by subtracting the tractions and displacements generated by the dislocations in an infinite body, $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{U}}$, and solving the modified boundary value problem, $\hat{\mathbf{T}} = \mathbf{T} - \tilde{\mathbf{T}}$ on S_T and $\hat{\mathbf{U}} = \mathbf{U} - \tilde{\mathbf{U}}$ on S_U , using FEM to obtain the corrective stress field $\hat{\sigma}$.

increment. When simulating bulk properties, periodic boundary conditions are sufficient to simulate a realistic stress state. However, to simulate micromechanical tests, requires simulating finite domains with complex geometry and boundary conditions. This can be achieved by coupling a DDD code with a finite element (FE) code. There are two commonly used methods for doing this. The superposition method [2] and the discrete–continuous model (DCM) [3]. In DCM the plastic strain generated by the dislocation motion is used in an elastic FEM code to solve for the total stress; rather than a corrective stress as in the superposition method. The DCM method therefore does not require the dislocation displacement field to be evaluated anywhere. However, the resolution of the predicted displacement field is then limited by the element size. Another challenge with DCM is avoiding ‘double counting’ of the dislocation stress field when segments are in the same finite element.

The superposition method originally proposed in 2D [2], then extended to 3D [4–6] uses FEM to calculate the corrective stress field, $\hat{\sigma}$, which arises due to any applied loading, and the loading generated by the elastic fields of any dislocations present inside the volume. First, a finite element mesh corresponding to the desired finite volume is constructed. As in any mechanical finite element problem, every FE surface node is either in S_T or S_U , depending on if tractions, \mathbf{T} , or displacements, \mathbf{U} , are specified. Then, at each time increment the tractions, $\tilde{\mathbf{T}}$, and hence the nodal forces, as well as the nodal displacements, $\tilde{\mathbf{U}}$, arising from the dislocation network in an infinite medium are evaluated analytically at the FE nodes on S_T and S_U respectively. These boundary values are then subtracted from the desired traction, \mathbf{T} , and displacement, \mathbf{U} , values. These modified boundary conditions, $\hat{\mathbf{T}} = \mathbf{T} - \tilde{\mathbf{T}}$ on S_T and $\hat{\mathbf{U}} = \mathbf{U} - \tilde{\mathbf{U}}$ on S_U , are then used to solve for the corrective stress, $\hat{\sigma}$, using an elastic FE code. The elastic stress in a finite body subject to the desired boundary conditions is finally found by simply adding the infinite body dislocation field, $\tilde{\sigma}$, and the corrective field, $\hat{\sigma}$, from FEM: $\sigma = \tilde{\sigma} + \hat{\sigma}$ as illustrated in figure 1. It should be noted that this is a purely elastic solution and that we do not account for any shape change, that is to say the boundary conditions are updated at each increment but the boundary, S , and the FE mesh remain unchanged. An interesting discussion of this point is in [7]. This requires one additional constraint which is that dislocations are prevented from exiting through any part of the surface where displacement boundary conditions are specified, S_u . Formally we are solving:

$$\left. \begin{aligned} \nabla \cdot \tilde{\sigma} &= 0 \\ \tilde{\sigma} &= \mathbf{C}: \tilde{\varepsilon} \\ \tilde{\varepsilon} &= \frac{1}{2}(\nabla \tilde{\mathbf{u}} + (\nabla \tilde{\mathbf{u}})^T) \end{aligned} \right\} \text{in } V, \quad (1)$$

$$\tilde{\sigma} \cdot \mathbf{n} = \tilde{\mathbf{T}} \text{ on } S_T, \quad (2)$$

$$\left. \begin{aligned} \tilde{\mathbf{u}} &= \tilde{\mathbf{U}} & t > 0 \\ \tilde{\mathbf{u}} &= \mathbf{0} & t = 0 \end{aligned} \right\} \text{on } S_U, \quad (3)$$

and the corrective problem

$$\left. \begin{aligned} \nabla \cdot \hat{\sigma} &= 0 \\ \hat{\sigma} &= \mathbf{C}: \hat{\varepsilon} \\ \hat{\varepsilon} &= \frac{1}{2}(\hat{\mathbf{u}} + (\nabla \hat{\mathbf{u}})^T) \end{aligned} \right\} \text{in } V, \quad (4)$$

$$\hat{\sigma} \cdot \mathbf{n} = \mathbf{T} - \tilde{\mathbf{T}} \text{ on } S_T, \quad (5)$$

$$\hat{\mathbf{u}} = \mathbf{U} - \tilde{\mathbf{U}} \text{ on } S_U \quad (6)$$

if the solution to both problems are known, then their superposition solves the desired mixed boundary value problem:

$$\left. \begin{aligned} \nabla \cdot \sigma &= \nabla \cdot (\hat{\sigma} + \tilde{\sigma}) = 0 \\ \sigma &= \hat{\sigma} + \tilde{\sigma} = \mathbf{C}: (\hat{\varepsilon} + \tilde{\varepsilon}) = \mathbf{C}: \varepsilon \\ \varepsilon &= \hat{\varepsilon} + \tilde{\varepsilon} = \frac{1}{2}(\nabla(\tilde{\mathbf{u}} + \hat{\mathbf{u}}) + (\nabla(\tilde{\mathbf{u}} + \hat{\mathbf{u}}))^T) = \frac{1}{2}(\mathbf{u} + (\nabla \mathbf{u})^T) \end{aligned} \right\} \text{in } V, \quad (7)$$

$$\sigma \cdot \mathbf{n} = \mathbf{T} \text{ on } S_T, \quad (8)$$

$$\left. \begin{aligned} \mathbf{u} &= \mathbf{U} & t > 0 \\ \mathbf{u} &= \mathbf{0} & t = 0 \end{aligned} \right\} \text{on } S_U. \quad (9)$$

Hirth and Lothe derived expressions for calculating the stress and displacement field due to a dislocation loop [8]. These expressions have been developed into expressions for an arbitrary finite segment as required in a DDD code. There are well validated analytic expressions, such as the non-singular expression derived by Cai *et al* [9] for evaluating the stress field, $\tilde{\sigma}$ associated with finite dislocation segments in an infinite medium. Furthermore, the corresponding tractions and FE nodal forces have been obtained analytically by Queyreau *et al* [10] which are ideally suited for implementation in DDD codes. Unfortunately, the expression for the displacement field, which was pioneered by de Wit [11], requires an integral over complete dislocation loops. Although it can be used for approximating some situations [12], it is unsuitable for implementation in a DDD code, where dislocations are discretised into finite segments, whose elastic fields are evaluated individually. As will be discussed in section 2, the displacement field for an isolated dislocation segment is ill defined. We are concerned here with developing a simple method for analytically evaluating the dislocation displacement on the surface of a volume, equation (3), so that the displacement boundary condition, equation (9), can be correctly enforced through equation (6). The simplest dislocation displacement field is for a triangular loop proposed by Barnett [13, 14]. Fivel and Depres [15] have developed an algorithm which breaks the dislocation network into triangular loops by generating a closure point for each segment such that screw dislocations do not contribute to the displacement. However the method requires projecting along the Burgers vector of each segment to find its closure point. When junctions have formed this is

inconvenient as it requires storing the old Burgers vectors. We have developed a new algorithm which does not require any history information.

Hence this paper aims to propose a method for calculating the displacement generated by an arbitrary dislocation structure which has been discretised into segments. The method is suited for a DDD code which utilises the superposition method. Alternatively the algorithm proposed here could simply be used for visualising the dislocation displacement field during post-processing, although it should be noted that not enforcing equation (6) during a simulation which utilises the superposition method outlined above will reduce the accuracy of the solution and could be a significant source of error for large dislocation densities.

2. Theory

2.1. Adapting Barnett's solution

In order to calculate the displacement contribution of a single finite straight dislocation segment, we require an analytic expression that describes the displacement contribution for an arbitrary segment that can be easily implemented into a dislocation dynamics code without the need to store any history information. Starting from the earlier expression in Hirth and Lothe [8], Barnett [13, 14] has derived an expression for the displacement, \vec{u} , produced by a triangular dislocation loop ABC at a given field point, X

$$\vec{u}(X) = \frac{-\vec{b}\Omega_{ABC}}{4\pi} - \frac{1-2\nu}{8\pi(1-\nu)}[f_{AB} + f_{BC} + f_{CA}] + \frac{1}{8\pi(1-\nu)}[g_{AB} + g_{BC} + g_{CA}], \quad (10)$$

$$\Omega_{ABC} = -4 \cdot \text{sgn}(\lambda_A \cdot \mathbf{n}) \arctan \left(\sqrt{\tan\left(\frac{s}{2}\right) \tan\left(\frac{s-a}{2}\right) \tan\left(\frac{s-b}{2}\right) \tan\left(\frac{s-c}{2}\right)} \right), \quad (11)$$

$$f_{AB} = (\vec{b} \times \vec{l}_{AB}) \ln \left(\frac{|\vec{R}_B|}{|\vec{R}_A|} \frac{1 + \lambda_B \cdot \vec{l}_{AB}}{1 + \lambda_A \cdot \vec{l}_{AB}} \right), \quad (12)$$

$$g_{AB} = [\vec{b} \cdot (\lambda_A \times \lambda_B)] \frac{\lambda_A + \lambda_B}{1 + \lambda_A \cdot \lambda_B}. \quad (13)$$

In these expressions, \vec{b} is the Burgers vector of the loop, ν is Poisson's ratio for the material, \mathbf{n} is the normal to the plane containing ABC , $\vec{R}_A = \vec{A} - \vec{X}$ is the vector from the field point X to A , and so on for \vec{R}_B and \vec{R}_C , and λ_A , λ_B and λ_C are their respective normalised vectors, i.e. $\lambda_A = \frac{\vec{R}_A}{|\vec{R}_A|}$ as shown in figure 2. The angles between these vectors are a , b and c , where $a = \arccos(\lambda_B \cdot \lambda_C)$, $b = \arccos(\lambda_A \cdot \lambda_C)$ and $c = \arccos(\lambda_A \cdot \lambda_B)$ and s is defined as: $s = \frac{a+b+c}{2}$. The \vec{l}_{AB} term is the normalised line direction of AB . The two terms f_{AB} and g_{AB} describe the elastic displacement due to segment AB . Note that $f_{AB} = \mathbf{0}$ for a pure screw segment and both $f_{AB} \rightarrow 0$ and $g_{AB} \rightarrow 0$ as the distance between the field point and the segment becomes much greater than the segment length. Whereas f_{AB} and g_{AB} are both undefined when the field point is on the line segment AB . The Ω_{ABC} term is the solid angle (see appendix A). Mathematically Ω is the area projected by the triangle on a unit sphere located at the field point X . Physically the solid angle accounts for the plastic slip step. The slip step on a free surface is left behind by a partially exited loop and is a prominent feature of micromechanical test data. The solid angle, Ω , is in the range $-2\pi \leq \Omega \leq 2\pi$, meaning the first term in (10) jumps between $\pm\vec{b}/2$ either side of the slip plane area bounded

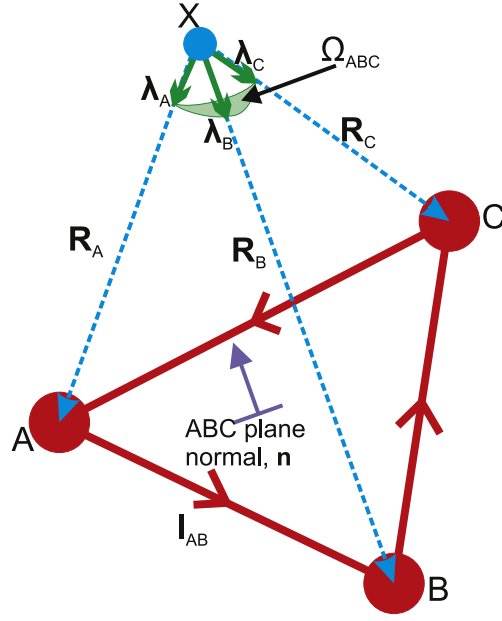


Figure 2. Barnett's solution gives the displacement as a result of any triangle, ABC , at any field point, X . The Ω_{ABC} term refers to the solid angle at X and f_{AB} and g_{AB} are vectors associated with segment AB .

by the triangular loop. Note that the displacement due to an isolated segment is ill defined, because of the Ω term. Importantly the Barnett solution outlined in equations (10)–(13) has an advantage over the solution described by Hirth and Lothe [8]. It does not make the assumption that \mathbf{b} lies in the habit plane of the loop (the plane containing A , B and C) when calculating the solid angle. That is to say the Barnett solution has no restriction on \mathbf{b} and so remains valid when the Burgers vector has a prismatic component.

Barnett's solution can be seen to work for any loop by considering the treatment of prismatic loops presented by Fivel and Depres [15]. They posit that a prismatic loop can be assigned a 'mirror loop' of opposite sign. These two prismatic loops can be thought of as four shear loops for which the displacement can be calculated. If the mirror loop is then moved to infinity, the displacement calculation for field points local to the original prismatic loop is mathematically equivalent to the same calculation carried out for this loop by itself because of how the f_{AB} and g_{AB} of the mirror loop reduce to zero while the Ω terms are equivalent.

Using Barnett's solution, a contribution to the total displacement from each segment in any closed loop can be found at any given field point by dividing the loop into triangles (figure 3(a)). Each triangle consists of one real segment which is part of the loop, and two non-real segments which join to a closure point P , common with all other segment triangles. The extra elastic terms hence cancel, and the solid angle terms for each triangle combine to give the complete solid angle of the whole loop. This works because:

$$\begin{aligned} f_{AB} &= -f_{BA} \\ g_{AB} &= -g_{BA} \\ \Omega_{ABC} &= \Omega_{ABP} + \Omega_{BCP} + \Omega_{CAP}. \end{aligned} \quad (14)$$

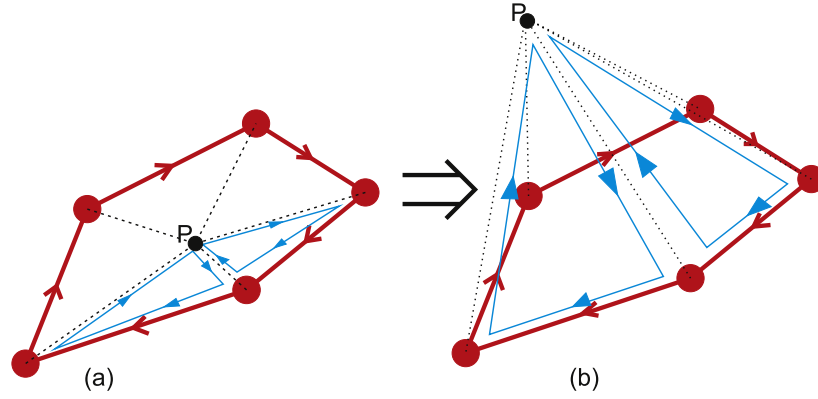


Figure 3. An arbitrary loop can be broken into triangles in order to calculate the displacement contribution. This can be done in the plane of the loop as in (a) which requires a different closure point for each slip plane. Or a unique closure point can be used which is not in the habit plane of the loop as in (b). For the correct combination of the triangles, adjacent sides of different blue triangles must be in opposite directions.

Previously, P has been chosen to be in the plane of the loop [15]. This is problematic when attempting to calculate the result for non-planar loops or multiple loops with varying positions and orientations. Fivel and Depres [15] devised a method for splitting up loops with arbitrary geometries into triangles using a projection which is on the slip plane along the Burgers vector to find a different closure points for each slip plane so that the entire loop is properly divided. The closure point is arbitrary but the total area of all virtual triangular loops is equal to the area of the original loop. Further details are in appendix B. This solution allows for calculation of displacements at any field point for the majority of dislocation structures, including segments which have cross slipped. However this solution can be inconvenient with large dislocation networks since the Burgers vector history of all segments must be tracked in order to correctly split loops that have been involved in dislocation collisions. Therefore we have developed a new method which uses a single closure point, P , for all segments in the model as illustrated in figure 3(b).

Using a single closure point has the dual advantage of working for both non-planar loops and loops which have collided, even if they have differing Burgers vectors. This is because:

$$\tilde{u}(X, \mathbf{b}_3) = \tilde{u}(X, \mathbf{b}_1) + \tilde{u}(X, \mathbf{b}_2), \quad (15)$$

where $\mathbf{b}_3 = \mathbf{b}_1 + \mathbf{b}_2$ and \mathbf{b}_1 and \mathbf{b}_2 are two different Burgers vectors. This is demonstrated more completely in appendix D.

It can be seen that the closure point now forms a pyramidal ‘tepee’ shape with the dislocation loop as the base and the closure point as the ‘peak’. For planar loops this is simply an irregular oblique pyramid. However this solution is not always correct, for while the elastic terms still cancel, there are situations in which the solid angle term is incorrectly calculated, as explained in appendix C. Whenever the field point at which the displacement is measured exists inside the polyhedron formed by the loop and the closure point, an exterior solid angle of incorrect sign is calculated instead of the desired result, i.e. $\Omega_{ABC} \neq \Omega_{ABP} + \Omega_{BCP} + \Omega_{CAP}$. If displacement is calculated piecewise by segment, it is not possible to ascertain whether or not a field point exists within such a volume, since there is insufficient information about the loop or loops to which the segment belongs. While this is true for a completely general case, for the surface of a finite volume this problem can be avoided. If a closure point is chosen so

that it is inside the volume, all field points on the surface of the volume would be outside of any polyhedron formed by the dislocation network and closure point. Therefore $\tilde{\mathbf{u}}(X)$ can be correctly calculated for every possible position of X on the surface, provided that no dislocations exit the volume. Fortunately, to couple with FEM using the superposition method, $\tilde{\mathbf{u}}$ need only be evaluated on the surface of the simulated volume. Dislocation segments are prevented from exiting through S_u , as this is unphysical and would produce an erroneous $\tilde{\sigma}$ stress to remove the slip step. However dislocations can exit the volume through any part of the surface where tractions are specified, S_T . Therefore we now need to extend the method to handle exited segments.

Before this, it should be noted that Barnett's solution gives the displacement of a loop that has swept out its current area. If we think of a real sample that has been cut and/or polished for testing, there is no displacement on the surface caused by the internal dislocation structure. At the start of a simulation before any load is applied, there should be no displacement on S_u . Subsequent displacements should only be caused by changes in the initial dislocation network that occur during the simulation. Therefore, when calculating $\tilde{\mathbf{u}}$ at time t , we must apply the displacement calculation to the dislocation network as it currently is and then subtract the result of the calculation for the initial dislocation structure such that $\tilde{\mathbf{u}}(t) = \tilde{\mathbf{u}} - \tilde{\mathbf{u}}(0)$. For example consider an initial dislocation loop which shrinks and is annihilated. This process should produce a non-zero displacement, but there is no information in the current dislocation network about the initial loop. More details are discussed in appendix B.

2.2. Calculating displacements for exited dislocations

Barnett's equation has been derived for a dislocation loop in an infinite isotropic solid. In order to calculate the displacement on the surface of the volume due to an exited dislocation segment, it is convenient to move the exited segment to an infinite distance from the volume whilst maintaining the connectivity of the loop. This issue is discussed in [16]. In a simulation, when a dislocation segment exits, it is moved to a position, that is very far (several orders of magnitude) from the surface relative to the size of the volume, designated as 'pseudo-infinity'. Extra virtual segments are then generated as necessary to connect the exited segment to the internal segments of the dislocation loop at the surface.

The direction a segment is moved when it has exited the surface must satisfy several criteria: the long range displacements must be correctly calculated, the extension of a dislocation loop must be consistent across a surfaces and the virtual segments formed must keep track of the line of the formed step on the surface. The long range displacement calculated is relative. As the angle of the projection vector is changed, the displacement on one side of the loop decreases in magnitude whilst the other side increases by the same amount, so that the degree to which the material on one side of the loop is displaced from the other is the same, namely \mathbf{b} . The consistency means that any segments in the same loop exiting out of the same surface must be projected in the same direction. Finally, virtual segments at pseudo-infinity must be able to be mapped back on to the surface they were projected from in order to recover where they exited. The simplest way to satisfy these conditions is to project along the surface normal.

To project exited nodes to the correct position, surface nodes must first be positioned correctly. This is done by finding the point, \mathbf{Q} , where a segment AB intersects the surface. If node \mathbf{A} exits and \mathbf{B} remains inside then

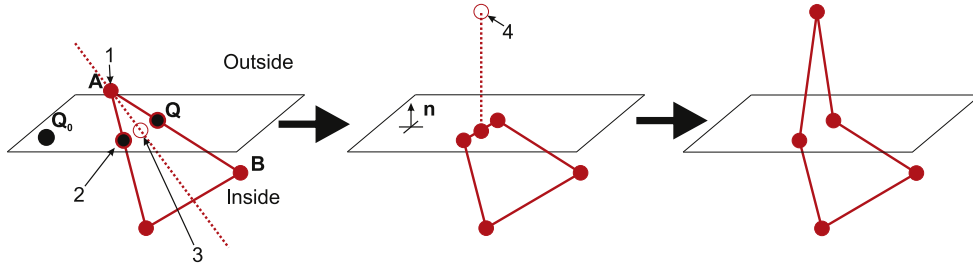


Figure 4. The process by which virtual nodes and segments are created when a node exits out of a surface. 1. A node at A which has exited is detected. 2. A new surface node at Q is created where each segment connecting the exited node to an internal node intersects the surface; calculated using equation (16). 3. The exited node is moved back onto the surface along the line of the sum of segments connected to it. 4. The exited node is moved to pseudo-infinity along the surface normal. Virtual nodes follow the line of the slip step, positioned at pseudo-infinity and can be directly projected back onto the surface along the surface normal to recover the slip step position.

$$Q = \left(\frac{(Q_0 - B) \cdot n}{(A - B) \cdot n} \right) (A - B) + B, \quad (16)$$

where A and B are the position vectors of the external and internal nodes respectively, Q_0 is an arbitrary point on the surface and n is normal to the surface through which A exited, see figure 4. It should be noted that it is important that nodes are projected consistently from the position of the slip step in the direction of the surface normal. This is because, while there is no effect on the long range displacement fields, the position and shape of the slip step is dependent on the virtual dislocation network being formed correctly. Surface nodes are also restricted so that they may only move on the surface through which the dislocation has exited. If a surface node is no longer connected to any internal dislocations because all connecting segments have exited, then it is projected to pseudo-infinity as part of the virtual network.

Once an exited dislocation node has been projected, there will be areas on the surface where the result of the displacement calculation will be incorrect due to the volume the dislocation loop creates with the closure point intersecting the surface of the simulated volume as shown in figure 5. In order to avoid this problem, adjustments must be made to the method to give the correct result.

For external parts of a dislocation structure, because it is known that virtual nodes have been projected from the surface, more information about the geometry of the loop is available when looking at individual segments. This is because

$$A = A' + Ln, \quad (17)$$

where A is the position vector of a virtual node, A' is its corresponding surface node, equivalent to the sum of the surface intersection points of all segments connected to A , $A' = \sum Q$, and L is some large factor, much greater than the largest dimension of the simulated volume, approximating infinity and n is the surface normal, as illustrated in figure 5. Since the segments are not inside the material, the elastic displacements caused by the presence of the line of the dislocation (f_{AB} and g_{AB}) are zero and so can be ignored. This just leaves the solid angle term, Ω , for the displacement arising from the plastic slip step that needs to be calculated. This must be done in such a way that the contribution to Ω from the virtual segments gives the correct value when added to the contributions from the internal segments so that the total Ω for the entire network is the solid angle expected at any given

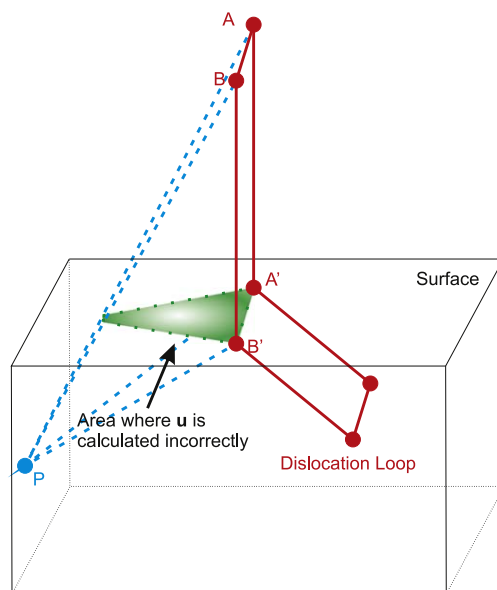


Figure 5. When calculating displacements using a common closure point, dislocation segments which have exited the volume have an area on the surface S_i associated with them within which displacements are calculated incorrectly.

field point on the surface of the simulated volume. To achieve this, the internal dislocation network must be closed at the surface by a virtual surface segment for which only the Ω term is calculated using the internal common closure point \mathbf{P} . The external part of the loop then has a unique closure point assigned to it and the correct Ω contribution can then be calculated as shown in figure 6(a). It should be noted that closure points for external parts of the loop do not have to be unique, they merely must not form a volume that intersects the surface. Assigning a common closure point to each face of the surface a dislocation segment could exit from would also work. However, when it comes to implementing in a DDD code, it was found to be quicker and easier to find a unique closure point for each exited segment than to find the face of the surface a segment has been projected from. Another advantage of assigning a unique closure point to each exited segment is that the algorithm is completely general and independent of the geometry being simulated.

The virtual parts of a dislocation network are not always simple and may consist of multiple segments at pseudo-infinity. Solid angle contributions for each of these virtual segments must be calculated. In order to do this, a sub loop must be found for all segments at pseudo-infinity. All segments (along the surface normal) joining a virtual segment to the surface can be ignored since their contribution to the displacement will be included in that of the sub-loop associated with the virtual segment (at pseudo-infinity) they connect to. Virtual nodes which are not attached to a virtual segment connected to the surface must be projected back on the the surface to find the corresponding corners of the sub-loop. An appropriate virtual surface segment must also be created to ensure the closure of the internal part of the dislocation network as shown in figure 6(b).

In the event that a junction exits the surface and there is a node at pseudo-infinity connecting to three other nodes at pseudo-infinity, this approach still functions correctly. The junction node represents the intersection of three sub-loops which will combine to give the

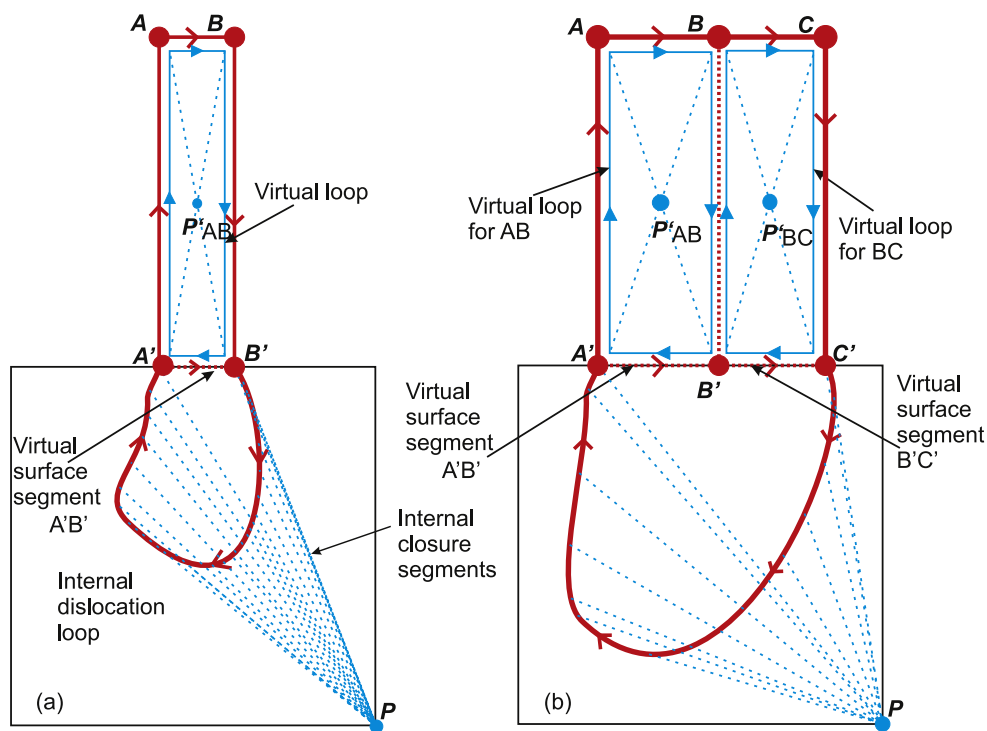


Figure 6. (a) One segment AB has exited the surface. Virtual segments are handled separately from the internal segments for the calculation of displacements by introducing a virtual surface segment connecting A' to B' to close the internal loop. Virtual surface segments use the common closure point P and like all virtual segments only contribute to Ω . Each virtual segments at pseudo-infinity e.g. AB has its own closure point e.g. P'_{AB} . (b) At a later time a second segment BC has exited which has closure point P'_{BC} . Every segment at pseudo-infinity has an associated rectangular loop assigned to it. Maintaining the sense of the loop direction ensures that the solid angles combine correctly.

correct total displacement provided the Burgers vectors and line direction are maintained. Therefore conservation of Burgers vector ensures the correct handling of any junctions present in the virtual network.

This treatment of exited segments is similar to the technique utilised by Hussein and El-Awady to simulate surface roughness evolution [12]. The key difference is that their technique for analysing surface roughness only takes exited segments into account and ignores the elastic displacement contributions from the internal dislocation segments. This results in a marginal underestimation of the height of the slip step for partially exited loops. Despite this, the technique is a good approximation, since the contribution to the height of a slip step by the internal structure is small in comparison to that of the exited segments. Additionally, the position and orientation of the slip steps is accurate. However, as dislocation density rises, the inaccuracies in \tilde{u} will increase. The method proposed here would provide better results in such a scenario as the elastic displacement from every segment is calculated. The complete algorithm for both internal and exited segments is shown below.

Algorithm 1. Pseudocode to find the displacement on the surface of a simulated volume. S is S_u during a simulation or $S_u \wedge S_t$ during post-processing. The algorithm is amenable to parallelisation over dislocation segments or field points.

Result: Dislocation displacement \tilde{u} on surface FE nodes

```

for  $\forall$  FE nodes  $\in S$  do
  for  $i = 1$  to  $N$  do //  $\forall$  dislocation segments
     $A = A(i)$  // nodal coordinates of segment
     $B = B(i)$ 
    if  $A \in V \wedge B \in V$  then // an internal segment
       $\Omega = \Omega + \text{Omega}(A, B, P)$  //  $\Omega_{ABP}$  using common closure point
       $f = f + f(A, B)$  //  $f_{AB}$ 
       $g = g + g(A, B)$  //  $g_{AB}$ 
    else
      if  $A \notin S \wedge B \notin S$  then // segment is not connected to surface
         $A' = \text{projection}(A, S)$  // project segment onto surface
         $B' = \text{projection}(B, S)$ 
         $P' = \text{centre}(A, B, B', A')$  // unique closure point
         $\Omega = \Omega + \text{Omega}(A, B, P') + \text{Omega}(B, B', P') + \text{Omega}(B', A', P') + \text{Omega}(A', A, P')$ 
        //  $\Omega_{ABB'A'}$  of the rectangular loop using its closure point  $P'$ 
         $\Omega = \Omega + \text{Omega}(A', B', P)$  // calculate  $\Omega_{A'B'P}$  for the segment projected
        onto the surface using  $P$ 
      else
        // virtual segments (parallel to  $\mathbf{n}$ ) connecting a virtual node at
        pseudo-infinity to a surface node
        // skip segment. Ignore to avoid double counting.
      end
    end
  end
   $u = \Omega + c_1 f + c_2 g$  //  $c_1$  and  $c_2$  are constants
   $\tilde{u}(t) = u - u(0)$  // subtract off  $\tilde{u}_0$  calculated at  $t = 0$ 
end

```

3. Results

Several tests of this method for calculating surface displacements were carried out to verify the method. Firstly, the displacement caused by a triangular loop was calculated using Barnett's solution on the surface of a cube containing the loop. This was compared to the results for when the displacement was calculated using several arbitrary closure points inside the cube. All of the surface displacements calculated were identical to, as shown in figure 7. The displacement generated by two loops just prior to collision was compared with the displacement after they had collided and formed a junction. There was a slight change in displacement caused by the movement of the loops but no change caused by the formation of the junction, as shown in figure 8.

The displacement on the surface due to a pair of square prismatic loops where the interstitial loop has exited along its glide prism and the vacancy loop remains inside V is shown in figure 9. This agrees with the displacement obtained by Fivel and Depres using their method [15].

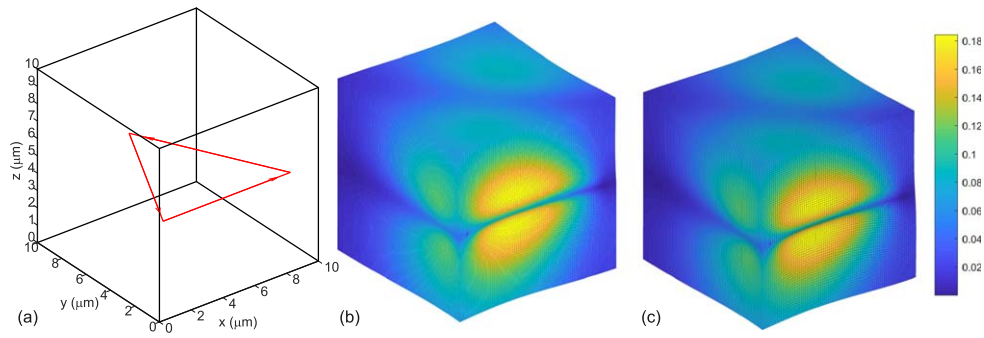


Figure 7. A triangular dislocation loop on the (001) plane with a $[110]$ Burgers vector (a) has its displacement field calculated on the surface of a cube using both Barnett's method (b) and using an arbitrary closure point (c). The results are indistinguishable so long as the closure point is inside the cube.

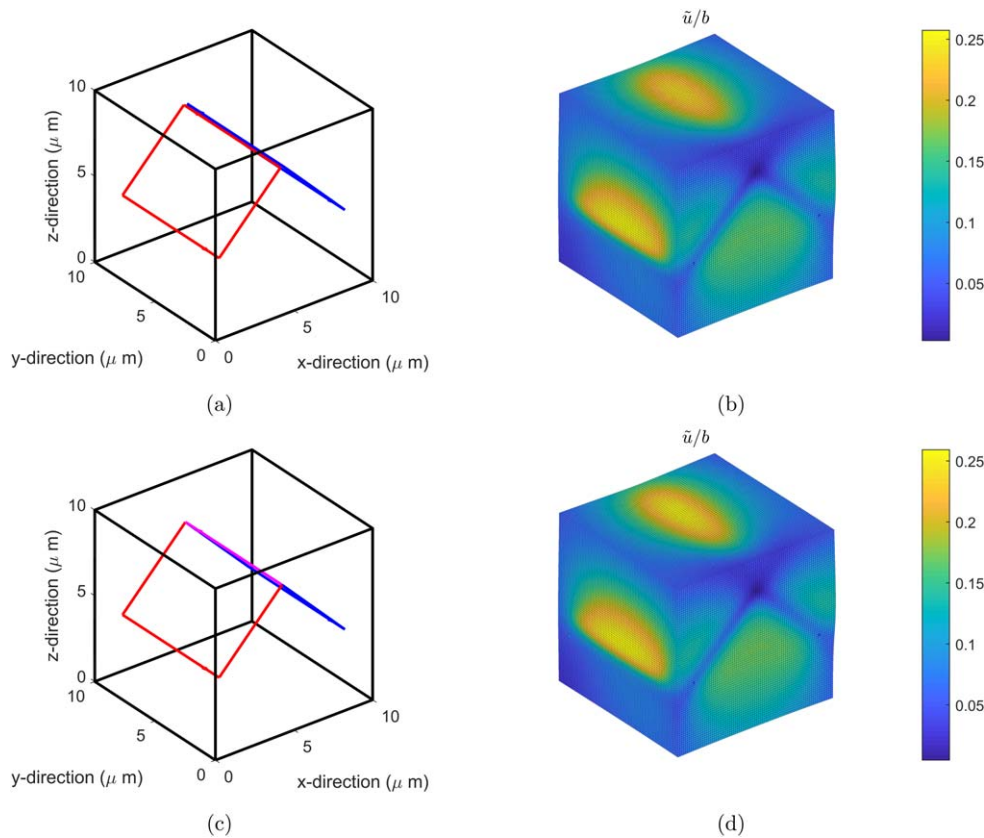


Figure 8. A cube aligned along $\langle 100 \rangle$ containing two square dislocation loops just prior to collision shown in (a) and the displacement they generate on the surface of the cube (b). After the the formation of a junction (c), the surface displacement is unchanged (d). The red loop is on the $(\bar{1}01)$ plane with a $[111]$ Burgers vector while the blue loop is on the (101) plane with a $[\bar{1}\bar{1}1]$ Burgers vector. The purple segment at the junction has a $[001]$ Burgers vector.

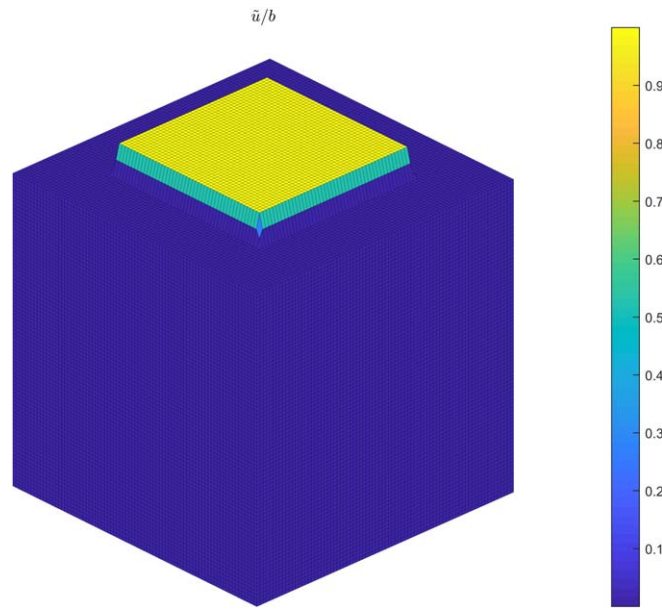


Figure 9. Displacement on the surface of a cube due to a square (001) [001] prismatic interstitial loop which has exited out of the top of the cube.

The algorithm described above was used to simulate an end loaded single crystal body-centred cubic micro cantilever beam. A new matlab based code *EasyDD* has been developed based on a modified version of *DDLlab* developed at LLNL by Arsenlis *et al* [9]. *EasyDD* is coupled with a custom written FEM code and utilises GPU acceleration [17] to enable accurate simulations on a desktop PC.

Identical simulations were performed evaluating the dislocation displacement \tilde{u} on S_u to enforce the displacement boundary conditions equation (6) and neglecting the dislocation displacements so that the only coupling between DD and FEM is through the traction boundary condition. A BCC crystal structure was chosen with Poisson's ratio and bulk modulus approximating that of tungsten. Sixty dislocation sources, ten on each {110} slip plane, were then scattered randomly throughout a single crystal cantilever with height and width of $5 \mu\text{m}$ and length of $18 \mu\text{m}$. The sources themselves must be closed loops rather than the commonly used isolated Frank–Reed source segments as this would produce erroneous \tilde{u} values due to Ω being ill defined. Physically this is correct as a dislocation segment cannot terminate in a perfect lattice. The cantilevers were fixed at one end with a displacement of $0.22 \mu\text{m}$ applied to the other end. It should be noted that the fixed end must not have any dislocations exiting through it which was achieved by pinning dislocations at this surface when they reached it. The resulting load–displacement curves are shown in figure 10 and the dislocation networks that evolved are shown in figure 11. There was an observable difference in both the load/displacement curves and final dislocation structures.

This method may also be used in post-processing to evaluate the displacement on the entire surface $S = S_u \cup S_t$ of a volume. A rectangular cantilever, with width and height of $5 \mu\text{m}$ and length of $30 \mu\text{m}$ was simulated which was fixed at one end with a displacement of

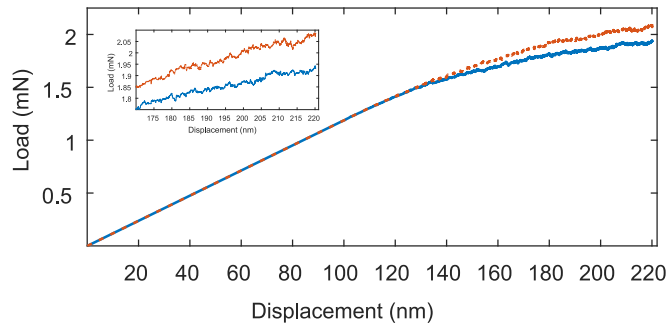


Figure 10. Simulated load–displacement curve for two identical microcantilever simulations, with modified traction boundary conditions only, equivalent to $\tilde{\mathbf{u}} = 0$, (red dashed line) and with modified traction and displacement boundary conditions $\tilde{\mathbf{u}}$ (blue line). A greater plastic response is observed when displacement boundary conditions are included in the simulation.

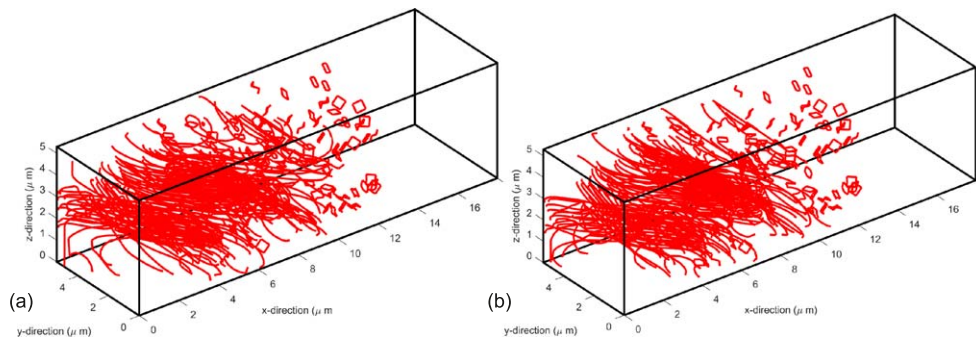


Figure 11. Dislocations network in single crystal beam bending simulations where (a) $\tilde{\mathbf{u}}$ is neglected and (b) $\tilde{\mathbf{u}}$ is used to modify the FEM boundary condition on S_u . A greater dislocation density and pile up at the neutral axis is observed in (a), despite the plastic response observed being smaller.

0.6 μm applied at opposite end. The initial dislocation structure was 60 randomly distributed dislocation loops, 10 on each $\{110\}$ slip plane, similar to the above case. After post-processing, slip steps were observed on S_t where dislocations had exited the surface as can be seen in figure 12. The complete final dislocation network consisted of 5736 segments totalling slightly more than 1 mm in length (ignoring segments of pseudo-infinite length), of which 907, with a total line length of approximately 150 μm , had exited. Both glide and cross-slip occurred.

4. Conclusion

A new method has been developed for accurately calculating the displacements on the surface of a volume due to dislocations for an arbitrary and complex dislocation network. Since the

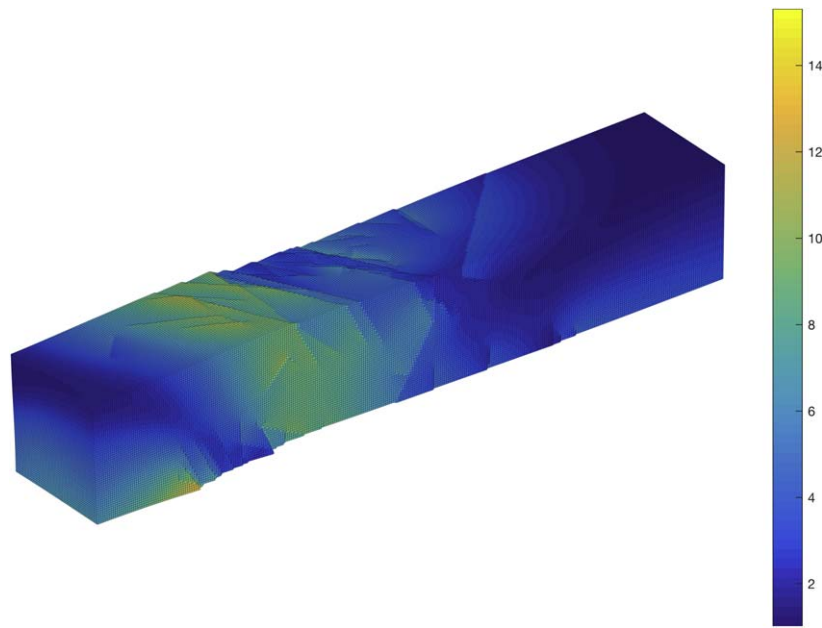


Figure 12. The dislocation displacement field evaluated on the surface of the simulated volume. Colour corresponds to the displacement magnitude. The fixed end is on the left of the image.

calculation is performed piecewise by segment, the method is readily compatible with DDD codes. A prerequisite is that connectivity of the network is maintained so that there are no segments not connected at one end. This means that a typical simple Frank–Read source with two pinned ends cannot be used as part of the initial dislocation structure. However, provided connectivity is maintained, the algorithm outlined here can calculate displacement fields for both shear and prismatic loops as well as non-planar loops, and complex junctions. Furthermore only knowledge about the current state of the dislocation segments is required alleviating the burden of having to track any history information. Provided the initial dislocation displacement field is evaluated once at the start of the simulation and subtracted off the algorithm can handle any dislocation motion.

The simplicity of the proposed method combined with its versatility mean it can be implemented in any DDD code. It can be used for both the calculation of \bar{u} in order to solve boundary problems when using the superposition method as well as analysis of surface deformation during post-processing. Neglecting the dislocation displacement when using the superposition method will generate an error which increases with dislocation density. The number of calculations required increases linearly with the number of dislocation segments present in the simulation. Therefore, for use in large scale simulations, the process should be parallelised for better performance, which is possible since calculations for individual segments are independent of each other.

Acknowledgments

ET would like to thank EPSRC for financial support through an Early Career Fellowship grant EP/N007239/1. We would also like to thank Professor Felix Hofmann and Dr Steve Fitzgerald for useful discussions.

Data availability statement

The processed data required to reproduce these findings are available to download from <http://doi.org/10.17632/92hz9nkj2r.1>

Appendix A. Solid angles

Solid angles are the three dimensional equivalent of angles in two dimensions. In two dimensions, an angle can be measured at a point where two lines meet. To find the angle in radians, a unit circle can be drawn using this point as the centre. The size of the angle is the length of the circumference between the two lines, as shown in figure A1(a).

A similar process may also be executed in three dimensions to find the solid angles at three dimensional vertices. Any point where two or more planes meet in three dimensional space may have a unit sphere placed using this point as the centre. The surface area of the sphere contained between these planes is the solid angle as shown in figure A1(b).

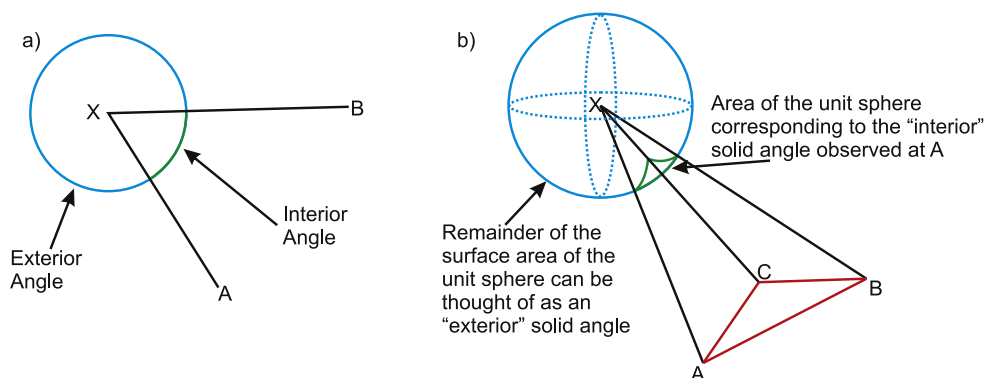


Figure A1. (a) A unit circle can be used to find the angle between lines XA and XB . This angle can be either interior or exterior, depending on which direction around the circle the measurement is carried out. (b) A unit sphere is used to measure the solid angle caused by the meeting of the planes containing XAB , XAC and XBC . It should be noted that the 'interior' solid angle is the projection of the triangle ABC onto the unit sphere. The 'exterior' solid angle is 4π minus the interior solid angle.

Appendix B. Plastic distortion

The plastic distortion tensor is unique and defined by the area swept out by the dislocation network. Although plastic distortion is not used in this paper it is interesting to note the connection between the area swept out by the loop and the solid angle. The increment in plastic distortion is

$$\Delta\beta = \sum_i \frac{\mathbf{b}_i \otimes \Delta\mathbf{A}_i}{V}, \quad (\text{B.1})$$

where the sum is over all segments in a volume V and the area swept out by a single segment in a time increment is

$$\Delta\mathbf{A}_i = \frac{1}{2} [\mathbf{l}_i^t \times \mathbf{v}_i^{(1)} + \mathbf{l}_i^{t+\Delta t} \times \mathbf{v}_i^{(2)}] \Delta t = \mathbf{n}_i \Delta A_i \quad (\text{B.2})$$

for a line segment i connecting node 1 \rightarrow 2 with nodal velocities $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$. For example, segment \mathbf{l}_{AB}^t at the start of the increment transforms to $\mathbf{l}_{AB}^{t+\Delta t} = \mathbf{l}_{AB}^t + (\mathbf{v}_B - \mathbf{v}_A)\Delta t$ at the end of the increment. This is demonstrated in figure B1 for a triangular loop ABC shown in red. It is clear geometrically that the change in area during the increment is the sum of the areas swept out by each segment,

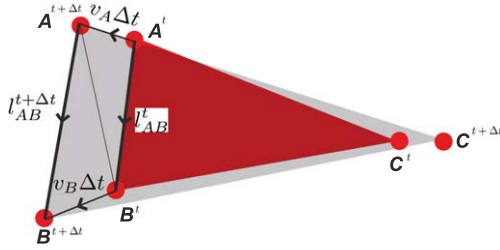


Figure B1. The area swept out by a triangular loop during a time increment is equal to the difference between the current and initial areas bounded by the dislocation loop.

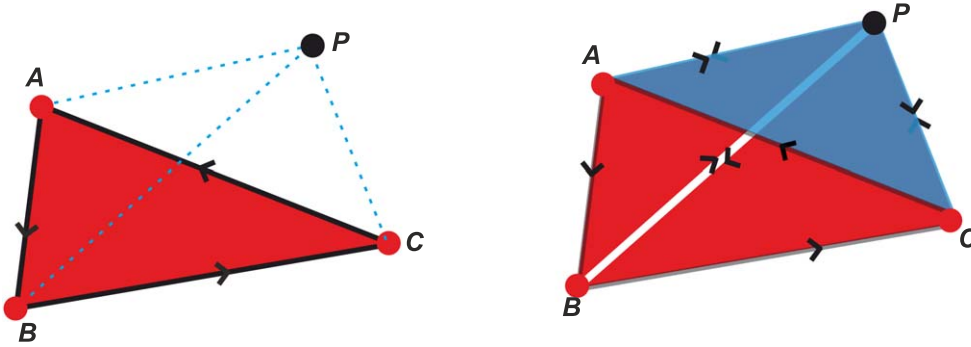


Figure B2. The area enclosed by a loop ABC is equal to the area enclosed by the three triangular loops $ABP + BCP + CAP$ generated by connecting each segment of the loop AB , BC and CA to an arbitrary closure point at P provided P is in the plane of the loop.

$$\frac{1}{2} \mathbf{l}_{AB}^{t+\Delta t} \times \mathbf{l}_{CA}^{t+\Delta t} - \frac{1}{2} \mathbf{l}_{AB}^t \times \mathbf{l}_{CA}^t = \sum_i \Delta \mathbf{A}_i = \Delta \mathbf{A}. \quad (\text{B.3})$$

By extension $A^t - A^0$ is the total area swept during a time t by a loop with initial area A^0 . Geometrically the solid angle $\Omega^t(\mathbf{X})$ is the area A^t projected onto the field point at \mathbf{X} . Therefore $\Omega^t(\mathbf{X}) - \Omega^0(\mathbf{X})$ is the swept area projected onto the field point.

The plastic distortion is unique therefore the solid angle must also be unique. However Ω is obtained by connecting every segment to an arbitrary closure point at \mathbf{P} to generate a triangular loop for each segment. The areas of each triangular loop have a sign associated with them depending on if their line direction is clockwise or anticlockwise. So that when summed, the total area is unique and equal to the area enclosed by the real dislocation loop. This was shown in [15] under the condition that the closure point were chosen anywhere in the plane of the loop as illustrated in figure B2. Clockwise loops (shown in red) have a signed area corresponding to a normal out of the page whereas the anticlockwise triangle CAP (shown in blue) has a negative area e.g. with a normal pointing into the page. Therefore when summed, some of the areas cancel, leaving only the original area ABC and the solution is unique.

Appendix C. Arbitrary closure point proof

Here we demonstrate that the closure point, required to convert a segment into a triangular loop, does not need to be in the plane of the segment which significantly simplifies the displacement calculation. Barnett's expression for the displacement of a triangular loop is:

$$\mathbf{u} = \frac{-\mathbf{b}\Omega_{ABC}}{4\pi} - \frac{1-2\nu}{8\pi(1-\nu)}[f_{AB} + f_{BC} + f_{CA}] + \frac{1}{8\pi(1-\nu)}[g_{AB} + g_{BC} + g_{CA}], \quad (\text{C.1})$$

$$f_{AB} = (\mathbf{b} \times \mathbf{l}_{AB}) \ln \left(\frac{|\mathbf{R}_B|}{|\mathbf{R}_A|} \frac{1 + \lambda_B \cdot \mathbf{l}_{AB}}{1 + \lambda_A \cdot \mathbf{l}_{AB}} \right), \quad (\text{C.2})$$

$$g_{AB} = [\mathbf{b} \cdot (\lambda_A \times \lambda_B)] \frac{\lambda_A + \lambda_B}{1 + \lambda_A \cdot \lambda_B}, \quad (\text{C.3})$$

where \mathbf{b} is the Burgers vector of the loop, ν is Poisson's ratio for the material and \mathbf{l}_{AB} is the normalised line direction of AB . Changing the line direction changes the sense of the loop and $\mathbf{u} \rightarrow -\mathbf{u}$. First consider f_{AB} :

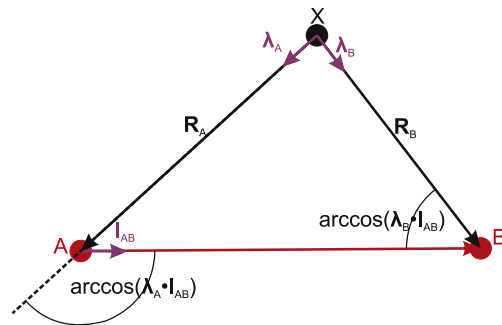


Figure C1. Triangle formed by dislocation line AB and field point X .

$$\mathbf{l}_{BA} = -\mathbf{l}_{AB} \rightarrow \mathbf{f}_{BA} = -(\mathbf{b} \times \mathbf{l}_{AB}) \ln \left(\frac{|\mathbf{R}_A|}{|\mathbf{R}_B|} \frac{1 - \lambda_A \cdot \mathbf{l}_{AB}}{1 - \lambda_B \cdot \mathbf{l}_{AB}} \right). \quad (\text{C.4})$$

As the vectors are normalised, $\lambda_A \cdot \mathbf{l}_{AB}$ and $\lambda_B \cdot \mathbf{l}_{AB}$ are the cosines of angles associated with triangle ABX as shown in figure C1. Since $\sin(\pi - \theta) = \sin \theta$, then starting from the sine rule:

$$\frac{\sin(\arccos(\lambda_A \cdot \mathbf{l}_{AB}))}{|\mathbf{R}_B|} = \frac{\sin(\arccos(\lambda_B \cdot \mathbf{l}_{AB}))}{|\mathbf{R}_A|}, \quad (\text{C.5})$$

$$\rightarrow \frac{\sin^2(\arccos(\lambda_A \cdot \mathbf{l}_{AB}))}{|\mathbf{R}_B|^2} = \frac{\sin^2(\arccos(\lambda_B \cdot \mathbf{l}_{AB}))}{|\mathbf{R}_A|^2}, \quad (\text{C.6})$$

$$\rightarrow \frac{|\mathbf{R}_A|}{|\mathbf{R}_B|} (1 - (\lambda_A \cdot \mathbf{l}_{AB})^2) = \frac{|\mathbf{R}_B|}{|\mathbf{R}_A|} (1 - (\lambda_B \cdot \mathbf{l}_{AB})^2), \quad (\text{C.7})$$

$$\rightarrow \frac{|\mathbf{R}_A|}{|\mathbf{R}_B|} \frac{1 - \lambda_A \cdot \mathbf{l}_{AB}}{1 - \lambda_B \cdot \mathbf{l}_{AB}} = \frac{|\mathbf{R}_B|}{|\mathbf{R}_A|} \frac{1 + \lambda_B \cdot \mathbf{l}_{AB}}{1 + \lambda_A \cdot \mathbf{l}_{AB}}, \quad (\text{C.8})$$

$$\therefore \mathbf{f}_{BA} = -\mathbf{f}_{AB}, \quad (\text{C.9})$$

where equation (C.7) is obtained using $\sin^2 \theta = 1 - \cos^2 \theta$. Similarly, for \mathbf{g}_{AB} :

$$\mathbf{g}_{BA} = [\mathbf{b} \cdot (\lambda_B \times \lambda_A)] \frac{\lambda_B + \lambda_A}{1 + \lambda_B \cdot \lambda_A} \quad (\text{C.10})$$

$$= -[\mathbf{b} \cdot (\lambda_A \times \lambda_B)] \frac{\lambda_A + \lambda_B}{1 + \lambda_A \cdot \lambda_B} \quad (\text{C.11})$$

$$= -\mathbf{g}_{AB}. \quad (\text{C.12})$$

Therefore if we define

$$\mathbf{F}_{AB} = -\frac{1 - 2\nu}{8\pi(1 - \nu)} \mathbf{f}_{AB} + \frac{1}{8\pi(1 - \nu)} \mathbf{g}_{AB} = -\mathbf{F}_{BA} \quad (\text{C.13})$$

then a triangular loop ABC connected to closure point P will have terms

$$\mathbf{F}_{AB} + \mathbf{F}_{BP} + \mathbf{F}_{PA} + \mathbf{F}_{BC} + \mathbf{F}_{CP} + \mathbf{F}_{PB} + \mathbf{F}_{CA} + \mathbf{F}_{AP} + \mathbf{F}_{PC} \quad (\text{C.14})$$

$$= \mathbf{F}_{AB} + \mathbf{F}_{BP} - \mathbf{F}_{AP} + \mathbf{F}_{BC} + \mathbf{F}_{CP} - \mathbf{F}_{BP} + \mathbf{F}_{CA} + \mathbf{F}_{AP} - \mathbf{F}_{CP} \quad (\text{C.15})$$

$$= \mathbf{F}_{AB} + \mathbf{F}_{BC} + \mathbf{F}_{CA} \quad (\text{C.16})$$

so the contribution to \mathbf{u} from \mathbf{F} is independent of the closure point as the virtual segments going to and from P always cancel. We now demonstrate that the solid angle terms can also be calculated for an out of plane closure point.

The solid angle of a polygon as seen from a field point is the projection of the area of the polygon onto a unit sphere centred at the field point. Barnett [13] obtained the following closed form expression for the solid angle of a triangular loop ABC at a field point X :

$$\Omega_{ABC} = -4 \cdot \text{sgn}(\lambda_A \cdot \mathbf{n}) \arctan \left(\sqrt{\tan\left(\frac{s}{2}\right) \tan\left(\frac{s-a}{2}\right) \tan\left(\frac{s-b}{2}\right) \tan\left(\frac{s-c}{2}\right)} \right), \quad (\text{C.17})$$

$$\mathbf{R}_A = \mathbf{A} - \mathbf{X}, \quad (\text{C.18})$$

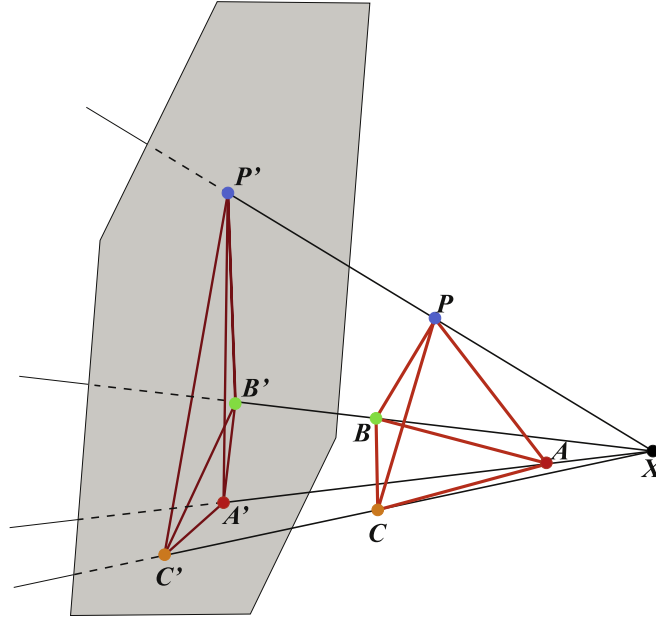


Figure C2. A projection of tetrahedron $ABCP$ onto a plane from field point X . The \mathbf{R} vectors go from X to their respective nodes. It can be seen that there are an infinite number of parallel planes onto which this projection can be made since the only requirement is that the plane intercept the four \mathbf{R} vectors while not being in between the field point X and the tetrahedron.

$$\lambda_A = \frac{\mathbf{R}_A}{|\mathbf{R}_A|}, \quad (\text{C.19})$$

$$\begin{aligned} \mathbf{n} &= (\mathbf{B} - \mathbf{A}) \times (\mathbf{C} - \mathbf{B}) = (\mathbf{B} - \mathbf{X} - (\mathbf{A} - \mathbf{X})) \times (\mathbf{C} - \mathbf{X} - (\mathbf{B} - \mathbf{X})) \\ &= (\mathbf{R}_B - \mathbf{R}_A) \times (\mathbf{R}_C - \mathbf{R}_B) \\ &= (\mathbf{R}_A \times \mathbf{R}_B) + (\mathbf{R}_B \times \mathbf{R}_C) + (\mathbf{R}_C \times \mathbf{R}_A), \end{aligned} \quad (\text{C.20})$$

$$\begin{aligned} a &= \arccos(\lambda_B \cdot \lambda_C), \quad b = \arccos(\lambda_A \cdot \lambda_C), \quad c = \arccos(\lambda_A \cdot \lambda_B) \\ s &= \frac{a + b + c}{2}. \end{aligned} \quad (\text{C.21})$$

In these expressions, \mathbf{n} is normal to the plane containing ABC , \mathbf{R}_A is the vector from the field point X to A , and so on for \mathbf{R}_B and \mathbf{R}_C , and λ_A , λ_B and λ_C are their respective normalised vectors. The angles between these vectors are a , b and c . Therefore using equations (C.19) and (C.20)

$$\lambda_A \cdot \mathbf{n} = \lambda_A \cdot (\mathbf{R}_B \times \mathbf{R}_C). \quad (\text{C.22})$$

Now consider a dislocation loop ABC being projected away from X onto an arbitrary plane, figure C2, to form a new triangle $A'B'C'$, such that $\mathbf{A}' = \mathbf{X} + m_A \mathbf{R}_A$, $\mathbf{B}' = \mathbf{X} + m_B \mathbf{R}_B$, $\mathbf{C}' = \mathbf{X} + m_C \mathbf{R}_C$, where m_A , m_B and m_C are positive scalars:

$$\mathbf{R}_{A'} = \mathbf{A}' - \mathbf{X} = \mathbf{X} + m_A \mathbf{R}_A - \mathbf{X} = m_A \mathbf{R}_A, \quad (\text{C.23})$$

$$\lambda_{A'} = \frac{\mathbf{R}_{A'}}{|\mathbf{R}_{A'}|} = \frac{m_A \mathbf{R}_A}{|m_A \mathbf{R}_A|} = \text{sgn}(m_A) \frac{\mathbf{R}_A}{|\mathbf{R}_A|} = \text{sgn}(m_A) \lambda_A = \lambda_A, \quad (\text{C.24})$$

$$\mathbf{n}' = m_A m_B (\mathbf{R}_A \times \mathbf{R}_B) + m_B m_C (\mathbf{R}_B \times \mathbf{R}_C) + m_C m_A (\mathbf{R}_C \times \mathbf{R}_A). \quad (\text{C.25})$$

Therefore using equations (C.24) and (C.25)

$$\lambda_{A'} \cdot \mathbf{n}' = \lambda_A \cdot \mathbf{n}' = m_B m_C \lambda_A \cdot (\mathbf{R}_B \times \mathbf{R}_C) = m_B m_C \lambda_A \cdot \mathbf{n}, \quad (\text{C.26})$$

$$\therefore \text{sgn}(\lambda_A \cdot \mathbf{n}) = \text{sgn}(\lambda_{A'} \cdot \mathbf{n}'), \quad (\text{C.27})$$

$$\therefore \Omega_{A'B'C'} = \Omega_{ABC}, \quad (\text{C.28})$$

where (C.28) follows from (C.27) and (C.17) and the fact the the angles a , b , c and s have not changed during the projection. It can also be seen that the sign of the solid angle changes when the sense of the loop changes since $\lambda_A \cdot (\mathbf{R}_C \times \mathbf{R}_B) = -\lambda_A \cdot (\mathbf{R}_B \times \mathbf{R}_C)$, therefore

$$\Omega_{ACB} = \Omega_{CBA} = \Omega_{BAC} = -\Omega_{ABC} = -\Omega_{CAB} = -\Omega_{BCA}. \quad (\text{C.29})$$

If ABC is connected to closure point P , we can form three new triangles: ABP , BCP and CAP which form a tetrahedron $ABCP$. For a given field point X , we can choose a plane on which to project this tetrahedron as shown in figure C2. If we define the shortest vector from X to the plane as N , if $\lambda \cdot N = 0$ then λ is parallel to the plane, and if $\lambda \cdot N < 0$ then λ is pointing away from the plane, in either case λ will not intercept the plane. Therefore provided:

$$\lambda_A \cdot N > 0 \quad \text{and} \quad \lambda_B \cdot N > 0 \quad \text{and} \quad \lambda_C \cdot N > 0 \quad \text{and} \quad \lambda_P \cdot N > 0 \quad (\text{C.30})$$

m_A , m_B , m_C and m_P are all positive. The solid angle calculation can now be done in plane for triangles $A'B'P'$, $B'C'P'$ and $C'A'P'$. Fivel and Depres demonstrated [15] that, so long as the sense of the loop is maintained, breaking an arbitrary planar loop up into triangles in plane using a common closure point and summing their respective solid angles yields the equivalent solid angle as the original loop. Therefore:

$$\Omega_{A'B'P'} + \Omega_{B'C'P'} + \Omega_{C'A'P'} = \Omega_{A'B'C'}. \quad (\text{C.31})$$

However, from equation (C.28), the solid angle of a triangular loop is equal to the solid angle of one face of a tetrahedron. This argument can be extended to all four faces such that:

$$\Omega_{ABC} = \Omega_{A'B'C'}, \quad \Omega_{A'B'P'} = \Omega_{ABP}, \quad \Omega_{B'C'P'} = \Omega_{BCP}, \quad \Omega_{C'A'P'} = \Omega_{CAP}. \quad (\text{C.32})$$

Therefore:

$$\Omega_{ABC} = \Omega_{ABP} + \Omega_{BCP} + \Omega_{CAP}. \quad (\text{C.33})$$

It is always possible to select a plane for which condition (C.30) is satisfied provided X is outside the tetrahedron formed by A , B , C and P . If X is inside this tetrahedron then no plane selected can satisfy condition (C.30) and so using this method calculates the wrong solid angle. This demonstrates that if the closure point P is chosen to be anywhere inside a finite volume V then \mathbf{u} arising from a triangular loop ABC can be evaluated on the surface of V provided ABC is inside V .

This method can be extended beyond triangular loops. Consider a square planar loop $ABCD$ broken up into four triangles in the habit plane of the loop using point Q as shown in figure C3(a). The displacement of these triangles are then calculated using an out of plane arbitrary closure point P , shown in figure C3(b). This generates the following \mathbf{F} terms:

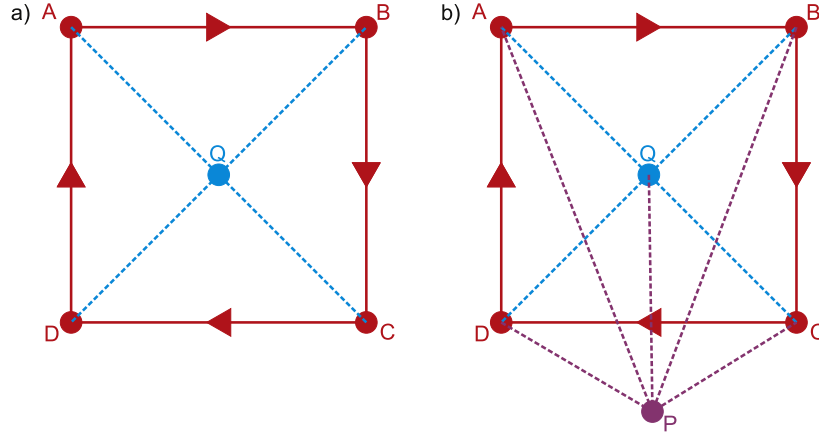


Figure C3. (a) Square loop $ABCD$ can be divided into four triangles using Q , ABQ , BCQ , CDQ and DAQ , for which each of the displacements can be calculated to give the total displacement of the whole loop when summed. (b) The loop can further be divided using out of plane closure point P , which renders the use of Q redundant. This approach can be used for any polygonal loop.

$$\begin{aligned}
 &F_{AB} + F_{BQ} + F_{QA} + F_{BC} + F_{CQ} + F_{QB} + F_{CD} + F_{DQ} + F_{QC} + F_{DA} + F_{AQ} + F_{QD} \\
 &= F_{AB} + F_{BP} + F_{PA} + F_{BC} + F_{CP} + F_{PB} + F_{CD} + F_{DP} + F_{PC} + F_{DA} + F_{AP} + F_{PD} \\
 &= F_{AB} + F_{BC} + F_{CD} + F_{DA}, \tag{C.34}
 \end{aligned}$$

$$\begin{aligned}
 \Omega_{ABCD} &= \Omega_{ABQ} + \Omega_{BCQ} + \Omega_{CDQ} + \Omega_{DAQ} \\
 &= \Omega_{ABP} + \Omega_{AQP} + \Omega_{QBP} + \Omega_{BCP} + \Omega_{BQP} + \Omega_{QCP} \\
 &\quad + \Omega_{CDP} + \Omega_{CQP} + \Omega_{QDP} + \Omega_{DAP} + \Omega_{DQP} + \Omega_{QAP}. \tag{C.35}
 \end{aligned}$$

Changing the sense of the loop changes the sign of the solid angle, equation (C.29). So eight terms in equation (C.35) cancel to give:

$$\Omega_{ABCD} = \Omega_{ABP} + \Omega_{BCP} + \Omega_{CDP} + \Omega_{DAP}. \tag{C.36}$$

This demonstrates that breaking the loop up in plane using Q is unnecessary and P can be directly used to calculate the Ω_{ABCD} term. This argument can also be extending for a many sided planar polygonal loop.

Appendix D. Colliding and non-planar loops

As \mathbf{b} only appears in the prefactor of f_{AB} and \mathbf{g}_{AB} , reversing the Burgers vector also reverses the resultant displacement vector. Therefore, simultaneously reversing the line direction and Burgers vector has no effect on the resultant displacement vector, as expected. The second identity required is that two dislocation loops occupying the same space but with different Burgers vectors, \mathbf{b}_1 and \mathbf{b}_2 , is equivalent to one loop with a combined burgers vector of $\mathbf{b}_3 = \mathbf{b}_1 + \mathbf{b}_2$. Again this follows as \mathbf{b} appears in the prefactor of every term in equation (C.1).

Now we can consider two colliding triangular loops, ABC and DBC . They are on separate (arbitrary) planes. The solid angle terms are calculated using arbitrary closure point P . If we define

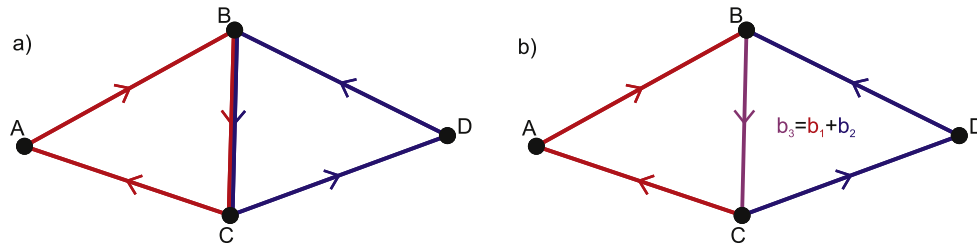


Figure D1. Two triangular loops, ABC and DBC , with Burgers vectors of b_1 and b_2 respectively, can be thought of as (a) two separate loops or (b) a junction when calculating the displacements they cause. The Burgers vector of the merged segment in (b) is the sum of the two separate segments in (a), b_3 .

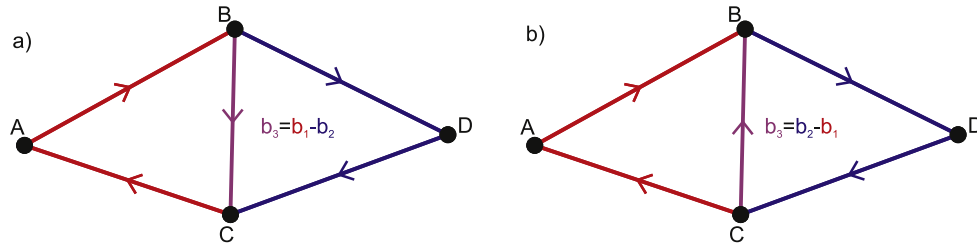


Figure D2. Two loops, ABC and DCB , have collided. There are two possible junctions that can form, (a) or (b). Conservation of Burgers vector means that BC in (a) has both opposite line direction and burgers vector to CB in (b). This results in the same displacement being calculated for both cases.

$$\omega_{ABC} = -\frac{\Omega_{ABC}}{4\pi} \quad (D.1)$$

and consider two separate loops just prior to the formation of a junction, as shown in figure D1(a), then for any field point outside of the polyhedron formed by $ABCDP$, the displacement is:

$$\begin{aligned} u(X) &= b_1 \omega_{ABC} + b_2 \omega_{DBC} \\ &\quad + F_{AB}(b_1) + F_{BC}(b_1) + F_{CA}(b_1) + F_{DB}(b_2) + F_{BC}(b_2) + F_{CD}(b_2) \\ &= b_1 \omega_{ABP} + b_1 \omega_{BCP} + b_1 \omega_{CAP} + b_2 \omega_{DBP} + b_2 \omega_{BCP} + b_2 \omega_{CDP} \end{aligned} \quad (D.2)$$

$$+ F_{AB}(b_1) + F_{BC}(b_1) + F_{CA}(b_1) + F_{DB}(b_2) + F_{BC}(b_2) + F_{CD}(b_2) \quad (D.3)$$

$$\begin{aligned} &= b_1 \omega_{ABP} + b_1 \omega_{CAP} + b_2 \omega_{DBP} + b_2 \omega_{CDP} + b_3 \omega_{BCP} \\ &\quad + F_{AB}(b_1) + F_{CA}(b_1) + F_{DB}(b_2) + F_{CD}(b_2) + F_{BC}(b_3). \end{aligned} \quad (D.4)$$

It can now be seen that the displacement vector generated by the two loops treated separately is identical to when they are treated as the junction shown in figure D1(b).

Conservation of Burgers vectors means that the sum of the Burgers vectors entering a node in a dislocation network must equal the sum leaving. In the example above, b_1 and b_2 enter and b_3 leaves node B. Thus for the conservation of Burgers vector we must have $b_1 + b_2 = b_3$. Similarly at point C, BC splits into CA and CD resulting in the same sum. This is consistent with how b_3 is calculated when the junction is formed. However, if we change

the line direction of one of the colliding loops, the sum changes. Loop ABC colliding with loop DCB instead of DBC , as shown in figure D2. If the junction forms such that at point B line AB enters and lines BD and BC leave, as shown in figure D2(a), then the Burgers vector sum is $\mathbf{b}_1 = \mathbf{b}_2 + \mathbf{b}_3$. Nodal based DD codes such as *ParaDiS* and *DDLab* have been written to account for this conservation of Burgers vector. This is convenient when the displacement is calculated because if two segments have opposite line directions and Burgers vectors then they give identical displacement contributions when using a common closure point. This means that segment BC with a Burgers vector of $-\mathbf{b}_2$ is indistinguishable from segment CB with a Burgers vector of \mathbf{b}_2 . This means that this example combines in the same way as the example in figure D1 to give the correct displacement. By extension, the same can be said of figure D2(b) where the junction is formed with the other possible configuration. Additionally, this can be used as a proof for the case for a non-planar loop, which can be thought of as a special condition of the collided loop, where $\mathbf{b}_1 = \mathbf{b}_2$ and $\mathbf{b}_3 = [000]$.

ORCID iDs

E Tarleton  <https://orcid.org/0000-0001-6725-9373>

References

- [1] Arsenlis A, Cai W, Tang M, Rhee M, Oppelstrup T, Hommes G, Pierce T G and Bulatov V V 2007 *Modelling Simul. Mater. Sci. Eng.* **15** 553–95
- [2] Giessen E V D and Needleman A 1995 *Modelling Simul. Mater. Sci. Eng.* **3** 689–735
- [3] Vattré A, Devincere B, Feyel F, Gatti R, Groh S, Jamond O and Roos A 2014 *J. Mech. Phys. Solids* **63** 491–505
- [4] Verdier M, Fivel M and Groma I 1998 *Modelling Simul. Mater. Sci. Eng.* **6** 755–70
- [5] Fivel M C and Canova G R 1999 *Modelling Simul. Mater. Sci. Eng.* **7** 753–68
- [6] Weygand D, Friedman L H, Giessen E V D and Needleman A 2002 *Modelling Simul. Mater. Sci. Eng.* **10** 437–68
- [7] Fivel M and Ei-Azab A 1999 Linking continuum mechanics and 3d discrete dislocation simulations *J. De Physique* **9** Pr9-261–270
- [8] Hirth J P and Lothe J 1982 *Theory of Dislocations* (Cambridge: Cambridge University Press)
- [9] Cai W, Arsenlis A, Weinberger C R and Bulatov V V 2006 *J. Mech. Phys. Solids* **54** 561–87
- [10] Queyreau S, Marian J, Wirth B D and Arsenlis A 2014 *Modelling Simul. Mater. Sci. Eng.* **22** 35004
- [11] de Wi R 1960 *Solid State Phys.—Adv. Res. Appl.* **10** 249–92
- [12] Hussein A M and El-Awady J A 2016 *Int. J. Fatigue* **87** 339–50
- [13] Barnett D M 1985 *Phil. Mag. A* **51** 383–7
- [14] Barnett D M and Balluffi R W 2007 *Phil. Mag. Lett.* **87** 943–4
- [15] Fivel M and Depres C 2014 *Phil. Mag.* **94** 3206–14
- [16] Po G, Mohamed M S, Crosby T, Erel C, El-Azab A and Ghoniem N 2014 *JOM* **66** 2108–20
- [17] Ferroni F, Tarleton E and Fitzgerald S 2014 *J. Comput. Phys.* **272** 619–28