Dislocation Analysis Tool for Atomistic Simulations

72

Alexander Stukowski

Contents

Abstract

Precise analysis and meaningful visualization of dislocation structures in molecular dynamics simulations are important steps toward physical insights. This chapter provides an introduction to the dislocation extraction algorithm (DXA), which is a computational method for identifying and quantifying dislocations in atomistic crystal models. It builds a bridge between the atomistic world of crystal defects and the discrete line picture of classical dislocation theory.

1 Introduction

Dislocations have two sides: On one hand, they are commonly viewed in dislocation theory as discrete line objects, which possess a characteristic topological charge – the *Burgers vector* – and which can glide through a crystal to produce plasticity or participate in various kinds of reactions. On the other hand, they constitute a partic-

A. Stukowski (✉)
Materials Modeling Division, Institute of Materials Science, Technische Universität Darmstadt, Darmstadt, Germany
e-mail: stukowski@mm.tu-darmstadt.de

W. Andreoni, S. Yip (eds.), *Handbook of Materials Modeling*,
https://doi.org/10.1007/978-3-319-44677-6_20

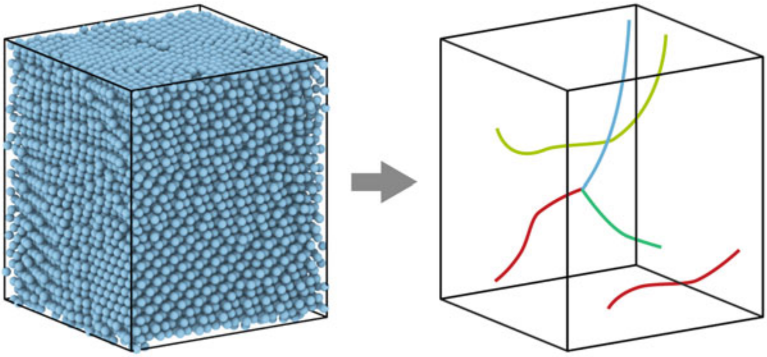

Fig. 1 In this section a computer algorithm is introduced that can convert an atomistic crystal model (left) to a discrete line representation of the contained dislocation defects (right)

ular kind of irregularity in the otherwise regular arrangement of atoms in the crystal lattice. Boundaries of extra planes of atoms are what we call edge dislocations, and screw dislocations denote helical distortions of atomic planes. Thus, to fully grasp the phenomenon of crystal dislocations, we must consider – and somehow unify – two complementary pictures: the *explicit* one (discrete line objects) and the *implicit* one (glitches in the lattice arrangement). Both pictures are extensively used by different modeling strategies, with discrete dislocation dynamics (DDD) and molecular dynamics (MD) being the most prominent ones to represent the two dislocation descriptions. This section will introduce ways to unify these seemingly disparate pictures of the same physical phenomenon and to accomplish a conversion between them (Fig. 1). Given the atomic positions in a crystal containing dislocation defects, we want to reconstruct the geometry of the one-dimensional lines these dislocations can be described as mathematically. The computational method introduced here to solve this problem has practical relevance for the modelling of dislocations at multiple length scales as it builds a bridge between the atomistic world and the mesoscale and, at the same time, provides a powerful analysis tool for MD simulations that greatly helps to understand dislocation processes.

2 Burgers Circuit Method

As an introductory example for the connection between dislocation line theory and atomistic crystal defects, we take a look at the classical Burgers circuit construction (Frank 1951), which is the canonical method (Bulatov and Cai 2006) already proposed in the 1950s to discriminate dislocations from other crystal defects and to determine their Burgers vectors. In the formulation employed here, a Burgers circuit C is a path in the dislocated crystal consisting of a sequence of atom-to-atom steps (line elements $\Delta\mathbf{x}$), as shown in Fig. 2a. The circuit is closed, thus $\sum_C \Delta\mathbf{x} = \mathbf{0}$.

We assume that there exists a mapping $\Delta\mathbf{x} \rightarrow \Delta\mathbf{x}'$ that translates each line element of the path to a corresponding image, $\Delta\mathbf{x}'$, in a perfect crystal

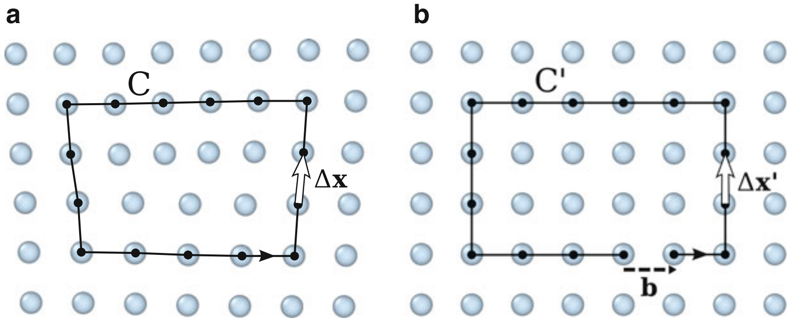

Fig. 2 Burgers circuit method to identify a dislocation. A closed circuit around the dislocation is translated from (**a**) the dislocated crystal to (**b**) the perfect reference crystal. The closure failure **b** is called the Burgers vector of the dislocation

lattice (Fig. 2b). Summing these transformed line elements algebraically along the associated path, C', gives the true Burgers vector of the dislocation enclosed by C:

$$\mathbf{b} = -\sum_{C'} \Delta\mathbf{x}'. \tag{1}$$

The Burgers vector **b** is the closure failure of the path after transferring it to the perfect reference crystal.

Note that the Burgers circuit procedure is typically performed by hand to analyze two-dimensional crystal images obtained from high-resolution microscopy or atomistic computer simulations. Human intuition and cognitive capabilities are required to spot irregularities in the crystal lattice that are potential dislocation defects and to apply the Burgers circuit test to them. Automating these steps poses a challenge when developing a computational dislocation identification method for three-dimensional atomistic crystal models.

The Burgers circuit procedure represented by Eq. 1 above is the discrete analogue of an equation used in continuum mechanics to define the Burgers vector of a Volterra dislocation:

$$\mathbf{b} = -\int_C (\mathbf{F}^{\mathbf{e}})^{-1} \mathrm{d}\mathbf{x}. \tag{2}$$

Here, C denotes any contour enclosing the mathematical dislocation line, and $(\mathbf{F}^{\mathbf{e}})^{-1}$ denotes the inverse of the elastic deformation gradient. This second-rank tensor acts on the infinitesimal line element d**x** and transforms it from the dislocated crystal configuration to an ideal, elastically unstrained reference configuration. This mapping is analogous to the explicit translation of atomic steps we did in the discrete formulation of the Burgers circuit procedure.

Notably, the resulting vector **b** stays the same if we change the original circuit C, as long as it still encloses the same dislocation. On the other hand, if $\mathbf{b} = \mathbf{0}$, we

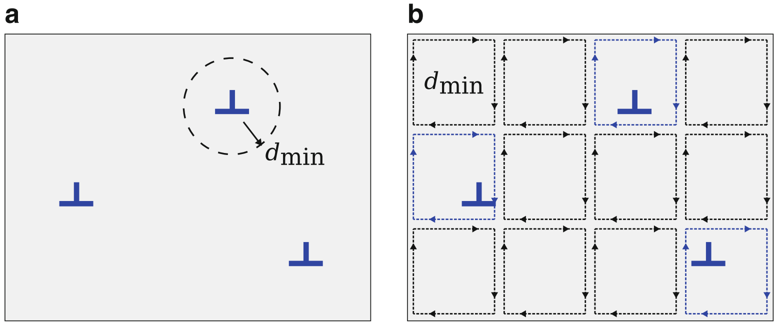

Fig. 3 Schematic depiction of the dislocation finding approach described in the text. (**a**) The given domain contains a set of dislocations with unknown positions. The parameter d_{min} denotes the lower bound for the separation distance between dislocations, which is on the order of one atomic lattice spacing and corresponds to the dislocation core diameter. (**b**) The domain is tessellated by a grid of Burgers circuits of diameter d_{min}. Circuits highlighted in blue exhibit a closure failure and are marked as containing a dislocation

know that the Burgers circuit did not enclose any defect with dislocation character. Here, however, we are deliberately ignoring the possibility that the circuit encloses multiple dislocations whose Burgers vectors cancel. One may thus ask, in the absence of a priori knowledge of the spatial distribution of dislocations in a given crystal, how can we – or rather a computer algorithm – construct the circuit C such that it encloses exactly one of the dislocations?

This general situation is depicted schematically in Fig. 3a: A set of dislocations with unknown positions which we would like to determine is distributed across a given continuum domain. It is safe to assume that any distribution of dislocations is such that one can specify a lower-bound d_{min} for the separation distance between any two distinct dislocations. In reality, this lower bound is given by the interatomic spacing in the crystal lattice, because that is also the minimum distance when the cores of two nearby dislocations necessarily start to overlap; hence they can no longer be treated as distinct defects. As shown in Fig. 3b, it is possible to construct a large number of non-overlapping circuits, each having size d_{min}, to completely cover the entire domain. By virtue of our construction, each circuit can contain at most one dislocation, and we effectively excluded the possibility of "missing" dislocations. The Burgers circuit test tells us which of the circuits contain a dislocation, and since their diameters are small, we can pinpoint the dislocations' positions with great precision (on the order of d_{min}) using this method.

3 Simple Algorithm for Finding Dislocations in Atomistic Crystals

The approach outlined above can be translated into a simple computer algorithm to detect and find all dislocations in an atomistic crystal (Stukowski 2014). We

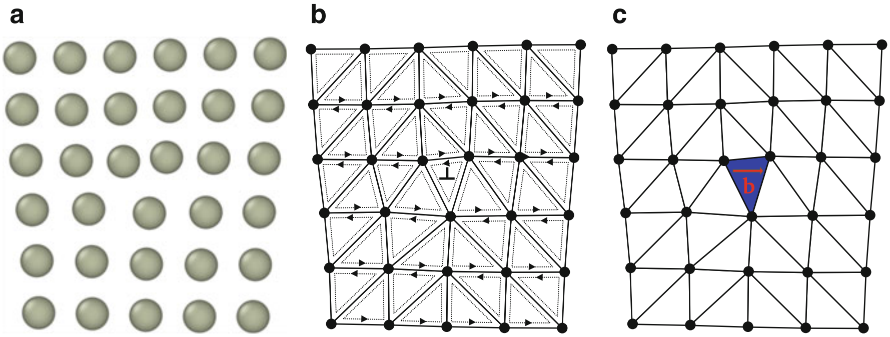

Fig. 4 (**a**) Input atomic positions. (**b**) Tessellation of the input domain into triangular circuits using the Delaunay construction. (**c**) Burgers circuit test, performed on each Delaunay triangle, reveals exactly one cell that contains the dislocation

assume that the crystal to be analyzed is specified as a set of atomic coordinates $\{\mathbf{x}_i\}$ as depicted in Fig. 4a. The Delaunay construction is used to tessellate the crystal domain into a set of triangles (Fig. 4b). The elements of the triangulation are space-filling and non-overlapping, and we can regard them as small, elementary Burgers circuits, which will allow us to find and locate all dislocations contained in the atomistic crystal.

Since every edge of a triangle *abc* of the Delaunay tessellation represents an atom-to-atom step, we can apply the discrete version of the Burgers circuit method to calculate the per-triangle closure failure $\mathbf{b}_{abc} = \Delta\mathbf{x}'_{ab} + \Delta\mathbf{x}'_{bc} + \Delta\mathbf{x}'_{ca}$ after mapping each edge vector $\Delta\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ connecting two successive atoms i and j to its corresponding ideal vector $\Delta\mathbf{x}'_{ij}$ in a perfect reference crystal lattice. Triangle circuits with closure failure $\mathbf{b}_{abc} \neq 0$ are marked as containing a dislocation, as shown in Fig. 4c.

There are different ways to accomplish the mapping of interatomic vectors from the dislocated crystal to the virtual reference lattice, $\Delta\mathbf{x}_{ij} \rightarrow \Delta\mathbf{x}'_{ij}$. In cases where the orientation of the dislocated crystal in the simulation coordinate system is known a priori, we can simply pick the vector $\Delta\mathbf{x}'_{ij}$ from a prescribed set of ideal lattice vectors taking the one that is closest to the elastically distorted vector $\Delta\mathbf{x}_{ij}$ (Stukowski 2014). In more general situations, a structure identification method such as common neighbor analysis (CNA) (Honeycutt and Andersen 1987; Faken and Jonsson 1994) or polyhedral template matching (PTM) (Larsen et al. 2016) must be used to first determine the local lattice orientation and then map atomic neighbor vectors to corresponding ideal lattice directions.

So far, we have considered only two-dimensional crystals where dislocations are point-like object in the plane. How does this approach extend to three-dimensional crystals containing linear dislocations? Here, the Delaunay tessellation of the atomistic model consists of tetrahedral cells, each being bordered by four triangular facets (Fig. 5). In the three-dimensional version of the algorithm, the closure failure $\mathbf{b}_{abc}$ must be computed for every triangular facet of the tetrahedral Delaunay cells.

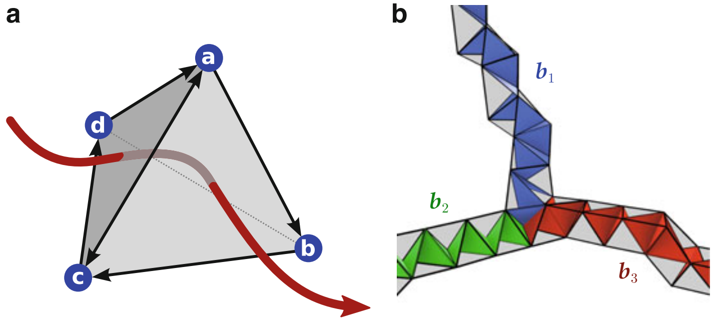

Fig. 5 (**a**) A tetrahedral cell of the three-dimensional Delaunay tessellation, which is spanned by four vertex atoms. A dislocation line enters and exits through the triangular facets of the cell. The algorithm described in the text identifies such facets using the Burgers circuit test. (**b**) Linear defects with dislocation character lead to "chains" of dislocated Delaunay cells, which may form junctions in three-dimensions as exemplarily shown here

If $\mathbf{b}_{abc} \neq 0$, a facet is marked as being intersected by a dislocation line. Since dislocations cannot end within an otherwise perfect crystal, because of the Burgers vector conservation law, a line entering a Delaunay cell through one of its triangular facets must exit the cell again through one of its other three facets. Accordingly, the dislocation can be viewed as a line piercing through a sequence of triangular facets and tetrahedral cells as illustrated by Fig. 5b.

4 Dislocation Extraction Algorithm (DXA)

So far we have deliberately ignored several important aspects that can play a role in more general situations. First and foremost, crystal dislocations have a finite core size (Fig. 6a). That means they are not mathematically thin, one-dimensional objects but rather tubelike objects spread over a certain space region. The core region typically extends over more than one interatomic spacing and is thus covering more than one Delaunay triangle element. In order to capture such dislocations, larger Burgers circuits are necessary to fully enclose the core (which is represented by a connected set of Delaunay elements that have been marked as "bad"; see Fig. 6b).

Secondly, dislocations may dissociate into partial dislocations. If we want to identify partial dislocations individually, e.g., Shockley partials in fcc crystals, using the Burgers circuit procedure, we have to take special provisions, as the circuit enclosing the dislocation necessarily passes through the adjacent stacking fault defect (Fig. 6c). Only if we map the atomic step leading through that stacking fault plane to the correct fractional lattice vector, we will obtain the right (fractional) Burgers vector of the partial dislocation.

Finally, crystals often contain other defects in addition to dislocations. A general dislocation identification algorithm must therefore be able to deal with non-dislocation irregularities such as free surfaces, point defects, grain boundaries, other

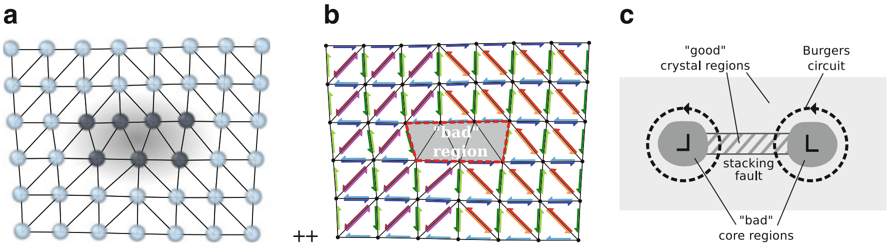

Fig. 6 (**a**) Dislocation with an extended core. Atoms that are part of the core (darker color) can be identified using a structural characterization technique such as common neighbor analysis. (**b**) Bad tessellation elements, for which no unambiguous mapping to the perfect reference lattice is possible, have been marked with a gray color. (**c**) Schematic depiction of a dissociated dislocation. Identification of the two partial dislocations requires Burgers circuits passing through the stacking fault

types of interfaces, and even disclinations in a robust way. And, as mentioned before, the crystal orientation and crystal structure may not be known in advance and can vary across space and time. The algorithm needs to adapt to these situations appropriately.

In order to address these challenges, a computer algorithm named the *dislocation extraction algorithm (DXA)* (Stukowski et al. 2012; Stukowski and Albe 2010) has been devised on the basis of the fundamental ideas described in the preceding sections. The DXA is capable of building a discretized line representation of all dislocations contained in a given atomistic crystal model (Fig. 1). The generated representation of dislocation lines found in the crystal is very similar to those employed by dislocation dynamics simulation models. The DXA is available as part of the OVITO (Stukowski 2010) data analysis and visualization software for atomistic simulations.

The DXA proceeds in several steps, starting with the atomic input coordinates, to arrive at the final line representation of the dislocations. Here is a synopsis of these processing steps, which are described in more detail in Stukowski et al. (2012):

1. The three-dimensional Delaunay tessellation is computed.
2. Atoms in the input crystal are identified that form a perfect crystal lattice. For this, the common neighbor analysis (CNA) method is used, which allows to identify common lattice types such as fcc, bcc, hcp, and diamond. The information is also used to determine local lattice orientations and map atom-to-atom vectors in the Delaunay tessellation to the ad hoc reference lattice.
3. Elements in the Delaunay tessellation are flagged as "bad" crystal regions if they contain disordered atomic arrangements. This includes the dislocation cores, where the atomic structure deviates considerably from one of the perfect crystals, but also other types of defects (Fig. 6b).
4. The separating surface between the "good" and the "bad" crystal regions in the Delaunay tessellation, the so-called interface mesh, is generated (see Fig. 7).

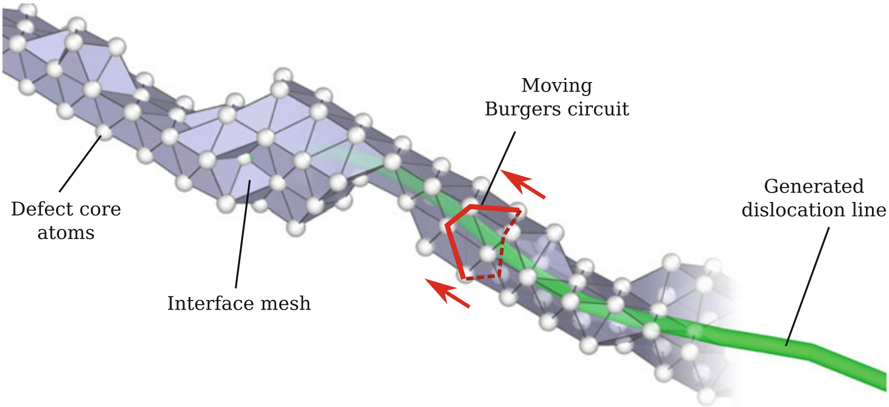

Fig. 7 Illustration of the interface mesh constructed by the DXA to enclose all defect atoms of a dislocation core. The algorithm uses an "elastic" Burgers circuit (red) that is moving on the interface mesh to sweep the dislocation line. While this circuit is being advanced in a stepwise fashion, triangle by triangle, a continuous line representation of the dislocation defect is produced (green)

5. The algorithm then generates a large number of trial circuits on the interface mesh until it encounters a first circuit that fully encloses a dislocation. This is detected by computing the Burgers sum (Eq. 1). The maximum size of the trial circuits is bounded by a user parameter controlling how wide dislocation cores may be for the algorithm to detect them.
6. The first circuit is subsequently used to discover the rest of the current dislocation line. This happens by advancing the circuit on the interface mesh and sweeping along the dislocation line as depicted in Fig. 7.
7. During this sweeping phase, a one-dimensional line representation of the dislocation is generated by computing the new center of mass of the circuit every time it is advanced along the boundary of the dislocation core. Here, a circuit can be pictured as a rubber band tightly wrapped around the dislocation's core. As the circuit moves along the dislocation segment, it may need to locally expand to sweep over wider sections of the core, e.g., kinks or jogs. To prevent the circuit from sweeping past dislocation junctions or interfaces, again a limit is imposed on the circuit length.
8. As a last step, a post-processing of the discretized dislocation lines is performed to reduce the number of sampling points.

The sweeping of dislocation lines, performed in steps 6 and 7 of the algorithm, in fact happens simultaneously on all segments of a dislocation network as depicted in Fig. 8. The initial seed circuits, constructed at the slimmest spots of the dislocation segments, split into pairs of circuits, each sweeping along the cores' surfaces in opposite directions. During this sweeping process, the upper limit for each circuit's maximum length is continuously raised, letting the circuits approach closer and closer to the dislocation junctions, which typically exhibit a wider cross section than

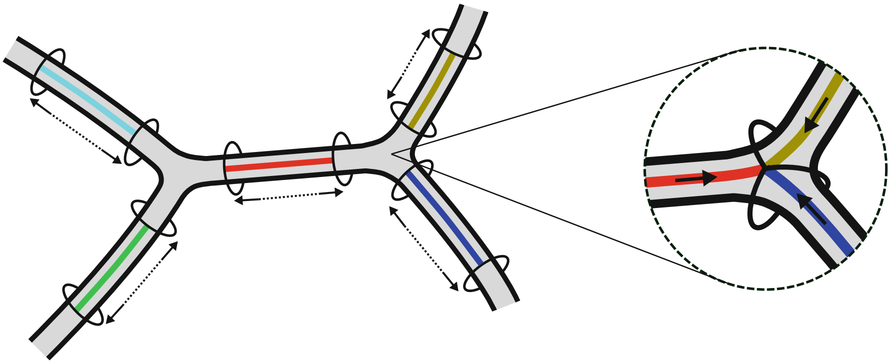

Fig. 8 Schematic depiction of the DXA line tracing process for a network of dislocations. All dislocation arms are simultaneously swept by pairs of Burgers circuits advancing on the core's surface in opposite directions (cf. Fig. 7). At junctions, the inbound circuits from different arms meet, and the algorithm outputs a nodal point to connect all lines traced by these circuits

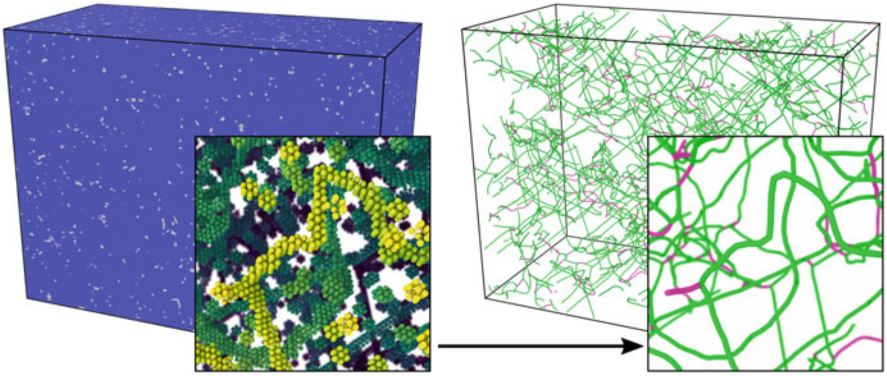

Fig. 9 Left: Molecular dynamics simulation of dislocation-based single crystal plasticity (Zepeda-Ruiz et al. 2017). The piece of tantalum crystal (33 million atoms) is being deformed under uniaxial compression. Common neighbor analysis filtering (Stukowski 2012) of the atomic coordinates reveals a high density of defects in the bcc lattice (inset). Right: After processing with the DXA, non-dislocation defects such as vacancies have been filtered out. The resulting line representation allows measuring dislocation densities and studying dislocation processes in great detail. Green and magenta lines represent $\frac{1}{2}\langle 111\rangle$ and $\langle 100\rangle$ dislocations, respectively

the dislocation arms. At some point, the converging circuits all meet in a junction, and the algorithm links up their corresponding line ends at a nodal point.

5 Use Cases of the DXA

The DXA can serve as a measurement tool to quantify the density of dislocations in molecular dynamics simulations, thanks to the conversion of the identified defects to a mathematical line representation (Fig. 9). The average dislocation density in a

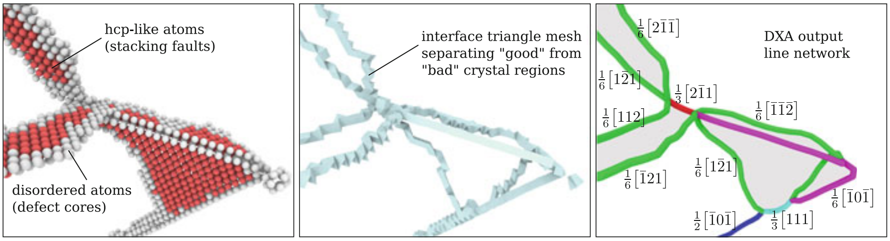

Fig. 10 Analysis of a complex dislocation junction in fcc Al that has formed through a reaction of two $\frac{1}{2}\langle 110\rangle$ dislocation loops on intersecting glide planes. Left: Atomic input configuration with applied CNA filtering to highlight stacking fault atoms (red) and defect core atoms (gray). Center: Visualization of the *interface mesh*, the intermediate structure constructed by the DXA to enclose the defect cores. Right: Output dislocation lines and Burgers vector labeling generated by the DXA

crystal, ρ, is simply computed from the integral line length divided by the simulation box volume (see, e.g., Zepeda-Ruiz et al. 2017). The generated description of the dislocation geometry may also be used, for instance, to measure the size of the plastic zone in nanoindentation simulations (Gao et al. 2015; Voyiadjis and Yaghoobi 2015; Remington et al. 2014; Yaghoobi and Voyiadjis 2016; Alhafez et al. 2016; Alabd Alhafez et al. 2017) or to obtain local dislocation densities. Since the DXA does not only yield the shape but also the Burgers vector of each dislocation segment, the total (or statistical) density as well as the geometrically necessary density of dislocations (GND) are accessible via the DXA.

Another typical use case of the DXA is detailed analyses of dislocation reactions in molecular dynamics and statics simulations (e.g., Zhang et al. 2017). These reactions include the formation and nucleation of new dislocations at other defects, e.g., free surfaces (Trushin et al. 2016), grain boundaries (Stukowski et al. 2010), crack tips (Vatne et al. 2013), or pores (Ruestes et al. 2014) in a material. Here, the DXA can generate a precise labeling of dislocations forming complex network configurations as demonstrated in Fig. 10.

6 Current Limitations of the DXA

While the DXA represents a great improvement over conventional, atom-based analysis techniques such as the CNA, it still has certain limitations that one should be aware of when using this analysis tool. This section provides a roundup of key issues that need to be taken into account while working with the DXA and its output. Note, however, that the DXA is the subject of ongoing research seeking to improve the algorithm and overcome some of the issues mentioned here.

Accuracy and ambiguity of dislocation representations In general, given an atomistic crystal configuration, the representation of the contained defects in terms of a set of discrete dislocation lines is not uniquely defined. For instance, dislocations in fcc crystals typically dissociate into pairs of partial dislocations (see upper left corner in Fig. 10). If the separation distance between such two partials becomes very small (on the order of 1–2 interatomic spacings), it is up to the algorithm to decide whether this particular atomistic configuration is better represented by a single dislocation line or two partials. The algorithm's choice is based on its particular notion of the dislocation core (in DXA terminology termed the "bad" crystal region, following Frank 1951). If the two dislocation core regions overlap, the algorithm is only able to construct a single Burgers circuit enclosing both defects. As a result, the two dislocations are fused into a single discrete dislocation line in the DXA's output representation. On the other hand, if the two cores are separated by just some "good" crystal region in between, then the DXA generates Burgers circuits around each of the individual partials, and two separate lines will be generated in the output. Thus, some "good" crystal atoms are required in between the two dislocations to separate them, cf. Figs. 6c and 10. A second type of ambiguity arises for very short dislocation segments and at dislocation junctions. Figure 11 depicts a detail of a dislocation network where four arms merge into a junction (a "4-junction"). However, they do not meet exactly in one point, and due to this slight dissociation of the junction, there is freedom of whether to describe this configuration as two separate 3-junctions instead, which are connected by an additional short dislocation segment. Which topology the DXA prefers depends on minutiae of the core morphology at this junction. As the four inbound Burgers circuits approach the junction, the upper limit on the circuits' lengths is continuously raised, letting the circuits stretch and advance further into the junction step by step. Simultaneously, the algorithm tries to generate an additional seed circuit around the core of the dissociated junction, i.e., in the inner area that has not been swept by the existing circuits yet. If the algorithm succeeds in spawning another Burgers circuit for the connecting segment before the existing circuits have met in the junction, then a topology with two 3-junctions results. Otherwise, the algorithm yields a single 4-junction. Effectively, the outcome is determined by the ratio of the core diameter of the dissociation segment and its length.

Supported crystal structures The DXA relies on an ad hoc mapping being established between the dislocated crystal and a corresponding ideal reference crystal lattice. This mapping is accomplished by means of an atomic structure identification method (Stukowski 2012), which maps the nearest neighbor vectors of each identified input atom to corresponding ideal lattice directions. The current implementation of the DXA uses the common neighbor analysis (CNA) method (Honeycutt and Andersen 1987) as a subroutine for this step, and it is thus currently limited to bcc, fcc, and hcp crystals and, thanks to a recent extension of the CNA method (Maras et al. 2016), also cubic and hexagonal diamond structures. Since the chemical types of atoms are not relevant to the DXA itself, crystalline compounds with a sub-lattice matching one of these supported structure types can also be processed by feeding only atoms from a sub-lattice to the algorithm. It is expected that future implementations of the DXA will employ new structure identification techniques other than the CNA in order to support a wider range of crystal structures.

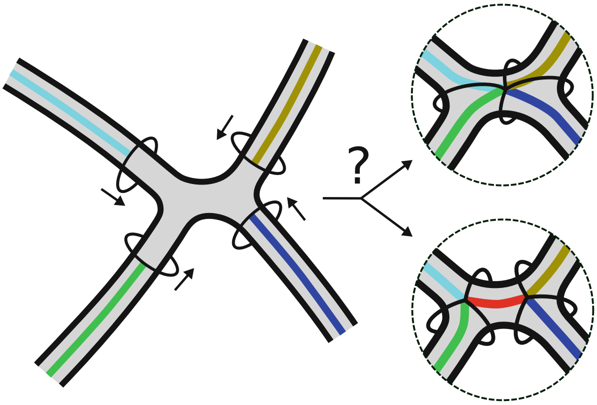

Fig. 11 At slightly dissociated dislocation junctions like the one shown here, an ambiguity arises of whether to connect all arms at a single node or instead create two nodes that are connected by a short extra segment

In principle, any computational method that establishes a local mapping between atomic neighbor vectors and the ideal reference lattice can serve as foundation for the dislocation detection part of the DXA.

Glide plane identification The DXA yields the geometric shape of a dislocation as well as its Burgers vector. This information alone, however, is not sufficient to identify the glide plane the dislocation is moving on. Thus, the DXA falls short of answering the question of active slip systems in a deforming crystal. In certain cases, however, one can use heuristic criteria to guess the glide planes of dislocations. For Shockley partials in fcc crystals, a dislocation's Burgers vector uniquely determines its glide plane, and no other information is needed. In other cases, if a dislocation has an edge component (is not pure screw), its glide plane can be determined from the Burgers vector and the line direction. In general, however, it is important to recognize that the glide plane of a dislocation is a dynamic property and requires the analysis of the dislocation's path it takes through the crystal, which is beyond the DXA's capabilities (see next item).

Tracking dislocations through time and space It is important to note that the DXA operates on instantaneous snapshots of an atomistic crystal and builds a line model of the dislocations at certain simulation times. In other words, it is a static analysis method, not a kinematic one. Since dislocations are not physical objects (see our introductory discussion), they do not possess unique identities that would allow to track them over time. This makes it difficult to automatically correlate successive snapshots of the evolving dislocation configuration as dislocations can move arbitrary distances between MD simulation snapshots, undergo reactions, and appear newly via nucleation and disappear via absorption or annihilation processes. In particular, the DXA cannot directly deliver dislocation velocity information, because tracking of dislocations would require additional heuristics to link dislocations in successive DXA snapshots.

References

Alabd Alhafez I, Ruestes CJ, Urbassek HM (2017) Size of the plastic zone produced by nanoscratching. Tribol Lett 66(1):20. https://doi.org/10.1007/s11249-017-0967-9

Alhafez IA, Ruestes CJ, Gao Y, Urbassek HM (2016) Nanoindentation of hcp metals: a comparative simulation study of the evolution of dislocation networks. Nanotechnology 27(4):045706. http://stacks.iop.org/0957-4484/27/i=4/a=045706

Bulatov VV, Cai W (2006) Computer simulations of dislocations. Oxford University Press, Oxford/New York

Faken D, Jonsson H (1994) Systematic analysis of local atomic structure combined with 3D computer graphics. Comput Mater Sci 2(2):279–286

Frank FC (1951) LXXXIII. Crystal dislocations – elementary concepts and definitions. Philos Mag Ser 7 42(331):809–819

Gao Y, Ruestes CJ, Tramontina DR, Urbassek HM (2015) Comparative simulation study of the structure of the plastic zone produced by nanoindentation. J Mech Phys Solids 75(Suppl C):58–75. https://doi.org/10.1016/j.jmps.2014.11.005, http://www.sciencedirect.com/science/article/pii/S00225096140%0221X

Honeycutt JD, Andersen HC (1987) Molecular dynamics study of melting and freezing of small Lennard-Jones clusters. J Phys Chem 91(19):4950–4963

Larsen PM, Schmidt S, Schiøtz J (2016) Robust structural identification via polyhedral template matching. Model Simul Mater Sci Eng 24(5):055007. http://stacks.iop.org/0965-0393/24/i=5/a=055007

Maras E, Trushin O, Stukowski A, Ala-Nissila T, Jónsson H (2016) Global transition path search for dislocation formation in Ge on Si(001). Comput Phys Commun 205(Suppl C):13–21. https://doi.org/10.1016/j.cpc.2016.04.001, http://www.sciencedirect.com/science/article/pii/S00104655163%00893

Remington T, Ruestes C, Bringa E, Remington B, Lu C, Kad B, Meyers M (2014) Plastic deformation in nanoindentation of tantalum: a new mechanism for prismatic loop formation. Acta Mater 78(Suppl C):378–393. https://doi.org/10.1016/j.actamat.2014.06.058, http://www.sciencedirect.com/science/article/pii/S13596454140%04881

Ruestes C, Bringa E, Stukowski A, Nieva JR, Tang Y, Meyers M (2014) Plastic deformation of a porous bcc metal containing nanometer sized voids. Comput Mater Sci 88(Suppl C):92–102. https://doi.org/10.1016/j.commatsci.2014.02.047, http://www.sciencedirect.com/science/article/pii/S09270256140%01517

Stukowski A (2010) Visualization and analysis of atomistic simulation data with OVITO – the open visualization tool. Model Simul Mater Sci Eng 18(1):015012. https://doi.org/10.1088/0965-0393/18/1/015012

Stukowski A (2012) Structure identification methods for atomistic simulations of crystalline materials. Model Simul Mater Sci Eng 20(4):045021. https://doi.org/10.1088/0965-0393/20/4/045021

Stukowski A (2014) A triangulation-based method to identify dislocations in atomistic models. J Mech Phys Solids 70:314–319. https://doi.org/10.1016/j.jmps.2014.06.009, http://www.sciencedirect.com/science/article/pii/S00225096140%01331

Stukowski A, Albe K (2010) Extracting dislocations and non-dislocation crystal defects from atomistic simulation data. Model Simul Mater Sci Eng 18(8):085001. https://doi.org/10.1088/0965-0393/18/8/085001

Stukowski A, Albe K, Farkas D (2010) Nanotwinned fcc metals: strengthening versus softening mechanisms. Phys Rev B 82:224103. https://doi.org/10.1103/PhysRevB.82.224103

Stukowski A, Bulatov V, Arsenlis A (2012) Automated identification and indexing of dislocations in crystal interfaces. Model Simul Mater Sci 20:085007

Trushin O, Maras E, Stukowski A, Granato E, Ying SC, Jónsson H, Ala-Nissila T (2016) Minimum energy path for the nucleation of misfit dislocations in Ge/Si(0 0 1) heteroepitaxy. Model Simul Mater Sci Eng 24(3):035007. http://stacks.iop.org/0965-0393/24/i=3/a=035007

Vatne IR, Stukowski A, Thaulow C, østby E, Marian J (2013) Three-dimensional crack initiation mechanisms in bcc-Fe under loading modes I, II and III. Mater Sci Eng A 560(Suppl C):306–314. https://doi.org/10.1016/j.msea.2012.09.071, http://www.sciencedirect.com/science/article/pii/S09215093120%13858

Voyiadjis GZ, Yaghoobi M (2015) Large scale atomistic simulation of size effects during nanoindentation: dislocation length and hardness. Mater Sci Eng A 634(Suppl C):20–31. https://doi.org/10.1016/j.msea.2015.03.024, http://www.sciencedirect.com/science/article/pii/S09215093150%02579

Yaghoobi M, Voyiadjis GZ (2016) Atomistic simulation of size effects in single-crystalline metals of confined volumes during nanoindentation. Comput Mater Sci 111(Suppl C):64–73. https://doi.org/10.1016/j.commatsci.2015.09.004, http://www.sciencedirect.com/science/article/pii/S09270256150%0573X

Zepeda-Ruiz LA, Stukowski A, Oppelstrup T, Bulatov VV (2017) Probing the limits of metal plasticity with molecular dynamics simulations. Nature 550:492

Zhang L, Lu C, Tieu K, Su L, Zhao X, Pei L (2017) Stacking fault tetrahedron induced plasticity in copper single crystal. Mater Sci Eng A 680(Suppl C):27–38. https://doi.org/10.1016/j.msea.2016.10.034, http://www.sciencedirect.com/science/article/pii/S09215093163%12424