Automated identification and indexing of dislocations in crystal interfaces

2012 Modelling Simul. Mater. Sci. Eng. 20 085007

(http://iopscience.iop.org/0965-0393/20/8/085007)

Modelling Simul. Mater. Sci. Eng. **20** (2012) 085007 (16pp) doi:10.1088/0965-0393/20/8/085007

Automated identification and indexing of dislocations in crystal interfaces

Alexander Stukowski, Vasily V Bulatov and Athanasios Arsenlis

Lawrence Livermore National Laboratory, Livermore, CA 94550, USA

E-mail: stukowski@mm.tu-darmstadt.de

Received 7 August 2012
Published 31 October 2012
Online at stacks.iop.org/MSMSE/20/085007

Abstract

We present a computational method for identifying partial and interfacial dislocations in atomistic models of crystals with defects. Our automated algorithm is based on a discrete Burgers circuit integral over the elastic displacement field and is not limited to specific lattices or dislocation types. Dislocations in grain boundaries and other interfaces are identified by mapping atomic bonds from the dislocated interface to an ideal template configuration of the coherent interface to reveal incompatible displacements induced by dislocations and to determine their Burgers vectors. In addition, the algorithm generates a continuous line representation of each dislocation segment in the crystal and also identifies dislocation junctions.

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

1. Introduction

Crystal interfaces play an important role in defining properties of crystalline materials. For instance, grain boundaries (GBs) can act as dislocation sources and sinks, or as barriers to dislocation motion, giving rise to Hall–Petch hardening. Absorption, emission and transmission of dislocations at GBs are important phenomena that can be studied by means of molecular dynamics (MD) and other atomistic simulation methods. The so-called *vicinal GBs* can be described in terms of secondary GB dislocations and are believed to be major sources of lattice dislocations in nanocrystalline and nanotwinned materials where conventional dislocation sources are suppressed. Furthermore, secondary GB dislocations can participate in stress-induced GB migration and deformation twinning.

Conventional structure identification methods for atomistic datasets [1] allow filtering out undisturbed crystal atoms to reveal defects for visualization purposes [2]. However, the existing methods do not provide a means to discriminate dislocations from non-dislocation defects, or to determine Burgers vectors of dislocations. While in most crystal lattices the number of observable dislocation types is small, the spectrum of dislocations in GBs can be much wider. With the Burgers circuit procedure, a theoretical method exists that—at least in principle—allows one to determine the Burgers vector of any dislocation. However, constructing Burgers

circuits manually is extremely laborious and error-prone. Furthermore, when it comes to analyzing time-resolved reactions, e.g. involving lattice and interfacial dislocations, the manual Burgers circuit method quickly becomes useless due to the complexity and rapid evolution of defect structures.

Recently, we developed a computational analysis method [3] to identify dislocations in atom-position datasets generated in atomistic simulations. This fully automated algorithm analyzes the atomic positions in the crystal and identifies all defects that are dislocations, determines their Burgers vectors and outputs a continuous line representation of the dislocation network. The original description of the dislocation extraction algorithm (DXA) was, however, limited to perfect lattice dislocations in only two simple lattices (fcc and bcc). In the following section we present a general formulation of the DXA algorithm that allows identifying perfect dislocations in arbitrary lattices. Then in section 3 we extend the DXA algorithm to the identification of partial dislocations and secondary dislocations in semicoherent crystal interfaces and GBs. Section 4 describes the application of our new general algorithm to the test case of lattice dislocations interacting with a GB. In section 5 we discuss applicability and limitations of our algorithm in the analysis of crystal defect structures and then summarize our results in section 6.

2. DXA for arbitrary lattices

In this section, we restate the DXA algorithm for the case of perfect dislocations in arbitrary crystal lattices and introduce a few important concepts to be used later. In the subsequent sections we will further generalize the approach to the analysis of dislocations in crystal interfaces.

2.1. The Burgers circuit

A dislocation can be identified by means of the Burgers circuit procedure suggested by Frank [4]. Given a closed path C (figure 1(a)), the Burgers vector $\boldsymbol{b}$ of the enclosed dislocation defect is defined as the line integral of the elastic displacement field [5]:

$$\boldsymbol{b} = \oint_C \frac{\partial \boldsymbol{U}^{\mathrm{e}}}{\partial l} \mathrm{d}l. \tag{1}$$

Note that the circuit C is specified in the dislocated crystal, i.e. in spatial coordinates. Hence, $\boldsymbol{b}$ is the *true* Burgers vector of the dislocation and is always a lattice vector of the perfect crystal. This is in contrast to the *local* Burgers vector, which is obtained by integration over a path that is closed in the perfect lattice (i.e. in the material coordinate system): the latter integral is affected by elastic strains and differs, in general, from the true Burgers vector.

The elastic displacement vector field $\boldsymbol{U}^{\mathrm{e}}(\boldsymbol{x})$ is given here in spatial coordinates (Eulerian description) and relates the dislocated configuration $\mathscr{B}$ to a virtual, stress-free reference configuration $\mathscr{L}$. For now, we assume $\mathscr{L}$ to be the perfect crystal lattice. The dislocated crystal is divided into a *good* region, where the atomic structure deviates only slightly from the ideal lattice structure, and a *bad* region near the dislocation line (the defect core) where the displacements are large and the atomic arrangements do not resemble the regular lattice. The Burgers circuit C must lie entirely in the *good* region.

In an atomistic description of the solid, the Burgers circuit is composed of a set of vectors $C = \{\Delta\boldsymbol{x}\}$ that form a closed atom-to-atom path (figure 1(b)), such that

$$\boldsymbol{0} = \sum_C \Delta\boldsymbol{x}. \tag{2}$$

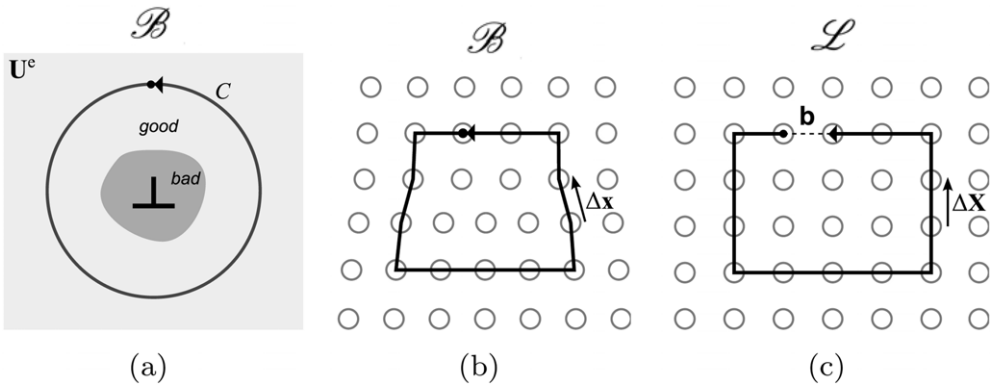


Figure 1. The Burgers circuit method in a continuum setting (*a*) and in an atomistic crystal (*b*), (*c*).

The line integral (1) can then be discretized. In our framework each vector $\Delta\boldsymbol{x}$ of the circuit is mapped to a corresponding vector $\Delta\boldsymbol{X}$ in the perfect crystal $\mathscr{L}$ (figure 1(*c*)). The true Burgers vector is then

$$\boldsymbol{b} = \sum_C \Delta\boldsymbol{X}. \tag{3}$$

2.2. *Mapping vectors from the dislocated to the perfect crystal*

To implement an automated dislocation identification based on the Burgers circuit method, two key problems need to be solved: (1) How to construct the atom-to-atom path C, such that it encloses a single dislocation, and (2) how to determine the ideal vector $\Delta\boldsymbol{X}$ that corresponds to a given atom-to-atom vector $\Delta\boldsymbol{x}$. We address the elastic mapping $\Delta\boldsymbol{x} \rightarrow \Delta\boldsymbol{X}$ first.

For the following considerations, we require that every vector $\Delta\boldsymbol{x}$ in a Burgers circuit should connect two atoms that are nearest neighbors (NNs). In case it does not, it should be always possible to refine the path C and replace $\Delta\boldsymbol{x}$ with a sequence of shorter atom-to-atom vectors ($\Delta\boldsymbol{x} = \sum_i \Delta\boldsymbol{x}_i$), such that each $\Delta\boldsymbol{x}_i$ connects NN atoms. For this purpose one can invoke an appropriate shortest-path search algorithm [6] to find a sequence of NN vectors connecting two distant atoms.

According to Frank [4], a proper Burgers circuit must pass through so-called *good* material only. Accordingly, an atom is defined to be part of a *good* region if the type of its NN environment is that of the perfect lattice. To assign a lattice type to each atom, one can use an atomic structure identification algorithm [1], such as the common neighbor analysis (CNA) [7] (figure 2(*b*)). On the other hand, any atom is defined to be a bad atom if its atomic environment is found to be different from the reference lattice or is not recognized at all by the structure identification algorithm. By structure identification algorithm we mean any computational method that attempts to assign a structural type e.g. fcc, bcc or hcp, to every atom in the crystal by analyzing the relative positions of its NN atoms (figure 3).

The vector $\Delta\boldsymbol{x}$ is defined to belong to a *good* region if at least one of two atoms it connects is a *good* atom. That means the atom has been identified to be of the structural type corresponding to the reference lattice *and* its NN vectors can be unambiguously mapped to corresponding bonds in the reference lattice in a one-to-one fashion (figure 2(*c*)). Since according to our initial requirement, the second atom, pointed to by the vector $\Delta\boldsymbol{x}$, must be among the NNs of the first, this determines the corresponding vector $\Delta\boldsymbol{X}$ in the ideal lattice configuration $\mathscr{L}$.

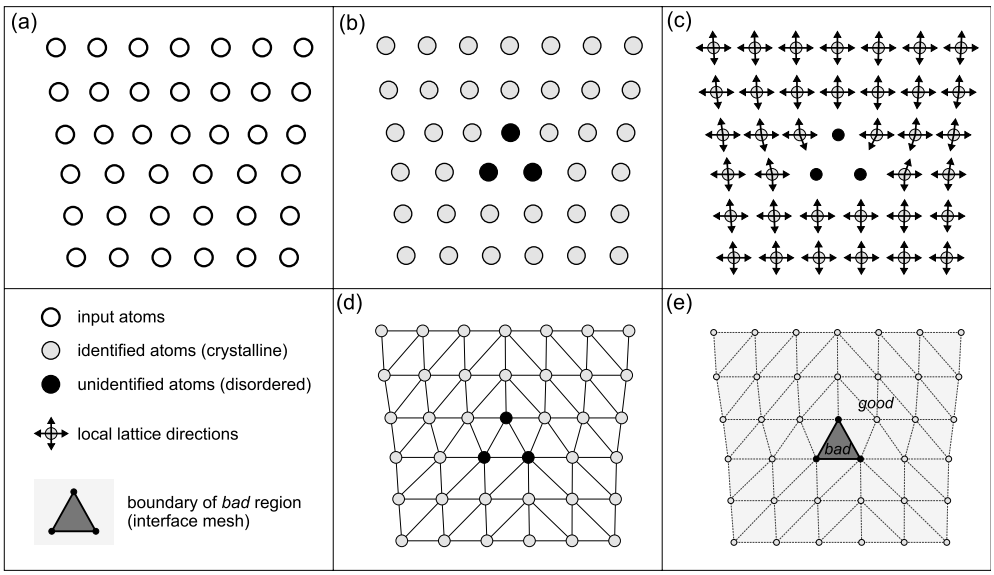


Figure 2. Two-dimensional schematic of the interface mesh construction steps: (*a*) input set of atomic coordinates; (*b*) atomic structure identification, labeling atoms as good (crystalline) or bad (unmatched); (*c*) mapping of the atomic bonds to ideal lattice directions (only for good atoms); (*d*) generation of the space-filling Delaunay tessellation; (*e*) partitioning of the tessellation into contiguous *good* and *bad* regions, and construction of the separating interface boundary.

We refer to the earlier papers [3, 8, 9] for more details on this procedure to obtain the elastic mapping $\Delta \boldsymbol{x} \rightarrow \Delta \boldsymbol{X}$.

Note that, for crystal lattices with point symmetries, the mapping of NN bonds to ideal lattice directions is not unique: for the same central atom, there exist as many equivalent one-to-one mappings (permutations) of its neighbor bonds to ideal lattice directions as there are elements in the local point symmetry group. For performing the Burgers circuit analysis it is necessary to pick the mapping in a consistent way for all atoms in the *good* region. A *locally compatible* mapping requires $\Delta \boldsymbol{X}_{ab} = -\Delta \boldsymbol{X}_{ba}$ to hold for any two NN atoms a and b, and $\Delta \boldsymbol{X}_{ab} + \Delta \boldsymbol{X}_{bc} + \Delta \boldsymbol{X}_{ca} = \boldsymbol{0}$ for a triplet of NN atoms a, b, c (all three being mutual NNs). In other words, the local lattice directions assigned to the bonds of a *good* atom must be aligned with those picked for its neighboring sites. To generate a *globally compatible* mapping for a *good* region, we start at some seed atom in that region for which we pick an arbitrary lattice orientation. We then adjust the bond mappings of all its neighbors such that their respective orientations become locally compatible. As described in more detail in [3] this is repeated recursively for the neighbors of neighbors and so on, until every atom in the *good* crystal region has been visited and 'reoriented' by permuting its neighbor list. Note, however, that no globally compatible mapping exists if the *good* region is intersected by a disclination.

2.3. Partitioning space into good and bad regions

In general, a defective crystal consists of one or more *good* regions in which one or more *bad* regions are embedded. The *bad* regions are associated with the cores of crystal defects (dislocations and others). At the same time a finite crystal may be thought of as being surrounded by an infinite *bad* region bordering with the crystal along its outer surfaces. Owing to the Burgers vector conservation a dislocation line cannot end inside a crystal. Furthermore,

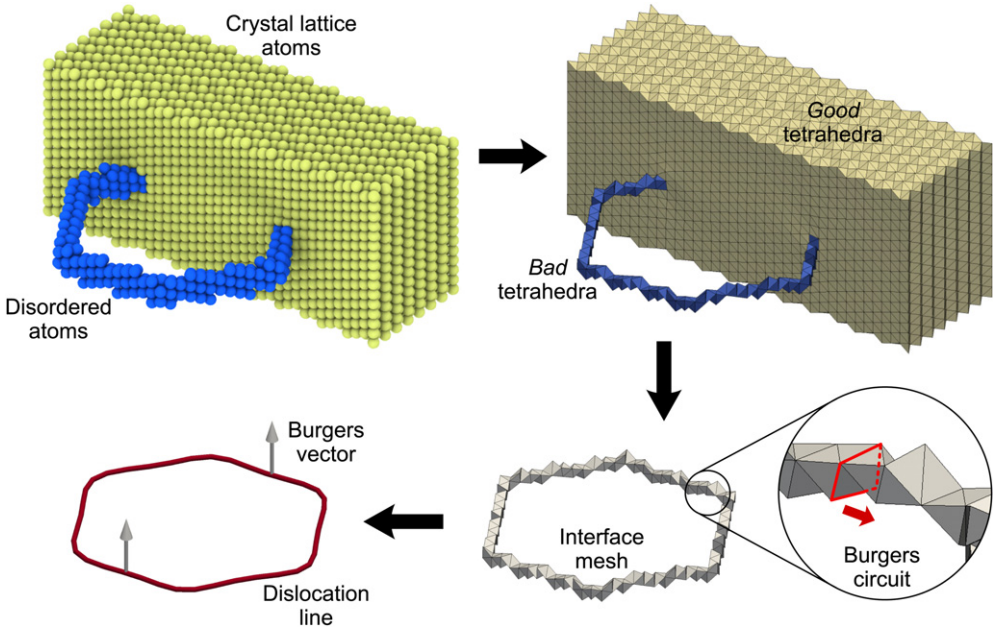


Figure 3. Illustration of the generalized DXA. In this example, a prismatic dislocation loop in a bcc single crystal is identified, indexed and converted to a continuous line representation (see text for a detailed description).

the *bad* region surrounding a dislocation that ends on the crystal's surface becomes connected with the outer bad region.

Two Burgers circuits are said to be *equivalent* if they are located within the same *good* region, and if one circuit can be continuously transformed into the other without breaking the circuit or letting it cross the region's boundary. If two Burgers circuits are equivalent, then they enclose the same dislocation (or the same set of dislocations) and their associated Burgers vectors are equal. A circuit encloses a single dislocation line if and only if its shortest equivalent circuit is completely contained within the boundary between the *good* crystal and the *bad* core region. Thus, for optimal resolution, the dislocation identification algorithm should work only with such irreducible circuits, which are embedded in the *good* region's boundary, to guarantee that each individual dislocation line will be identified.

Computationally, we partition the crystal into *good* and *bad* regions as follows. Taking the set of atomic coordinates, we first generate a space-filling decomposition of the simulation domain into tetrahedral elements (figure 2(*d*)) using the *Delaunay* construction[1]. The tetrahedra of the tessellation connect four atoms each, and they will subsequently be classified as belonging to either a *good* or a *bad* region. To this end, we attempt to assigned an ideal vector $\Delta \boldsymbol{X}_{ab}$ to each edge $a \rightarrow b$ of the tessellation using the method described in section 2.2. The assignment, however, may fail in the following cases:

- An edge may be completely inside a *bad* crystal region meaning that the structure identification algorithm has failed to identify the local crystal structure around both end atoms so that no local lattice orientation can be established.

[1] We do not make use of the specific properties of a Delaunay tessellation. Any other space-filling tessellation connecting all atoms and composed of non-empty, non-overlapping tetrahedra would work too.

- A tessellation edge may connect two distant atoms that are not immediate neighbors. In such a case we attempt to find an atom-to-atom path that connects the two end atoms of the edge as described in section 2.2. Failure to map any one of the sub-steps in that path to an ideal lattice path signals that the path crosses a *bad* region and that our algorithm is unable to assign an ideal vector to the tessellation edge.

Even when no ideal lattice vector could be assigned to a tessellation edge for one of the above reasons, the algorithm does not give up and tries to infer an ideal lattice vector from the ideal vectors assigned to two adjacent edges forming a triangle with some third vertex. Only when all of this fails, the tessellation edge is left unassigned.

In the next step, each tetrahedral element is classified as either a *good* or a *bad* tetrahedron based on the ideal vectors assigned to its six edges (figure 2(*e*)). A tetrahedron is classified as *bad* if mapping to the ideal lattice failed for at least one of its edges. For a tetrahedron to be classified as *good*, all six of its edges should have been assigned ideal lattice vectors, which must form a *compatible* set. This means the six lattice vectors must form a closed tetrahedron in the reference configuration $\mathscr{L}$, i.e. the sum of the ideal edge vectors must vanish for each of the four triangular faces of the tetrahedron (e.g., $\Delta\boldsymbol{X}_{ab}+\Delta\boldsymbol{X}_{bc}+\Delta\boldsymbol{X}_{ca} = \boldsymbol{0}$ for a triangular face *a*–*b*–*c* of the tetrahedron). This condition corresponds to performing the Burgers circuit test for each of the four faces of the tetrahedron and making sure that none of the faces is intersected by a dislocation. Within a *good* tetrahedron, the elastic mapping between the physical configuration $\mathscr{B}$ and the stress-free reference configuration $\mathscr{L}$ is compatible and unique (up to rigid-body rotations) allowing one to compute a local elastic deformation gradient [10].

2.4. Construction of the interface mesh

To resolve individual dislocations, especially in dense dislocation tangles, it is important to ensure that Burgers circuits over which the integral (3) is computed do not enclose more than one dislocation. As was stated in section 2.3, for a circuit to have minimal length among all other possible circuits enclosing the same dislocation it should lie on the boundary between the dislocation's core region (bad) and the surrounding crystal lattice (good). To search for such minimal circuits efficiently, it is useful to first extract the boundary surface that separates *good* tetrahedra from *bad* tetrahedra. This boundary is a two-dimensional manifold consisting of all triangular tessellation faces shared by one *good* and one *bad* tetrahedron. We refer to this oriented and closed manifold as the *interface mesh*. It consists of vertices (atoms), edges and triangular facets. With the interface mesh being a closed manifold, each edge of it is shared by exactly two triangular faces, and it is convenient to use a *halfedge* structure [11, 12] to represent it. In this data structure, each halfedge $a \rightharpoonup b$ is paired with an opposite halfedge $a \leftharpoonup b$, bounding the facet on the opposite side.

While generating the interface mesh, the ideal vectors $\Delta\boldsymbol{X}_{ab}$ previously assigned to the Delaunay edges are copied over to the halfedges of the interface mesh. Since each triangular face of the interface mesh is derived from a face of a *good* tetrahedron, the three halfedges of every interface mesh face will have three ideal lattice vectors associated with them that add up to zero. Furthermore, given that a halfedge $a \rightharpoonup b$ is assigned an ideal vector $\Delta\boldsymbol{X}_{ab}$, its reverse halfedge $a \leftharpoonup b$ is assigned the reverse vector $\Delta\boldsymbol{X}_{ba} = -\Delta\boldsymbol{X}_{ab}$.

2.5. Construction of Burgers circuits and discovery of dislocations

To discover dislocation lines, the algorithm constructs trial circuits on the interface mesh (which encloses dislocations as well as non-dislocation defects, i.e. all *bad* material). Such a trial circuit is a closed sequence of three or more halfedges, and its Burgers vector is computed from

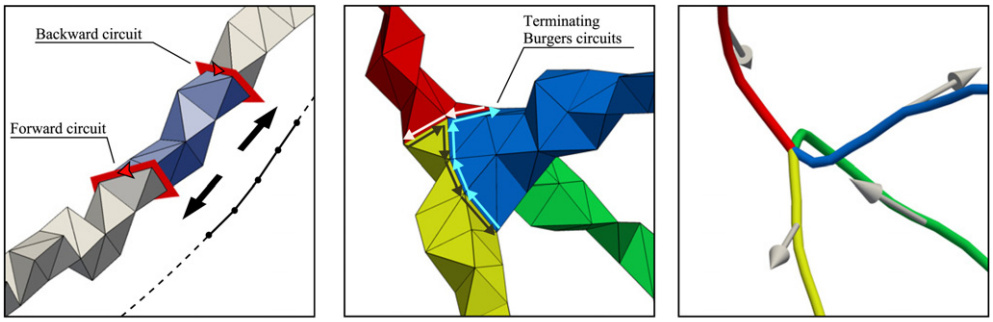


Figure 4. (*a*) Illustration of the sweeping process by a stepwise advancement of a forward and a backward circuit. (*b*) At a dislocation junction, multiple circuits meet on the interface mesh. (*c*) Upon detection of the junction, the corresponding line segments are connected in the generated line network.

the discrete integral (3) by summing their respective ideal vectors. All possible trial circuits up to some prescribed maximum length (usually not more than ten edges) can be efficiently enumerated using a recursive breath-first search algorithm on the graph of halfedges forming the interface mesh.

Trial circuits of increasing length are generated until a circuit with a non-zero Burgers vector is encountered. As described in the next section (and in more detail in a previous paper [3]), this seed Burgers circuit is subsequently used to discover the rest of the dislocation it encloses.

2.6. Tracing dislocations by sweeping along the interface mesh

Once a circuit with a non-zero Burgers vector is discovered in the previous step, the algorithm creates a duplicate of the circuit in which the sense of all halfedges is reversed. The original circuit is a so-called *forward circuit*, while the reversed copy is a *backward circuit*. The two circuits are then advanced in the opposite directions on the interface mesh sweeping along the dislocation line (figure 4(*a*)). During this sweeping phase, a one-dimensional line representation of the dislocation is generated by computing the new center of mass position of a circuit each time it advances 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 two circuits from sweeping past dislocation junctions or interfaces, we impose a hard limit on the maximum length of circuits.

The circuits are advanced along the interface mesh in elementary moves—sweeping one triangular mesh facet at a time. A single move entails modification of the circuit's halfedge sequence, for instance, by replacing a single halfedge with two other halfedges to traverse one mesh facet on the right-hand side of the circuit[2]. Moves that reduce the length of the circuit are given precedence over moves that extend it. Note that every such unit move is guaranteed to leave the Burgers vector of the circuit unchanged. Once a mesh facet has been traversed by an advancing circuit, it is marked as belonging to the current dislocation segment and no other

[2] The handedness of every triangular mesh facet and, hence, the direction of sweeping moves is defined with respect to the circuit's sense direction and the direction of the positive normal of the interface mesh. The latter is defined to point toward the good crystal region.

circuit is allowed to sweep the same triangle again. The advancement of the Burgers circuit is halted whenever its length reaches a prescribed maximum number of halfedges or all mesh triangles in front of it have been already swept/assigned by some other circuit.

Once it is found that the currently processed dislocation segment cannot be extended any further, the algorithms returns to probing trial circuits on still unvisited parts of the interface mesh (section 2.5) until a new circuit with a non-zero Burgers vector is discovered. Eventually, the algorithm will have found every distinct dislocation segment in the crystal and converted it to a continuous line. At this stage of the analysis, each extracted dislocation segment is associated with the following data record:

- The forward circuit and the backward circuit on the interface mesh that terminate the segment.
- An ordered sequence of points that was obtained by computing the position of a circuit's center of mass each time any of the two circuits advanced. Connecting these points together yields a line representation of the dislocation segment.
- The Burgers vector of the segment computed by evaluating the discrete line integral over the initial forward circuit.

Finally, the algorithm identifies dislocation junctions by detecting instances when multiple circuits run into each other on the interface mesh (figure 4(*b*)). The terminating circuits of three (or more) dislocation segments forming a junction will leave no unvisited facets between them so that every halfedge within one terminating Burgers circuit will have an opposite halfedge that is part of one of the other terminating circuits. The Burgers vector conservation at any junction node identified by the algorithm (figure 4(*c*)) is automatically obeyed simply because, by construction, the two ideal lattice vectors assigned to a halfedge and its opposite halfedge add to zero.

This concludes the description of the general DXA for arbitrary crystal lattices. Note that, in contrast to the original formulation of the algorithm [3], which was limited to fcc and bcc lattices, a space-filling tessellation was used here to partition the simulation domain into *good* and *bad* regions. The interface mesh, subsequently used to detect and sweep dislocation segments, is now defined as the boundary between *good* and *bad* tetrahedra without reference to any underlying atomic structure of the crystal.

3. Partial dislocations and interfacial dislocations

So far we have been defining the *good* crystal as a region containing atoms whose neighbor bonds can be uniquely mapped to a perfect crystal. Here the perfect crystal serves as the strain-free reference configuration to which the Burgers circuit's segments are translated. As we show below, this concept of a strain-free reference configuration can be naturally extended to coherent crystal interfaces.

Consider first the familiar Burgers circuit procedure to the case of a partial dislocation bounding a semi-infinite stacking fault (figure 5(*a*)). To be able to determine the Burgers vector, the Burgers circuit encircling the partial dislocation has to pass through the stacking fault at some point. However, inside the stacking fault the atomic arrangement differs significantly from that of the perfect crystal making it virtually impossible to map atomic bonds inside the fault to a perfect lattice. Consequently, the stacking fault defect must be part of the *bad* crystal region, which a valid Burgers circuit must not cross. Conversely, if the atomic bonds inside the stacking fault could be mapped back to a perfect lattice, then the stacking fault would be part of the *good* region and the Burgers circuit would remain valid. However, in this case the circuit would consist entirely of steps mapped to perfect lattice vectors and the resulting

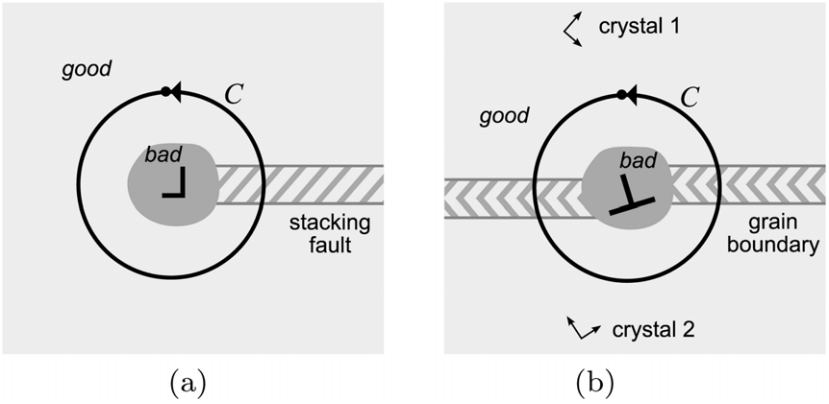


Figure 5. A Burgers circuit enclosing (*a*) a partial dislocation and (*b*) a GB dislocation.

Burgers vector could only be a full lattice vector (including zero), not a partial. To enable the detection and indexing of partial dislocations in coherent crystal interfaces, we therefore need to extend our definition of *good* crystal regions.

3.1. Perfect crystal interface

Perfect and infinite stacking faults, antiphase boundaries, GBs and other coherent (lattice-matched) crystal interfaces all have in common that they do not induce long-ranged stress fields in either of the two crystal lattices. This property makes them candidates for the reference configuration, which is needed to define the purely elastic displacements as required for the Burgers circuit analysis (equation (1)). Therefore, in our extended definition, a *good* crystal region includes all atoms that can be locally mapped to either a perfect lattice or some specific type of coherent crystal interface structure.

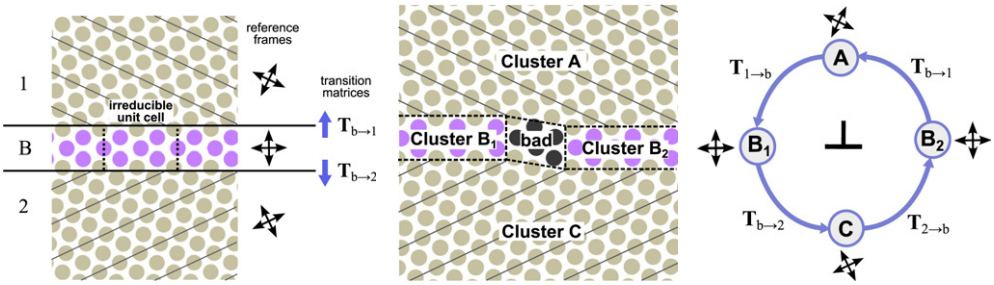
Note that, for our purposes, stacking faults and antiphase boundaries can be viewed as particular types of GBs with no lattice misorientation ($\Sigma = 1$). Therefore, it is sufficient to define only one general mapping procedure for coherent crystal interfaces that will enable the identification of partial lattice dislocations too.

For a given grain misorientation and boundary plane inclination, the equilibrium structure of a coherent interface can be calculated using an appropriate atomistic model [13], usually in a simulation cell with periodic boundary conditions to model an infinite interface. Finding the ground state of a crystal interface may require removing certain atoms prior to relaxing the atomic positions inside the interface, and determining the optimal relative shift of the two crystals [14].

A bicrystal containing a coherent interface can be divided into three regions (see figure 6(*a*)): two perfect crystals and the core region separating them. In the core region, the atomic arrangement is distinctly different from the perfect lattice structure(s). However, in every coherent interface one can identify a periodic motif (or unit template) commensurate with the two adjacent lattices. An atomic structure identification algorithm can be 'trained' to recognize occurrences of such an ideal boundary structure template in the simulation data.

3.2. Dislocated interface

Analogous to a single crystal, a dislocated crystal interface (figure 5(*b*)) can be divided into *good* regions, where the atomic arrangements deviate only slightly from the structure of the



(a) Perfect interface (b) Dislocated interface (c) Cluster graph

Figure 6. (*a*) A bicrystal containing a coherent GB is divided into three regions 1, 2 and B corresponding to two grains and the boundary core. Two rotation matrices $\boldsymbol{T}_{b\to1}$ and $\boldsymbol{T}_{b\to2}$ are derived from this ideal reference template to describe the misorientation of the interface's coordinate frame with respect to Cartesian frames of the two grains. (*b*) For the dislocation analysis, the dislocated interface is partitioned into four contiguous *good* clusters and a distorted dislocation core region. (*c*) From this decomposition, a graph representation of the bicrystal is constructed in which the edges are associated with the rotation matrices between adjacent clusters.

ideal, fully-coherent interface, and *bad* regions near the dislocation lines (their cores) where atomic configurations do not resemble and cannot be mapped onto the ideal interface structure.

Using an atomic structure identification algorithm we mark sections of the dislocated interface where the local atomic arrangement closely matches the known structure of the ideal interface. As a result, the bicrystal is partitioned into contiguous *clusters* of atoms whose local arrangement matches the ideal crystal or the ideal interface template, figure 6(*b*). On the other hand, the atomic structure in the dislocation core does not match any of the pre-defined reference structures: All such atoms are attributed to the *bad* crystal region. Subdividing the *good* region into atomic clusters serves the following purpose: within one cluster, every bond vector $\Delta\boldsymbol{x}$ can be mapped to the same reference configuration, i.e. all ideal vectors $\Delta\boldsymbol{X}$ can be expressed in the same frame of reference associated with that particular cluster.

3.3. Reference frames

Even though, in principle, the Cartesian reference frame for a contiguous cluster of good atoms can be chosen arbitrarily, we use the following convention: the reference frames $\mathfrak{R}_1$ and $\mathfrak{R}_2$, which are used to express ideal vectors in the two crystal grains, are aligned with their respective grain's lattice orientation (so-called isoclinic configurations [15]). This ensures that all ideal lattice vectors, and with them all computed Burgers vectors, will take on familiar values (e.g. $\Delta\boldsymbol{X} = a/2\langle1\,1\,0\rangle$ in fcc). For the perfect interface it is convenient to select the Cartesian reference frame to be aligned with the interface plane normal and the periodic directions of the motif (see figure 6(*a*)). All ideal vectors within the good interface regions are expressed in this third frame $\mathfrak{R}_b$ that is, in general, rotated with respect to the two lattice frames.

The relative orientations of these three reference frames are fully specified by two rotation matrices $\boldsymbol{T}_{b\to1}$ and $\boldsymbol{T}_{b\to2}$, and the product rotation $\boldsymbol{T}_{b\to2}(\boldsymbol{T}_{b\to1})^{-1} = \boldsymbol{T}_{1\to2}$ provides the misorientation of the two crystal grains of the ideal interface.

Any Burgers circuit enclosing an interfacial dislocation must pass through multiple good clusters, each being associated with one of the three frames $\mathfrak{R}_1$, $\mathfrak{R}_2$ or $\mathfrak{R}_b$. Therefore, to

compute the closed circuit integral (1) it is necessary to sum vectors expressed in different reference frames. It is convenient to associate each ideal vector $\Delta \boldsymbol{X}$ with the frame it is expressed in, and store it as a pair $(\Delta \boldsymbol{X}, \mathfrak{R})$. Then the summation rule for two such vectors is

$$(\Delta \boldsymbol{X}_A, \mathfrak{R}_A) + (\Delta \boldsymbol{X}_B, \mathfrak{R}_B) = (\boldsymbol{T}_{A\to B}\Delta \boldsymbol{X}_A + \Delta \boldsymbol{X}_B, \mathfrak{R}_B), \tag{4}$$

where $\boldsymbol{T}_{A\to B}$ is the transformation matrix that translates an ideal vector from frame $\mathfrak{R}_A$ to $\mathfrak{R}_B$. The Burgers circuit sum $\boldsymbol{b} = \sum_C \Delta \boldsymbol{X}$ is evaluated by successively applying summation rule (4) to the sequence of steps in the circuit. The resulting Burgers vector $(\boldsymbol{b}, \mathfrak{R})$ is then expressed in the reference frame of the last lattice vector of the circuit, which is associated with one of the good clusters adjacent to the dislocation core region. One is free to express this Burgers vector in either one of the two Cartesian lattice frames, $\mathfrak{R}_1$ or $\mathfrak{R}_2$, by applying the appropriate transformation matrix.

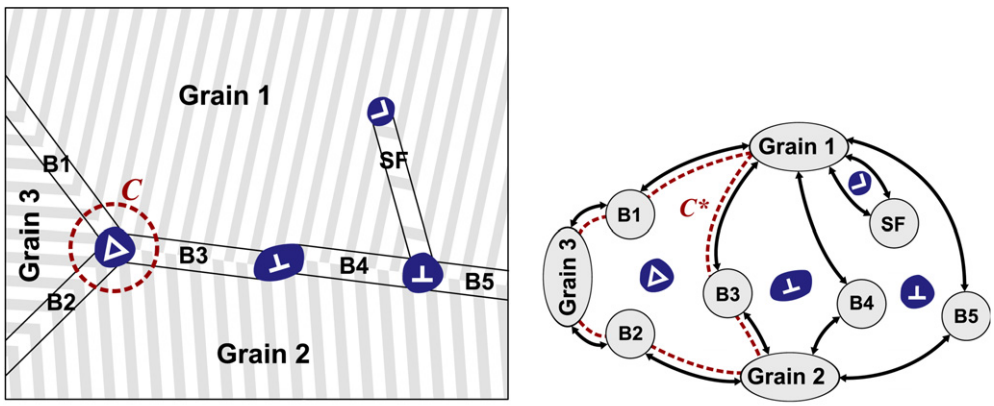
3.4. Polycrystals

A polycrystal consists of multiple grains separated by a network of GBs of various types. Given a sufficiently rich catalog of boundary templates representing various coherent boundaries in the five-dimensional misorientation-inclination space, one can expect every boundary (or boundary facet) in the GB network to be close to one or another ideal boundary type. Leaving aside a possible non-uniqueness of any such partitioning of the 5-space, deviations from the ideal boundary templates are accommodated by secondary GB dislocations, steps, and/or their combinations (misconnections). The main purpose of our new extended DXA is to identify and index such GB dislocations[3].

In the two preceding sections we described how a bicrystal with a single GB dislocation can be partitioned into four clusters (figure 6(*b*)). Here we consider a general polycrystal (figure 7(*a*)) and use the same procedure to partition it into a larger number of clusters—one for each grain or contiguous atom cluster matching an ideal interface template. Any such partitioning can be represented by a graph, as shown in figure 7(*b*). Each node in this graph corresponds to a single contiguous cluster and every pair of neighboring clusters is connected by an edge. Every edge of this so-called *cluster graph* stores the rotation matrix describing the (strain-free) misorientation relationship between two connected clusters. As was already explained, these transformation matrices are pre-computed from the relative orientations of the crystal grains in the corresponding interface template. Note that, during generation of the cluster graph, possible symmetry transformations of the lattice and interface structures need to be taken into account to determine the actual transformation matrix connecting the reference frames of two clusters.

A closed Burgers circuit C, passing through more than one atomic cluster, can be mapped to a circuit C^* in this abstract cluster graph. Sequential application of the transformation matrices associated with the edges of C^* allows us to perform a simple *disclination* test: any circuit passing through n clusters and enclosing a disclination [16] must yield a net rotation, i.e. $\boldsymbol{T}_{n\to 1}\boldsymbol{T}_{n-1\to n}\ldots\boldsymbol{T}_{2\to 3}\boldsymbol{T}_{1\to 2} \neq \boldsymbol{I}$ (Frank–Nabarro circuit). Disclinations are common defects in polycrystal, e.g. associated with the triple junctions, and their identification is an important prerequisite for a robust dislocation identification algorithm. This is because any Burgers circuit enclosing a disclination is invalid and must be discarded even if it is entirely contained in the *good* region.

[3] We leave it to future work to extend the algorithm to indexing steps and misconnections.



(a) Input microstructure (b) Associated cluster graph

Figure 7. (*a*) A small fragment of a polycrystalline microstructure partitioned into clusters by the atomic structure identification algorithm. Three grains and several coherent GB segments (B1–B5) as well as a stacking fault (SF) have been identified. The unidentified *bad* regions in between are potential locations of dislocations. (*b*) The cluster graph derived from the input microstructure. The edges between adjacent clusters are associated with matrices that rotate vectors from one frame to the next. The Burgers circuit C enclosing the triple junction line has been translated into a corresponding circuit C^* in the cluster graph making it possible to detect disclinations (the triangle symbol).

3.5. Synopsis of the generalized DXA

In the preceding sections we developed all the concepts required for a general and robust DXA. The algorithmic steps are as follows:

(i) An atomic structure identification method is invoked trying to match every atom to one of the known ideal structure templates and to partition the system into contiguous clusters of good (matched) atoms.

(ii) A space-filling tessellation of the system is generated using the Delaunay construction.

(iii) The algorithm attempts to assign an ideal vector $(\Delta \boldsymbol{X}, \mathfrak{R})$ to every tessellation edge of the Delaunay tetrahedra based on the local mapping of atomic bond vectors to the ideal templates structure assigned in step 1.

(iv) Every tetrahedron is classified as *good* or *bad* based on the six ideal lattice vectors assigned to its edges (local compatibility test).

(v) A two-dimensional manifold (interface mesh) is built that separates contiguous sets of *good* and *bad* tetrahedra.

(vi) Closed circuits of increasing length are generated on the interface mesh until a first circuit with a non-zero Burgers vector is found.

(vii) This Burgers circuit (and a reverse copy) is advanced along the interface mesh, triangle by triangle, in both directions producing a one-dimensional line representation of the dislocation segment.

(viii) Steps (vi) and (vii) are repeated until the entire interface mesh has been scanned for circuits on its surface with non-zero Burgers vector content and all dislocation segments in the crystal are discovered.

(ix) Dislocation junctions are identified by detecting collisions of multiple circuits on the interface mesh.

4. An application example

We demonstrate the capabilities of the described algorithm with an example. The test configuration, shown in figure 8, is a snapshot taken from an MD simulation of nanoindentation in an fcc bicrystal using an interatomic potential representing aluminum. The bicrystal contains a $\Sigma 11\,[0\,1\,\bar{1}](3\,1\,1)$ symmetric tilt boundary that is impinged on by several prismatic dislocation loops nucleated and pushed into the material by the indenter. The reactions between extrinsic lattice dislocations and the GB result in absorption, dissociation and re-emission of dislocation segments into the second grain. In the depicted simulation snapshot, a prismatic dislocation loop is partially absorbed by the GB and a leading Shockley partial is transmitted into the lower grain, giving rise to complex junction reactions involving several boundary dislocations and partial lattice dislocations within both grains.

For the atomic structure matching algorithm to identify structural types of all atoms outside the dislocation cores, in this particular case it is sufficient to define only three ideal structure templates: the perfect fcc lattice, the intrinsic stacking fault, and the ideal $\Sigma 11$ boundary structure. Figure 8(*a*) shows the outcome of this first phase of the analysis, where the atoms are colored according to the structure templates they match.

Figure 8(*b*) shows the generated interface mesh that encloses all tetrahedra that fail the local compatibility test, i.e. that are intersected by a dislocation. Finally, the extracted dislocation lines are depicted in figure 8(*c*). The Burgers vector of each segment has been determined by summing the ideal vectors over the enclosing circuits, equation (3). By default, the so-computed Burgers vector is expressed in the lattice frame in which the segment is embedded. If needed, it can be automatically transformed into any other frame using the misorientation matrices stored in the generated *cluster graph*.

The configuration depicted by figure 8 comprises only a small sub-region of a larger simulation domain. The dislocation analysis of the full snapshot, which is shown in figure 9 and which comprises 7.5 million atoms, takes 30 s on 64 processor cores. The resulting dataset contains about 300 dislocation segments and 120 junction nodes. The side-by-side picture (figure 9), demonstrates that the algorithm is capable of providing a precise and high-resolution description of the complex dislocation network in the region beneath the indenter. Hence, the dislocation analysis can optimally complement the conventional atomistic visualization, which fails to elucidate the dense and therefore unidentifiable accumulation of defects. More applications of the DXA, such as measuring the dislocation density and quantifying the population of slip systems, have been discussed and demonstrated before [17, 18].

5. Discussion

The described DXA relies on an atomic structure recognition method (to identify local atomic arrangements). This recognition is based on a set of ideal reference structures, or templates, which must be supplied by the user (including the ideal crystal lattice, possible stacking faults, ideal CSL boundaries etc.). In other words, it is left to the user *to define* which reference configurations to use for the subsequent dislocation analysis. That means that any GB dislocation identified by the algorithm is *extrinsic* to the local reference interface type, while the atomic reference pattern carries the remaining *intrinsic* misorientation of the interface. Ultimately, it is left to the user to separate the dislocation content into intrinsic and extrinsic components, whereas the DXA itself is unambiguous.

Resolution and efficiency of the DXA depend upon robustness of the atomic structure identification algorithm. In particular, the DXA can subsequently fail to identify every dislocation in a crystal if the structure identification algorithm fails to recognize some of the

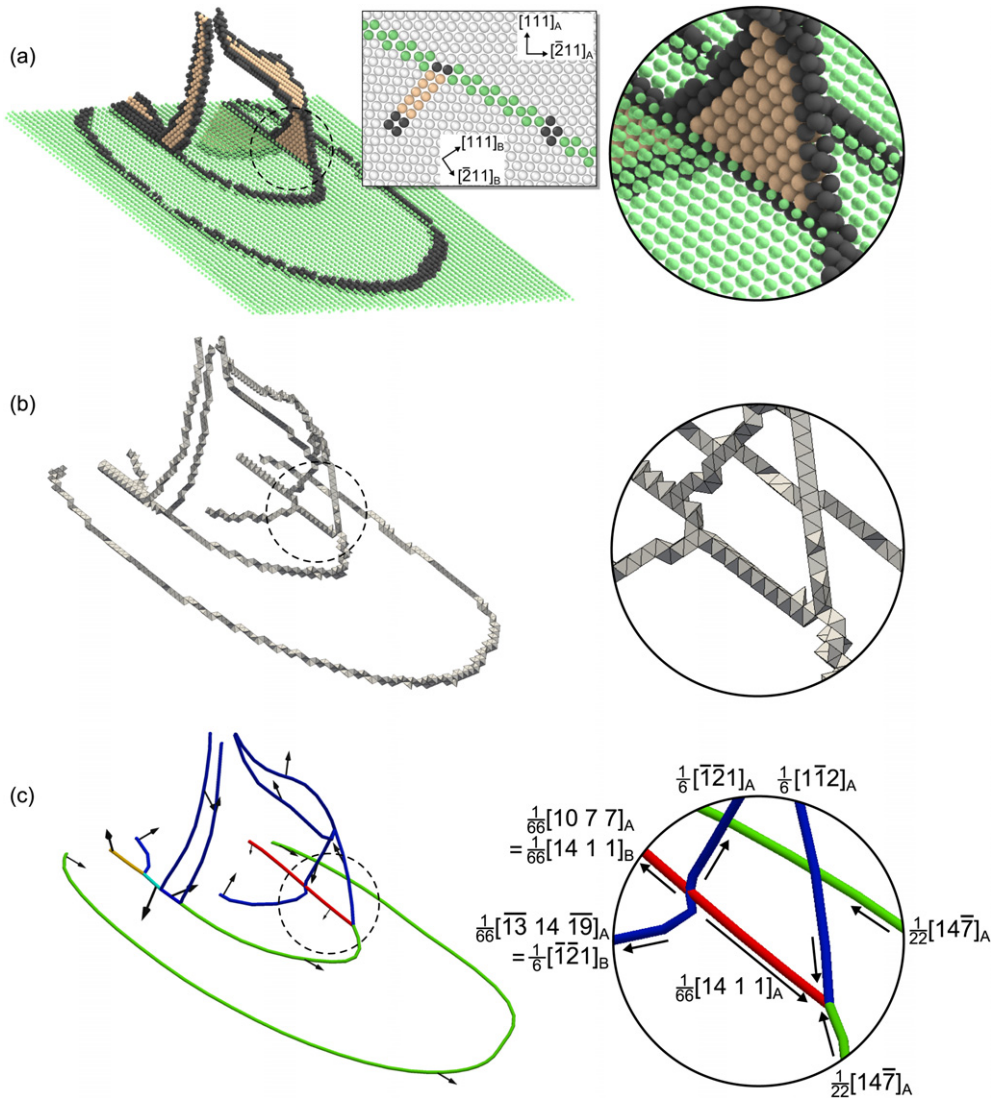
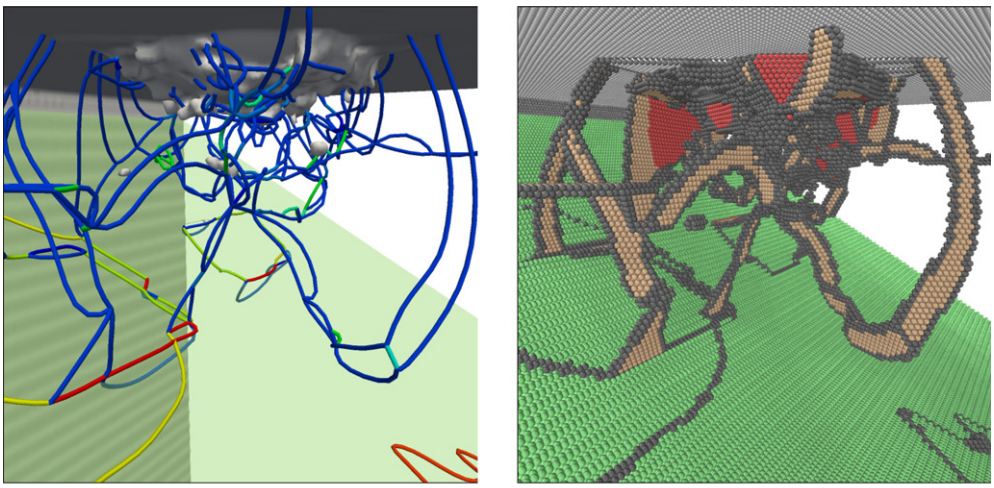


Figure 8. Lattice dislocations reacting with a $\Sigma 11$ $[01\bar{1}](311)$ symmetric tilt boundary in fcc aluminum. The depicted configuration is a small fragment of a larger nanoindentation simulation (see figure 9). (*a*) Local atomic structure analysis reveals a $\Sigma 11$ boundary (green atoms), stacking faults (beige atoms), and disordered atoms (dark gray). The inset shows the orientation of reference frames in the two grains. (*b*) An interface mesh enclosing the dislocation cores is generated. (c) Dislocation network extracted by the algorithm. The line colors indicate the Burgers vector types, e.g. $\frac{1}{6}\langle 112\rangle \mapsto$ blue, $\frac{1}{22}\langle 147\rangle \mapsto$ green, $\frac{1}{66}\langle 1\ 1\ 14\rangle \mapsto$ red. In the magnified view on the right, all dislocation segments are labeled with their respective Burgers vectors as determined by the algorithm. The Burgers vectors can be represented in either one of the two lattice frames (grains A and B).

good atoms surrounding their dislocation cores. Development of robust structure recognition methods, capable of discriminating between a wide variety of distinct atomic arrangements in the presence of elastic distortions and other perturbations, remains a challenge. Methods such as the CNA are sensitive to thermal vibrations that can impair their structure recognition acuity and, subsequently, impede the dislocation analysis. In addition to the described DXA, other



(a) (b)

Figure 9. Results of the automated dislocation analysis of a nanoindentation simulation showing the extracted dislocation line network (*a*) and the conventional atomistic visualization (*b*) for comparison. As indicated by the line colors, dislocations in the $\Sigma 11$ grain boundary have non-standard Burgers vectors in contrast to those occurring in the bulk fcc region (which include perfect dislocations, Shockley partials, and stair-rod dislocations).

computational crystal analysis techniques, such as the decomposition of atomic-level strain into elastic and plastic components [10], depend on the same class of methods for identifying local atomic structures and, accordingly, may suffer the same limitations.

Any two dislocations are recognized as separate lines only if their cores do not overlap, i.e. each core must be fully surrounded by some *good* material. Otherwise the generated interface mesh will enclose both dislocations, and they will be identified as a single super dislocation. The cores of some GB dislocations can be fairly wide making it difficult to resolve defects in the boundary. In such cases, identification and indexing of GB dislocations can be difficult and error-prone no matter which type of method—manual or automated—is employed.

6. Summary

We have described an extended version of the dislocation extraction algorithm (DXA) [3] that identifies and indexes dislocations in arbitrary lattices and coherent crystal interfaces, including partial dislocations and secondary grain boundary dislocations. Our algorithm aims to automatically detect any such dislocations in atomistic configurations, compute the Burgers vectors, and represent arbitrary dislocation networks by connected and fully indexed line segments so familiar from the classical dislocation theory.

Our approach is based on the (implicit) calculation of the incompatible elastic displacement field surrounding dislocations that involves mapping bond vectors from the distorted atomic configuration to an ideal, strain-free reference configuration. The reference configuration is determined locally using an atomic structure identification algorithm and a set of ideal structural motives that include, in addition to the perfect lattices, also the ideal stacking faults and coherent crystal interfaces.

The described algorithm reveals dislocations in atomistic simulations of complex crystal structures and automatically extracts and labels them in a matter of seconds. Therefore, the

extended dislocation extraction algorithm has clear advantages over the manual Burgers circuit analysis that is error-prone, laborious and, in many cases, even unfeasible. In developing the algorithm we tried to achieve the often-conflicting goals of generality, efficiency and high resolution. We hope the described technique will enable its users to significantly enhance the useful information content of atomistic simulations, which often suffer from intractable complexity.

Acknowledgments

This work was performed under the auspices of the US Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. VB acknowledges support of the Department of Energy, Basic Energy Sciences, Materials Sciences and Engineering Division.

References

[1] Stukowski A 2012 Structure identification methods for atomistic simulations of crystalline materials *Modelling Simul. Mater. Sci. Eng.* **20** 045021
[2] Stukowski A 2010 Visualization and analysis of atomistic simulation data with OVITO—the Open Visualization Tool *Modelling Simul. Mater. Sci. Eng.* **18** 015012 (Software available at http://ovito.org/)
[3] Stukowski A and Albe K 2010 Extracting dislocations and non-dislocation crystal defects from atomistic simulation data *Modelling Simul. Mater. Sci. Eng.* **18** 085001
[4] Frank F C 1951 LXXXIII. Crystal dislocations—elementary concepts and definitions *Phil. Mag. Ser. 7* **42** 809–19
[5] Hirth J P and Lothe J 1982 *Theory of Dislocations* 2nd edn (New York: Wiley)
[6] Dijkstra E W 1959 A note on two problems in connexion with graphs *Numer. Math.* **1** 269–71
[7] Honeycutt J D and Andersen H C 1987 Molecular dynamics study of melting and freezing of small Lennard-Jones clusters *J. Phys. Chem.* **91** 4950–63
[8] Hartley C S and Mishin Y 2005 Characterization and visualization of the lattice misfit associated with dislocation cores *Acta Mater.* **53** 1313–21
[9] Stukowski A, Markmann J, Weissmüller J and Albe K 2009 Atomistic origin of microstrain broadening in diffraction data of nanocrystalline solids *Acta Mater.* **57** 1648–54
[10] Stukowski A and Arsenlis A 2012 On the elastic-plastic decomposition of crystal deformation at the atomic scale *Modelling Simul. Mater. Sci. Eng.* **20** 035012
[11] Weiler K 1985 Edge-based data structures for solid modeling in curved-surface environments *IEEE Comput. Graph.* **5** 21–40
[12] Mäntylä M 1987 *An Introduction to Solid Modeling* (New York: Computer Science Press)
[13] Olmsted D L, Foiles S M and Holm E A 2009 Survey of computed grain boundary properties in face-centered cubic metals: I. Grain boundary energy *Acta Mater.* **57** 3694–703
[14] Wolf D 1990 Structure-energy correlation for grain-boundaries in fcc metals: III. Symmetrical tilt boundaries *Acta Metall. Mater.* **38** 781–90
[15] Mandel J 1971 *Plasticité Classique et Viscoplasticité* (CISM Course No 97) (Berlin: Springer)
[16] Nabarro F R N 1967 *Theory of Crystal Dislocations*. (Oxford: Clarendon)
[17] Stukowski A and Albe K 2010 Dislocation detection algorithm for atomistic simulations *Modelling Simul. Mater. Sci. Eng.* **18** 025016
[18] Stukowski A, Albe K, and Farkas D 2010 Nanotwinned fcc metals: strengthening versus softening mechanisms *Phys. Rev.* B **82** 224103