JOM, Vol. 66, No. 3, 2014
DOI: 10.1007/s11837-013-0827-5

Computational Analysis Methods in Atomistic Modeling of Crystals

ALEXANDER STUKOWSKI[1,2]

1.—Fachgebiet Materialmodellierung, Institut für Materialwissenschaft, Technische Universität Darmstadt, Darmstadt, Germany. 2.—e-mail: stukowski@mm.tu-darmstadt.de

This article discusses computational analysis methods typically used in atomistic modeling of crystalline materials and highlights recent developments that can provide better insights into processes at the atomic scale. Topics include the classification of local atomic structures, the transition from atomistics to mesoscale and continuum-scale descriptions, and the automated identification of dislocations in atomistic simulation data.

INTRODUCTION

Materials properties and the underlying mechanisms at the microscale are usually governed by structural changes of the material and the presence, creation, or interaction of crystal defects. Molecular dynamics (MD) and other atomistic modeling techniques are powerful and established tools that enable us to study the dynamics of such processes with full atomic detail. However, the accurate and robust identification of phases and crystal defects, either manually or computationally, is an essential requirement for understanding and interpreting the observed atomic-scale processes in such simulations. Classical MD simulations do not keep track of the crystal structure and crystal defects explicitly and merely yield the trajectories of motion of individual atoms. This makes it necessary to explicitly recover such information and identify crystal defects in atomistic simulation output in a postprocessing step using adequate computational methods.

Computer algorithms that can identify the characteristic arrangements of atoms in crystal phases and crystal defects typically take into account certain geometric fingerprints[1,2] such as distances between atoms, angles between bonds, or the topology of the bond network. There is, however, a class of defects that cannot be identified using this type of approach: Dislocations, which play a key role in the plastic deformation of many crystalline materials, require a completely different treatment. To this end, the Burgers circuit method, the classical recipe to test for the presence of a dislocation and to determine its Burgers vector, has recently been transcribed into a fully automated computer algorithm,[3,4] which allows to identify and extract all dislocation segments in a simulated crystal.

Computational analysis methods also play important roles in linking atomistic descriptions of a solid to higher-level theories or coarser materials models. Classical continuum concepts such as the strain and deformation fields, or mesoscale concepts such as the grain boundary network and the surface geometry, have no immediate definitions at the atomic level. Thus, new theoretical concepts and computational methods are necessary to derive such information from atomistic simulation models and to make it available for a treatment with classical theories—or as input to coarser materials models. Suitable computer algorithms need to be developed to explicitly compute materials quantities such as grain boundary and surface area, plastic strain, or lattice orientation and rotation from the discrete atomic coordinates.

The following sections briefly describe the underlying concepts, capabilities, and applications of advanced computational methods that have been devised to address the challenges in atomistic modeling outlined in this introduction.

IDENTIFICATION OF ATOMIC STRUCTURES

Atomistic simulations have contributed significantly to our understanding of fundamental materials properties and their underlying atomic-scale phenomena. With the availability of more accurate simulation models and more powerful computing resources, increasingly complex problems in

(Published online December 10, 2013)

materials science can be addressed and studied at the atomistic level. However, to interpret atomistic simulations, qualitative and quantitative knowledge of the local structure and composition of the material and its defects is essential. For this purpose, several computational methods have been developed in the past that assign a structural type to every atom based on an analysis of their local environments (see Ref. 5 for a review).

Conventional classification techniques such as the common neighbor analysis method (CNA),[6] the bond-angle analysis method,[7] and the centrosymmetry parameter[8] are often used in atomistic simulation studies to filter out atoms that form a perfect crystal lattice and to reveal internal defects such as dislocations or grain boundaries. These methods are efficient enough to be performed on the fly during a simulation, which, for example, allows to reduce the output data from large-scale simulations by storing only particles with an atypical environment.

In general, a well-designed atomic structure identification method should fulfill several requirements, which sometimes are in conflict with each other.[5] The method should be able to correctly distinguish several structural environments (phases) based on the local arrangement of atoms and irrespective of the crystal orientation (rotational and translational invariance). At the same time, the algorithm should reliably assign a structure type to most particles in the system and avoid errors arising from displacements of particles from their equilibrium or symmetry positions due to thermal vibrations or elastic lattice distortions.

The identification methods in use today take only one or two shells of neighbors around an atom into account to classify the local structure as depicted in Fig. 1a. This effectively limits their application to simple phases with a monatomic basis, such as face-centered cubic (fcc), hexagonal close packed (hcp), or body-centered cubic (bcc). The continuing development of accurate interatomic potentials for binary or ternary systems, however, has recently shifted the focus of atomistic simulations toward more complex systems, which might exhibit intermetallic phases, phase transitions, and intricate interface structures. The identification of such extended structures is beyond what traditional methods can accomplish, and new algorithms are being developed for this purpose.

Schablitzki et al.[9] have recently proposed a computational method to determine intermetallic phases based on extended topological fingerprints. Their polyhedron-based analysis method can discern crystal phases and identify point and extended defects in transition metal alloys. The method proceeds in two steps: First, the local environment of each atom is classified by constructing its (Voronoi) coordination polyhedron and finding the best-matching reference polyhedron in the database. In the second stage, the medium-range order in the vicinity of each atomic site is identified by performing one or more hops to neighboring sites. The algorithm counts how often certain hopping sequences of coordination polyhedra occur and then compares this characteristic number with the stored fingerprints in the database.

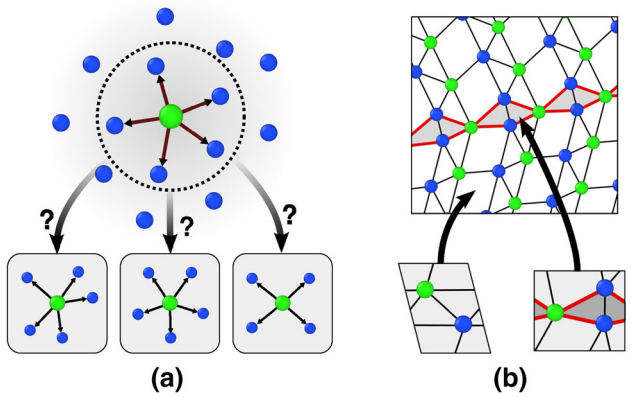

Fig. 1. Local versus nonlocal atomic structure identification strategies: (a) finding the best-matching structure in the database based on a local coordination analysis and (b) identification of complex lattices and extended defect structures using a graph-based pattern matching method.

Labeling crystal defects in simulation data is another important application of structure identification techniques. Tracking defect reactions and quantifying defect densities in large-scale atomistic simulations are problems that call for automated algorithms similar to those used in other fields like computer vision and data mining. Grain boundaries and phase interfaces, for instance, are often composed of characteristic structural units or three-dimensional atomic patterns. Extended identification methods like the one described above can be used to find occurrences of such structures in a simulation (Fig. 1b). From a user's point of view, algorithms are beneficial that support an extendable database of reference structures, i.e., where fingerprints are not hardcoded in the algorithm. For instance, the software Crystal Analysis Tool, developed by the author, can automatically generate search patterns for complex crystal phases and defects of interest from template structures and unit cells provided by the user. The encoded fingerprints are stored in a catalog file and can subsequently be used by the search algorithm to find occurrences of the corresponding atomic structures in simulation snapshots. Similar to the approach of Schablitzki et al., our pattern-matching algorithm comprises two stages: First, the nearest-neighbor environment of each atom is classified using a conventional coordination structure identification method. The user can choose between several techniques such as the CNA (ideal for highly symmetric coordination structures) or the neighbor distance analysis[5] method (for low-symmetry structures typically found in extended defects). At this point of the

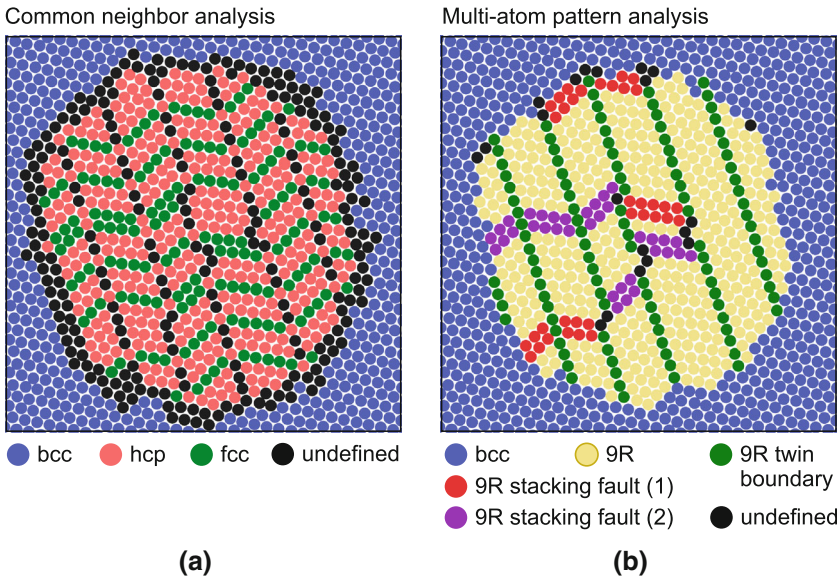

Fig. 2. Cross-section of a simulated Cu-rich precipitate with multiply twinned 9R structure in bcc-Fe.[11] Colors indicate the local structure type as determined by (a) the conventional CNA and (b) the graph-based analysis method implemented in the Crystal Analysis Tool (Color figure online).

matching process, the structural bonds between neighboring atoms are established. This happens through an adaptive cutoff scheme[5] to support multiphase systems with varying nearest-neighbor distances. As a result of this first algorithm step, the input data set has been converted into a graph representation: Each atomic site is now labeled with its local coordination type and chemical type, and a network of bonds (edges) has been constructed.

In the second phase of the pattern-matching process, an efficient subgraph-matching algorithm[10] is employed to find regions in the input graph that match a pattern graph from the database. A proper match requires not only that the topology of the bond network is identical, but also that the coordination types and chemical types of the involved atoms match. Our subgraph searching algorithm supports pattern graphs that are periodic in three, two, or zero directions, corresponding to crystal phases and point-like and planar defects, respectively.

Figure 2 shows a Fe-Cu alloy simulation that has been processed by the described structure identification algorithm. In this simulation study,[11] a novel Monte Carlo technique[12] was used to model the structural phase transformation of the precipitate from bcc to a multiply twinned 9R structure ("herringbone" structure), which is a closed-packed phase consisting of an alternating sequence of hcp and fcc-like atomic layers. Conventional local coordination analysis methods such as the CNA are insufficient to identify the 9R lattice with its complex unit cell. The described graph-based pattern matching algorithm, however, allows to mark all regions of the simulation snapshot that match the 9R pattern and, given appropriate template structures, even can extract defects such as stacking faults and twin boundaries automatically.

FROM ATOMISTICS TO CONTINUUM DESCRIPTIONS

Atomistic simulation methods can provide direct insight into competing deformation mechanisms. However, to develop a theoretical understanding of such processes, one has to connect observations from atomistic models to continuum theory and mesoscale models. One strategy is to extract measurables from atomistic simulations, which have direct implications for higher-level theories or which can be incorporated into continuum mechanical models of material behavior. In addition to, e.g., defect densities, which can be obtained by means described in the other parts of this article, kinematic quantities such as displacement or strain fields can help to interpret processes at the nanoscale. One example is the definition of an atomic deformation gradient developed by Zimmerman et al.,[13] which describes the microscopic deformation at each atomic site. This computation is performed on the basis of two simulation snapshots of an atomistic system: the initial state and the deformed state of the solid. Similarly to prior methods,[14–17] the atomic deformation gradient tensor $\mathbf{F}$ is computed from the relative displacements of nearby particles using a least-squares procedure. In practice, the motion of all nearest neighbors (in the undeformed state) is considered for the atomic tensor, but the extent of nonlocality of the metric can be controlled by taking into account additional neighbor sets. Including the influence of distant particles will smooth out spatial fluctuations and thermal

noise[18] and must be controlled with appropriate weight functions.[14] Other quantities such as the atomic shear strain can be derived from the atomic deformation gradient tensor **F**, which are very useful in revealing local material deformation. In particular, MD simulations of amorphous materials like metallic glasses[17,19] benefit from such analysis techniques, as they can make shear transformation zones and the development of shear bands visible.

The atomic strain tensor discussed above, and other atomic quantities like the *slip vector,*[20] help to reveal the effects of competing deformation mechanisms such as dislocation motion, twinning, or grain boundary migration and are well suited for visualization purposes. However, they are still defined discretely at atomic sites. For a direct coupling with continuum models,[21,22] an integrable field representation is necessary, and care needs to be taken when converting atomistic to field quantities. In the case of the deformation gradient, this conversion step can be effectively avoided by directly computing the **F** tensor field on a Delaunay mesh connecting all particles.[16] In this approach, a space-filling, nonoverlapping tessellation is constructed on the basis of the initial atomic positions, which subsequently follows the atoms during deformation (Fig. 3). At any simulation timestep, the local deformation tensor can be directly calculated in a tessellation element from the displacements of its corner atoms,[23] yielding a piecewise constant deformation gradient field. Time and volume averaging can then be applied to smooth this microscopic field and project it to a coarser grid.

ELASTIC–PLASTIC DECOMPOSITION OF DEFORMATION

In the framework of plasticity theory, it is common to perform a multiplicative decomposition of the deformation into elastic and plastic parts,[24] for example $\mathbf{F} = \mathbf{F}^{\mathbf{e}}\mathbf{F}^{\mathbf{p}}$. Here, $\mathbf{F}^{\mathbf{p}}$ refers to the irreversible deformations induced by the creation and motion of defects, while the gradient $\mathbf{F}^{\mathbf{e}}$ describes the reversible deformations caused by elastic loading of the material. From this definition, it is obvious that such a decomposition of the total, compatible deformation gradient field **F** could provide additional insight into the various competing deformation mechanisms and their role in nanoscale plasticity of crystalline materials. In classical MD and statics, however, the full deformation is the only quantity that is readily accessible from the atomic displacements. That is why it has not been clear for a long time how to perform the elastic–plastic decomposition in the context of atomistics.

As part of a study on the origin of microstrain broadening in x-ray diffraction patterns of nanocrystalline metals,[25] a method was proposed by the author to compute a *purely elastic* atomic deformation gradient tensor. Assuming that an atom, together with its neighbors, forms an elastically strained but otherwise perfect crystal, we can computationally map this local group of atoms to corresponding sites in an ideal lattice. Once this mapping has been determined, we can compute the elastic deformation gradient $\mathbf{F}^{\mathbf{e}}$ from it in the same way we compute **F**, but now taking the ideal crystal lattice as reference configuration instead of an initial simulation snapshot.

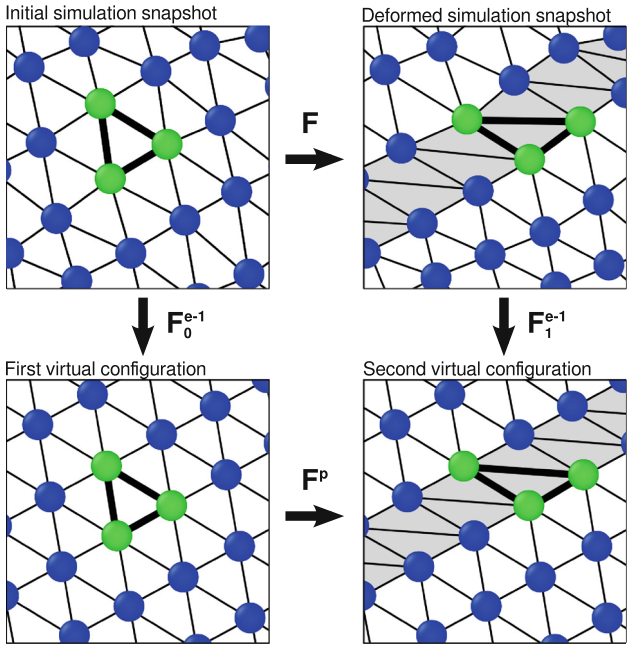

Fig. 3. Schematic illustration of the elastic–plastic decomposition scheme, $\mathbf{F} = \mathbf{F}_1\mathbf{F}^{\mathbf{p}}\mathbf{F}_0^{\mathbf{e}-1}$, and its calculation from atomistics using a Delaunay tessellation. Gray tessellation elements have experienced plastic deformation. Mapping both the initial and the deformed simulation snapshots to corresponding ideal, stress-free configurations yields the purely plastic deformation gradient $\mathbf{F}^{\mathbf{p}}$. Note that, in practice, all calculations are performed on a per tessellation element basis so that the two virtual configurations are never explicitly constructed.

Several aspects of this approach deserve further explanation (see Ref. 23 for a detailed discussion of the method). While the total deformation gradient **F** is an incremental quantity, because it is computed from two different snapshots of the same simulation, the elastic tensor $\mathbf{F}^{\mathbf{e}}$ is an instantaneous quantity associated with a single system state. It describes the inherent or residual elastic deformation of the crystal with respect to an *ad hoc*, ideal state constructed by the algorithm. In fact, the ideal crystal lattice never gets constructed explicitly. The algorithm only associates ideal lattice vectors with the neighbor bonds in the physical configuration. This effectively avoids problems due to dislocations, which locally render the elastic field incompatible. Furthermore, because the virtual reference lattice has a fixed orientation, the computed $\mathbf{F}^{\mathbf{e}}$ tensor carries information about the current crystal orientation at the atom's location in addition to the local elastic strain.

Defect cores, i.e., regions of the material where the atomic arrangement differs considerably from that of the perfect lattice, require special attention. Here, the perfect lattice is no longer a valid choice for a reference configuration, and an "ideal" defect state must be used instead. The rationale is that the computed elastic deformation must not include the plastic deformation associated with the creation of the defect, for instance a stacking fault left behind by a partial dislocation, but only the elastic distortion of the defect stemming from remote sources of stress. This poses a challenge: How do we pick the right local reference configuration for different regions in the simulation snapshot? The answer leads back to the first part of this article. Using an atomic structure identification method, we can determine the local structure in the vicinity of each atom and use the corresponding ideal template structure as reference to calculate the purely elastic deformation. The described method works only under the condition that the correct reference structure is included in the pattern catalog and the structure identification method can identify it. Note that we are making use of the fact that structure identification algorithms usually tolerate some elastic displacements. Open questions regarding the ambiguity of the ideal, stress-free reference state and our ability to practically determine it still must be addressed in future work.

Once the current elastic deformation gradient has been determined, the purely plastic deformation can be obtained from the relation $\mathbf{F^p = F^{e-1}F}$. In general, however, one cannot ignore residual elastic strains already present in the initial configuration of the atomistic system. Then, a threefold decomposition of the total deformation gradient,[23] $\mathbf{F = F_1^e F^p F_0^{e-1}}$, is appropriate, which takes into account the elastic lattice strains in both the initial and final configuration as illustrated in Fig. 3.

In amorphous materials and in regions of crystals having a highly disordered structure, for instance general high-angle grain boundaries (Fig. 4), it is usually impossible to map atoms to an ideal reference state and the elastic–plastic decomposition method cannot be applied. However, it is still possible to estimate the plastic strain localized in such disordered regions by subtracting elastic contributions (estimated from stress measurements, for instance) and the computed lattice slip-based strain from the total deformation of the sample. Such a quantitative approach has been used in a recent study to gain insight into the hierarchy of competing deformation mechanisms in nanocrystalline alloys.[26]

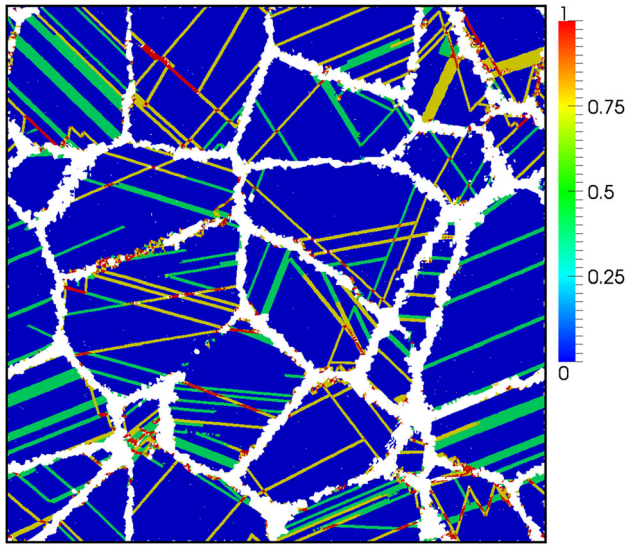

Fig. 4. Plastic *von Mises* shear strain field computed from a three-dimensional MD simulation of nanocrystalline Pd (15 nm grain size) after 11.4% uniaxial deformation. The integrated tensile strain generated by crystal slip processes (dislocations) in the grain interior is 6.7% and can be directly measured with the described method. Grain boundary shuffling and sliding processes (white regions) have contributed 2.8% plastic strain, and the remaining 1.9% strain is elastic (Color figure online).

AUTOMATED DISLOCATION IDENTIFICATION

Dislocation theory and derived simulation techniques such as discrete dislocation dynamics (DDD) explicitly treat dislocations as line objects that interact with stress fields, move, and undergo reactions according to appropriate rules.[27] However, important effects such as splitting of the dislocation core, nucleation, and interaction of dislocations with other defects may be difficult or impossible to incorporate in such models. If microscopic details of dislocations behavior are essential, then a fully atomistic description of the dislocated crystal becomes necessary.

However, while fully atomistic models do not put restrictions on the dislocations types and processes that can occur, it is harder for the observer to identify and keep track of dislocations as they are no longer explicit objects. In the past, atom-filtering techniques such as the centrosymmetry parameter[8] provided the only way to make dislocations visible in MD simulations by revealing atoms that are part of defect cores (Fig. 5a). No quantitative data on the type of dislocations (their Burgers vector), line length, or topology of the dislocation network are provided by such a visual approach.

Two different routes were pursued in recent years to identify dislocation lines in atomistic data sets and to reconstruct a geometric representation of these defects. The first line of work,[3,4,29] leading to the so-called dislocation extraction algorithm (DXA), makes use of the classical Burgers circuit test,[30] which involves constructing a closed path around a dislocation defect and mapping it to a perfect crystal lattice to compute the Burgers vector. Advantages of this approach are its robustness and generality: The Burgers circuit procedure is independent of structural details of the dislocation

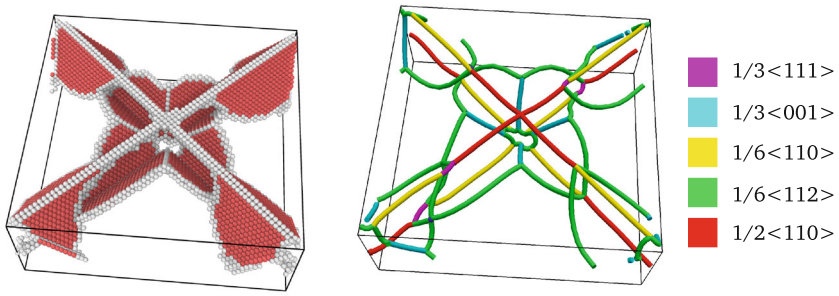

Fig. 5. DXA applied to a MD simulation of interfacial dislocations in a Cu/Ni interface[28] (Color figure online).

core, it always yields the true Burgers vector of a dislocation, and no previous knowledge of possible dislocation types is required. Figure 5 demonstrates how the DXA can identify different partial dislocations in fcc crystals. The second approach is based on metrics that are sensitive to the lattice misfit associated with dislocation cores.[31,32] By computing the curl of the elastic deformation gradient at an atomic site, the atomic Nye tensor can be obtained. After converting it to a continuum field using an interpolation scheme, numerical integration over an area normal to the dislocation yields the Burgers vector.[33] For dislocations with a compact core, a small integration area is sufficient to obtain a vector that is close to the true Burgers vector of the dislocation. Numerical errors, however, stemming from the calculation of the higher-order gradient of the elastic displacements and the numeric integration of the interpolated field, make a heuristic and lattice-specific method necessary to finally assign the most likely true Burgers vector to a dislocation segment, a step that may fail in some cases.

Automatically determining the Burgers vector of a dislocation in an atomistic crystal is only one part of the algorithmic problem. In addition, a line representation of the dislocation network needs to be generated. This can be achieved either through a skeletonization algorithm that reduces the network of atomic bonds inside the dislocation core to a linear chain,[34] or, alternatively, by first reconstructing the geometric boundary of the dislocation core, and then converting it to a linear curve using a moving-circuit procedure.[4] Attention must be paid when identifying nodes of the dislocation network, where multiple segments join. The DXA approach guarantees that the sum of Burgers vectors at a node is exactly zero as required by dislocation theory.

SURFACE GEOMETRY RECONSTRUCTION

The DXA described in the previous section can generate a smooth line representation of dislocation defects, which, in contrast to the original discrete crystal description, is suitable for determining dislocation densities, dislocation characters or the population of slip systems.[35,36] A similar argument motivates the development of a computational method that can yield a geometric representation of the external and internal surfaces of atomistic solids. Possible applications include measurements of internal surface area and solid volume fraction in atomistic simulations of materials with voids,[37] nanoporous gold structures,[38] or consolidation of nanoparticles.[39] Because surfaces can have great impact on, and, at the same time, are influenced by the deformation of such materials, monitoring the surface area, surface curvature distribution, and porosity are important analysis tools.

In a porous single crystal, the specific surface area per solid volume can be determined from atomic distribution functions.[40] An alternative method, which is based on Monte Carlo sampling, is the so-called "rolling ball" method.[41] It yields the surface area accessible to a probe sphere of a user-defined size. However, only algorithms that explicitly reconstruct the geometric surface shape can exactly measure both the surface area and the solid volume as well as the curvature gradients.[42]

An algorithm that solves this computational problem is implemented in our Crystal Analysis Tool code and makes use of the mathematical *alpha complex* concept.[43] Here, a Delaunay tessellation of the atomistic solid is generated to subdivide space into tetrahedral elements first. Each tetrahedral Delaunay element is subsequently classified as either *solid* or *empty* by computing the alpha complex. The alpha complex, being a subcomplex of the Delaunay tessellation, includes all Delaunay tetrahedra that have an empty circumscribing sphere with squared radius equal or smaller than a parameter α. Here "empty" means that the open sphere does not include any atoms. The parameter α controls the size of the imaginary probe sphere. Only space regions that are accessible to the probe are considered empty—and solid otherwise. The triangulated surface of the structure is constructed from the dividing

surface between solid and empty space regions, which consists of all triangular facets of the Delaunay tessellation adjacent to both a solid and an empty tetrahedron. In a final fairing step,[42] the triangulated surface mesh is smoothed to remove atomically sharp steps.

Figure 6 shows the results of the surface reconstruction method applied to a MD simulation of nanoporous gold. The parameter, which controls the level of detail reproduced by the constructed surface mesh, has been set to $\alpha = \alpha^2/2$ (corresponding to a probe-sphere radius equal to the nearest-neighbor distance of the gold crystal lattice) to resolve small features. The closed two-dimensional manifold produced by the algorithm provides a well-defined means to determine the surface area and solid volume of atomistic solids with complex topology. Figure 7 shows an analysis of the surface evolution of the nanoporous gold structure under deformation.

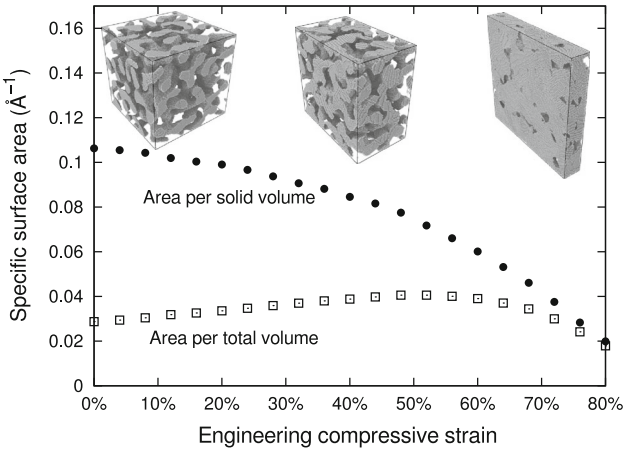

Fig. 7. Specific surface area of nanoporous gold as a function of uniaxial compression. The initial relative density is 30% with a ligament diameter of 3 nm. The surface area and solid volume have been determined with the method described in the text.

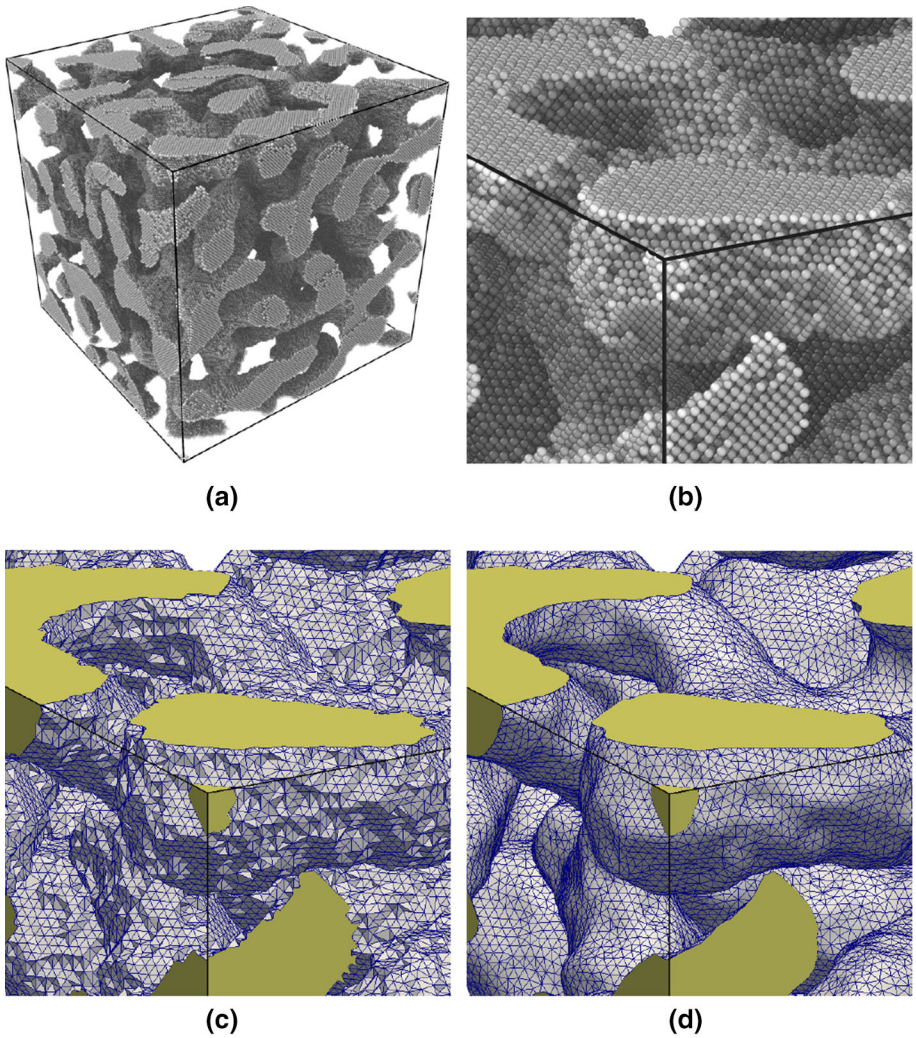

Fig. 6. (a) MD simulation of a nanoporous gold structure (3 nm ligament size). (b) Close-up view of the atomistic structure. (c) Triangulated surface mesh. (d) Smoothed surface mesh.

OUTLOOK

In this article, several novel computational analysis methods have been discussed that can be used in atomistic modeling of crystalline materials. The DXA is already rapidly establishing itself as a useful tool in many MD studies of crystal nanoplasticity. In addition to delivering quantitative data on the overall dislocation structure, it enables a detailed analysis of dislocation reaction processes at grain boundaries.[4] However, alongside with analysis algorithms like the DXA, tailored visualization tools have to be developed, which allow to simultaneously display, inspect, and interrelate the original atomistic data and the derived representations, to let users fully leverage such analysis capabilities.

The approaches discussed in this article to compute continuum metrics from atomistics and separate elastic and plastic strain may be applicable in future concurrent coupling simulation methods. Implementations of the quasicontinuum method,[44] for instance, often use a local mesh refinement criterion based on the total deformation of the material to control the automatic switching from a coarse to an atomistic description of the crystal. Being able to isolate elastic and plastic deformation at the atomic level could provide the basis for a local *coarsening* criterion in such models to enable the transition back to the coarse material description, which is computationally more efficient.

The discrete dislocation description generated by the DXA may prove useful in coupling atomistic models to DDD models.[45] The so-called coupled atomistics and discrete dislocations (CADD) method,[46] a multiscale simulation technique where dislocations are transferred from an atomistic domain into the continuum region and vice versa, has so far been implemented only in two dimensions. Detecting three-dimensional dislocations approaching the coupling interface may be achieved through use of the DXA. However, the reverse transition, generating an atomistic crystal description from an arbitrary dislocation configuration, is still an unsolved problem because atomistic dislocations core details need to be supplemented.

Several analysis techniques discussed in this article have been implemented in the program code Crystal Analysis Tool, which is available from the author upon request, and in the atomistic visualization and analysis software OVITO,[47] available at www.ovito.org.

REFERENCES

1. A.S. Keys, C.R. Iacovella, and S.C. Glotzer, *J. Comp. Phys.* 230, 6438 (2011).
2. C.L. Phillips and G.A. Voth, *Soft Matter* 9, 8552 (2013).
3. A. Stukowski and K. Albe, *Model. Simul. Mater. Sci.* 18, 085001 (2010).
4. A. Stukowski, V.V. Bulatov, and A. Arsenlis, *Model. Simul. Mater. Sci.* 20, 085007 (2012).
5. A. Stukowski, *Model. Simul. Mater. Sci.* 20, 045021 (2012).
6. J.D. Honeycutt and H.C. Andersen, *J. Phys. Chem.* 91, 4950 (1987).
7. G.J. Ackland and A.P. Jones, *Phys. Rev. B* 73, 054104 (2006).
8. C.L. Kelchner, S.J. Plimpton, and J.C. Hamilton, *Phys. Rev. B* 58, 11085 (1998).
9. T. Schablitzki, J. Rogal, and R. Drautz, *Model. Simul. Mater. Sci.* 21, 075008 (2013).
10. J.R. Ullmann, *J. ACM* 23, 31 (1976).
11. P. Erhart, J. Marian, and B. Sadigh, *Phys. Rev. B* 88, 024116 (2013).
12. B. Sadigh, P. Erhart, A. Stukowski, A. Caro, E. Martinez, and L. Zepeda-Ruiz, *Phys. Rev. B* 85, 184203 (2012).
13. J.A. Zimmerman, D.J. Bammann, and H. Gao, *Int. J. Solids Struct.* 46, 238 (2009).
14. P.M. Gullett, M.F. Horstemeyer, M.I. Baskes, and H. Fang, *Model. Simul. Mater. Sci.* 16, 015001 (2008).
15. M.F. Horstemeyer and M.I. Baskes, *MRS Proc.* 578, 1 (1999).
16. P.H. Mott, A.S. Argon, and U.W. Suter, *J. Comput. Phys.* 101, 140 (1992).
17. F. Shimizu, S. Ogata, and J. Li, *Mater. Trans.* 48, 2923 (2007).
18. G.J. Tucker, J.A. Zimmerman, and D.L. McDowell, *Model. Simul. Mater. Sci.* 18, 015002 (2010).
19. D. Sopu, Y. Ritter, H. Gleiter, and K. Albe, *Phys. Rev. B* 83, 100202 (2011).
20. J.A. Zimmerman, C.L. Kelchner, P.A. Klein, J.C. Hamilton, and S.M. Foiles, *Phys. Rev. Lett.* 87, 165507 (2001).
21. G.J. Wagner, R.E. Jones, J.A. Templeton, and M.L. Parks, *Comput. Method. Appl. M* 197, 3351 (2008).
22. J.A. Zimmerman, E.B. Webblll, J.J. Hoyt, R.E. Jones, P.A. Klein, and D.J. Bammann, *Model. Simul. Mater. Sci.* 12, S319 (2004).
23. A. Stukowski and A. Arsenlis, *Model. Simul. Mater. Sci.* 20, 035012 (2012).
24. E.H. Lee, *J. Appl. Mech.* 36, 1 (1969).
25. A. Stukowski, J. Markmann, J. Weissmüller, and K. Albe, *Acta Mater.* 57, 1648 (2009).
26. J. Schäfer, A. Stukowski, and K. Albe, *J. Appl. Phys.* 114, 143501 (2013).
27. V.V. Bulatov and W. Cai, *Computer Simulations of Dislocations* (Oxford University Press, 2006).
28. F. Akasheh, Private Communication (May 2013).
29. A. Stukowski and K. Albe, *Model. Simul. Mater. Sci.* 18, 025016 (2010).
30. F.C. Frank, *Philos. Mag. Ser. 7* 42, 809 (1951).
31. M. Elsey and B. Wirth, Paper presented at the Proceedings of the 8th International Conference on Computer Vision Theory and Applications, VISAPP'13, 2013.
32. C.S. Hartley and Y. Mishin, *Acta Mater.* 53, 1313 (2005).
33. C. Begau, J. Hua, and A. Hartmaier, *J. Mech. Phys. Solids* 60, 711 (2012).
34. C. Begau, A. Hartmaier, E.P. George, and G.M. Pharr, *Acta Mater.* 59, 934 (2011).
35. N. Naveen Kumar, R. Tewari, P.V. Durgaprasad, B.K. Dutta, and G.K. Dey, *Comp. Mater. Sci.* 77, 260 (2013).
36. A. Stukowski, K. Albe, and D. Farkas, *Phys. Rev. B* 82, 224103 (2010).
37. C.J. Ruestes, E.M. Bringa, A. Stukowski, J.F. Rodriguez Nieva, G. Bertolino, Y. Tang, and M.A. Meyers, *Scripta Mater.* 68, 817 (2013).
38. D. Farkas, A. Caro, E. Bringa, and D. Crowson, *Acta Mater.* 61, 3249 (2013).
39. D. Sopu, K. Albe, Y. Ritter, and H. Gleiter, *Appl. Phys. Lett.* 94, 191911 (2009).
40. J. Löffler and J. Weissmüller, *Phys. Rev. B* 52, 7076 (1995).
41. A. Shrake and J.A. Rupley, *J. Mol. Biol.* 79, 351 (1973).
42. J. Erlebacher and I. McCue, *Acta Mater.* 60, 6146 (2012).
43. H. Edelsbrunner and E.P. Mücke, *ACM Trans. Graphic* 13, 43 (1994).
44. E.B. Tadmor, M. Ortiz, and R. Phillips, *Philos. Mag. A* 73, 1529 (1996).

45. T. Junge, J.-F. Molinari, W.A. Curtin, and J. Song (Paper presented at the IV European Conference on Computational Mechanics, Paris, France, 16–21 May 2010).
46. L.E. Shilkrot, R.E. Miller, and W.A. Curtin, *Phys. Rev. Lett.* 89, 025501 (2002).
47. A. Stukowski, *Model. Simul. Mater. Sci.* 18, 015012 (2010).