Visualization and analysis of atomistic simulation data with OVITO–the Open Visualization Tool

To cite this article: Alexander Stukowski 2010 Modelling Simul. Mater. Sci. Eng. 18 015012

View the article online for updates and enhancements.

Related content
- Extracting dislocations and non-dislocation crystal defects from atomistic simulation data
  Alexander Stukowski and Karsten Albe
- AJAC: Atomic data calculation tool in Python
  Amani Tahat, Jordi Marti, Kaher Tahat et al.
- Visual physics analysis VISPA
  Oxana Actis, Michael Brodski, Martin Erdmann et al.

Recent citations
- Size of the Plastic Zone Produced by Nanoscratching
  Iyad Alabd Alhafez et al
- Tensile behavior of Cu 50 Zr 50 metallic glass nanowire with a B2 crystalline precipitate
  Matias Sepulveda-Macias et al
- Molecular dynamics simulation of bubble nucleation on nanostructure surface
  Yujie Chen et al
Visualization and analysis of atomistic simulation data with OVITO–the Open Visualization Tool

Alexander Stukowski

Institut für Materialwissenschaft, Technische Universität Darmstadt, Petersenstr. 23, D-64287 Darmstadt, Germany
E-mail: stukowski@mm.tu-darmstadt.de

Received 24 September 2009, in final form 4 November 2009
Published 15 December 2009
Online at stacks.iop.org/MSMSE/18/015012

Abstract
The Open Visualization Tool (OVITO) is a new 3D visualization software designed for post-processing atomistic data obtained from molecular dynamics or Monte Carlo simulations. Unique analysis, editing and animations functions are integrated into its easy-to-use graphical user interface. The software is written in object-oriented C++, controllable via Python scripts and easily extendable through a plug-in interface. It is distributed as open-source software and can be downloaded from the website http://ovito.sourceforge.net/.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Molecular dynamics (MD), molecular statics and Monte Carlo based simulations are nowadays standard methods for materials modeling with atomic scale resolution. Such atomistic simulations generate temporal-spatial information on three-dimensional atomic configurations and/or atom trajectories, which have to be analyzed in order to gain scientific insights into a physical system.

Thus, powerful visualization techniques play a key role, moreover, since the simulated systems become larger and more complex. The task of visualization software is to translate the raw atomic coordinates into a meaningful graphical representation to enable interpretation by the scientist. In many cases sophisticated analysis techniques can help to extract more essential information from the atomistic system. Again, it is the task of the visualization software to produce a meaningful depiction of this derived data.

The aim of this paper is to introduce a new, freely available visualization software called OVITO. Being a single integrated application it covers both the analysis and the visualization of large-scale atomistic datasets produced by molecular dynamics/statics and Monte Carlo simulation codes. Its name is an acronym for Open Visualization Tool, emphasizing that this software has been designed with flexibility and extensibility in mind. With this paper we
intend to make the scientific community acquainted with the capabilities of OVITO, which go beyond those of established visualization software packages [1–4].

2. Flexible processing pipeline

User demands on visualization tools go far beyond just displaying a large number of atomic spheres in space. State-of-the-art visualization packages should be flexible enough—and easy-to-use at the same time—to support the scientist in extracting the desired information from the simulation data in an interactive manner. OVITO is one of the first freely available programs that employs the powerful concept of a data processing pipeline to achieve this goal. Its underlying idea is as follows: from a pool of available visualization and analysis functions (building blocks) the user builds up a sequence of processing steps, which is applied to the atomistic input data. Each processing step in this sequence modifies the data coming from the previous step. The result of this processing pipeline is finally shown on the screen. Figure 1 illustrates how a typical processing pipeline to visualize the grain boundaries of a nanocrystalline microstructure might look in OVITO.

In contrast to a classical work-flow the pipeline concept allows one to alter the sequence of operations at any time, i.e. OVITO works in a non-destructive way with the input data. The user can change the parameters, the order of the processing steps or the input data whenever needed and the system will immediately re-evaluate the processing steps and update the resulting visualization display in realtime. Through intelligent data caching and parallelization techniques this all happens with a minimum usage of memory and processing resources.

The design of OVITO’s data model allows the user to associate an arbitrary number of data fields with each atom. These fields include, for instance, the atomic coordinates, types, velocities, energies, etc. The various processing steps introduced above operate on these data fields and/or calculate new fields on output. Moreover, the user can—based on mathematical expressions—define custom scalar, vector or tensor data fields for each atom. In the current version, OVITO provides functions for the following tasks:

• Coloring atoms based on their type, selection state or any other per-atom value stored in the input file or computed in the processing pipeline
• Transformation of atoms and the simulation cell
• Calculation and visualization of displacement vectors from the differences between two states of the system
• Interactive slicing and cutting of atomic structures
• Display of periodic images and wrapping of atoms at periodic boundaries
• Selection of atoms based on user-definable criteria
• Calculation and display of atomic bonds
• Ambient lighting calculation and shading of atoms [6] for better visualization of three-dimensional atomic structures
• Sophisticated analysis functions including
  – Common neighbor analysis [7]
  – Ackland–Jones analysis [8]
  – Cluster analysis
  – Coordination number calculation
  – Calculation of intrinsic and extrinsic atomic-level strain tensors [9, 10]

3. Additional features

Nowadays powerful graphics cards with high-performance 3D rendering capabilities have become available at low prices. Since this trend towards cheap graphics hardware is expected to continue, OVITO employs hardware-accelerated rendering based on the OpenGL standard to display a large number of atoms in realtime. Even with an entry-level graphics card OVITO is able to display several millions of atoms smoothly by using state-of-the-art rendering techniques such as shader-programs directly executed on the processor of the graphics card (GPU).

OVITO can handle multi-timestep data files, making it possible to visualize the system’s evolution with time. Every user operation can be undone and almost every parameter can be animated, making it even possible to create fly-through movie animations of the atomic structures. Furthermore, OVITO is able to display three-dimensional polygonal data and geometric shapes, allowing the user to enrich the atomic data with additional visual aids. For publication of simulation results OVITO can produce high-quality output images, movies and POV-Ray scene files [5].

OVITO has been developed as a platform independent software that runs on all major operating systems, including Microsoft Windows and Linux. Figure 2 shows a screenshot of the application’s main window.

The OVITO package has been designed with extensibility in mind since it cannot be assumed that the requirements of each and every user can be met in the first place. Therefore, the plug-in based architecture of OVITO enables the users to easily extend the software to their individual needs. Almost every aspect of OVITO can be extended by writing a small plug-in that seamlessly integrates into the program without making any changes to the core OVITO code base.

Data input and output are also part of the plug-in architecture. That is, the user can easily add import (and export) capabilities for the file format of his or her favorite atomistic simulation package to OVITO. In the current version, OVITO comes with import and export functions for the popular XYZ format [11] and several file formats used by the simulation packages LAMMPS [12], IMD [13] and VASP [14].

OVITO provides a rich graphical user interface that allows the user to view and analyze his or her simulation results interactively. In addition to an interactive visualization it is often necessary to process many simulation files in a row (batch processing). Therefore, OVITO features a scripting interface based on the Python language [15]. Most of OVITO’s functions can be invoked from Python scripts, which are run either within the graphical user interface or
Figure 2. Screenshot of the main window of OVITO. The window is divided into four viewports, each displaying the atomic structure from different viewing directions. The panel on the right allows the user to build up the processing pipeline and set the modification parameters.

from the console. Automated loading, batch processing and saving of simulation files allow one to integrate OVITO into a user’s custom tool chain.

4. Case study

In this section we demonstrate some of OVITO’s capabilities with a case study. Let us assume we want to study the microscopic deformation mechanisms of nanocrystalline palladium. To this end, we have performed a MD simulation of a tensile test experiment. The three-dimensional model structure being deformed is made up of 54 grains with an average size of 15 nm in a cubic simulation box with periodic boundary conditions. The simulated specimen is shown in figure 3(a) and contains approximately 6.2 million atoms, a system size that is handled by OVITO without difficulty on a standard consumer PC. Our MD simulation software has periodically dumped all atomic positions to a sequence of output files during the straining simulation.

The first thing to do is to make the grain boundaries of our bulk structure visible, as they play a crucial role for the plasticity of nanocrystalline materials. This can be achieved by performing a coordination analysis that determines the number of nearest neighbors of each atom. Palladium has a face-centered cubic structure, i.e. fully coordinated lattice atoms in the interior of the grains have exactly 12 nearest neighbors. Atoms, in contrast, that form the grain boundaries are usually mis-coordinated, that is, their coordination number deviates from 12, which gives us a simple criterion for filtering out grain boundary atoms.

To this end, we apply the Coordination Analysis operator to the input data. This operator takes one parameter, the nearest-neighbor cutoff distance and calculates the number of bonds of each atom. The results are stored in a new integer data field named Coordination.
now delete all fully coordinated atoms by first selecting them with the Select by Expression operator. This operator takes a Boolean expression that is evaluated for each atom. We set this expression to be ‘Coordination == 12’. Afterwards, all selected atoms are removed by the Delete Selected Atoms operator. Note that the atoms are not completely lost if we apply this operator. The delete operator only removes them from the final state of the system shown on the screen. We have the freedom to deactivate the delete operator again or to change any parameter of the preceding operators. OVITO will always re-evaluate the processing pipeline if needed.

Figure 3(b) shows the current state of the system after bulk crystalline atoms have been removed. The remaining mis-coordinated atoms form a grain boundary structure, but it is hardly visible due to the identical coloring of all atoms. This is a common visualization problem found for three-dimensional atomic structures. OVITO provides a remedy: the Ambient Lighting operator performs a lighting simulation to calculate a shading value for each atom, which makes the three-dimensional structure more apparent. Figure 3(c) displays the shaded grain boundaries. The screenshot (figure 2) shows OVITO’s user interface at this point of the case study. The current processing pipeline can be found in the upper right part of the main window.

Next, we want to take a closer look at grain boundary sliding, a plastic deformation mechanism that is found in nanocrystalline materials in addition to classical dislocation glide. When grain boundary sliding occurs, whole grains slide over each other to accommodate the macroscopic strain imposed on the sample. The sliding of complete grains at, say, 3% tensile strain can be made visible by calculating the displacement vectors of each atom. A displacement vector is the difference between the current deformed position of an atom and its initial position in the unstrained state. The Calculate Displacements operator performs this calculation by taking the differences between two atomic data files. The displacements calculated by the operator are stored for each atom in a new vector data field named
Displacement. Since we are only interested in the relative motion of neighboring grains, and not in displacements caused by the overall macroscopic strain, we first have to remove the macroscopic strain from the simulation box before calculating the displacement vectors. This is done by scaling the deformed simulation box, including all atomic positions, back to its initial shape using the Affine Transformation operator. Its key parameter is a $3 \times 3$ transformation matrix that is applied to all atomic positions.

The calculated displacement vectors are now ready for visualization. A first option is to color each atom according to one component of its displacement vector. If the vectors in one grain all point in the same direction, and in a neighboring grain they all point in the opposite direction, then sliding must have occurred at the grain boundary and will appear as a sharp contrast in the coloring of atoms. This can be seen in figure 3(d) where atoms have been shaded according to the $Y$ component of the displacement vectors (tensile axis was $Z$). Here, we have used the Slice operator to cut out a slice from the structure, followed by the Color Coding operator, which assigns colors (or shades of gray) to each atom based on the values of an arbitrary atomic data field.

Alternatively, OVITO can directly visualize the displacement vectors as arrows. Figure 3(e) displays a close-up view of a grain boundary with arrows indicating the relative motion of atoms in the adjacent grains.

5. Availability

To make OVITO available to as many researchers as possible it has been released as open-source software under the GNU General Public License. Thus, OVITO can be used free of charge, everyone can contribute to the software, extend it to his own needs and share newly developed plug-ins with other users. The C++ source code of OVITO can be downloaded from the website http://ovito.sourceforge.net/.

Acknowledgments

The author acknowledges the financial support of the Deutsche Forschungsgemeinschaft (FOR714).

References

[4] Avizo® visualization software (VSG, Visualization Science Group)