Summary

Nanoparticles have attracted continued interest in academia and industry over the last few decades due to their remarkable properties that differ from the same materials in bulk. These properties are dependent on not only the size of the nanoparticles but also their shape. It is thus of great importance for nanoscientists to be able to predict the shape of nanoparticles of different materials and in different environments (Marks & Peng, 2016).

In the continuum (large particle) limit, the equilibrium shape of nanoparticles can be determined with the so-called Wulff construction (Wulff, 1901). Wulff’s theorem states that the distance from the center of the nanoparticle to a facet is proportional to the surface energy of that facet. The equilibrium shape, often referred to as the Wulff shape, can thus be determined provided that the orientation-dependent surface energy is known.

The Wulff construction has been generalized to nanoparticles of icosahedral and decahedral geometry (Marks, 1983) as well as nanoparticles on surfaces (so-called Winterbottom constructions (Winterbottom, 1967)). The regular Wulff construction has been implemented in several software packages, including a submodule of the Python package pymatgen (Ong et al., 2013), a no longer maintained C++ package (Roosen, McCormack, & Carter, 1998), and a Wolfram Mathematica implementation with a graphical user interface (Zucker, Chatain, Dahmen, Hagège, & Carter, 2012). While the latter code has support for Winterbottom constructions, we have found no publicly available software that implements the icosahedral and decahedral Wulff construction. The aforementioned codes also seem to lack the ability to transform the created shapes into an atomistic representation, i.e., a nanoparticle of the Wulff shape consisting of atoms in a crystal structure, a feature of critical importance if the Wulff construction is to be used for atomistic simulations.

WulffPack is a Python package that carries out the Wulff construction and its generalizations using an efficient algorithm based on calculation of the convex hull (Barber, Dobkin, Dobkin, & Huhdanpaa, 1996) of the vertices of the dual of the Wulff polyhedron (Roosen et al., 1998; Virtanen et al., 2019). The user provides surface energies and crystal symmetry and WulffPack returns a versatile object that, at its core, contains the coordinates of the Wulff shape. Extraction of symmetry operations is handled internally with spglib (Togo & Tanaka, 2018). WulffPack includes functionality for visualizing the constructed shapes using Matplotlib (Hunter, 2007) (Fig. 1). There are also functions for analyzing the constructed shape, most notably in terms of area fraction of symmetrically inequivalent facets. This quantity is important in applications where properties of the material are facet-dependent, such as in catalysis. Finally, using the Atomic Simulation Environment (Larsen et al., 2017), an atomistic representation of the Wulff shape can also be extracted.

An extensive user guide including a documentation of the API is available at http://wulffpack.materialsmodeling.org/.
Figure 1: Truncated octahedron (left), decahedron (middle) and icosahedron (right), as visualized in WulffPack.

Acknowledgements

This work was funded by the Knut and Alice Wallenberg Foundation and the Swedish Research Council.

References


