Surfaxe: Systematic surface calculations

Katarina Brlec[1, 2], Daniel W. Davies[1, 2], and David O. Scanlon[1, 2, 3]

1 Department of Chemistry, University College London, 20 Gordon Street, London WC1H 0AJ, United Kingdom **2** Thomas Young Centre, University College London, Gower Street, London WC1E 6BT, United Kingdom **3** Diamond Light Source Ltd., Diamond House, Harwell Science and Innovation Campus, Didcot, Oxfordshire OX11 0DE, UK

Summary

Surface science is key to understanding the properties of a wide range of materials for energy applications, from catalysts to solar cells to battery components. Computational modelling based on quantum mechanics is often used to calculate surface properties of materials, which in turn determine their stability and performance. The maturity of these "first-principles" methods, coupled with the huge amount of computational power accessible today, means they can now be used predictively in high-throughput screening workflows to suggest new materials for specific applications before they are synthesised. The `surfaxe` package provides a framework for such screening workflows, automating each stage of the process.

Statement of need

Accurate descriptions of electronic structure are needed to calculate surface properties including surface formation energies, adsorption energies and absolute electron energies (ionisation potential, electron affinity, and work function). However, surface calculations diverge significantly from the typical setup in periodic codes where the bulk crystal is described by repeat boundary conditions in all three dimensions. To reveal a surface, the bulk must be cleaved into a slab and periodicity reduced to just two dimensions. Additional parameters must then be considered such as variance in slab thickness, vacuum size, surface termination (dangling bonds), and net electrostatic dipole moments, all of which complicate the calculation workflow and make reliable determination of properties more difficult.
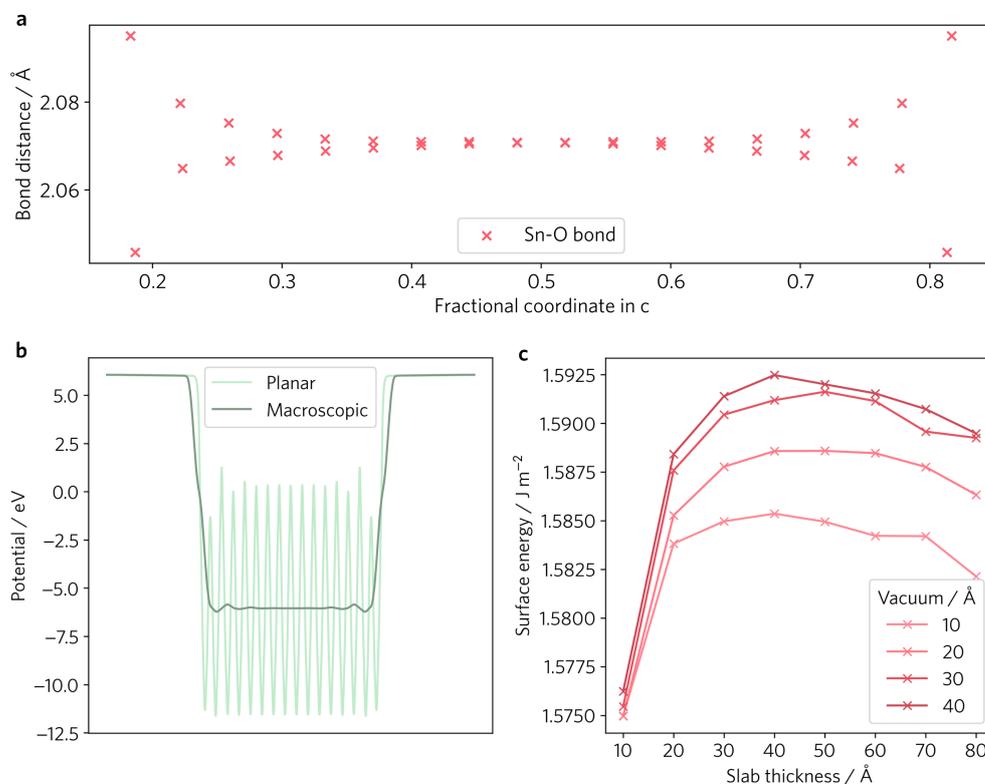
Surfaxe

The aims of `surfaxe` are:

- To act as a framework for the automation of surface calculations, with particular emphasis on ensuring that properties are converged with respect to the additional parameters that are introduced compared to bulk calculations
- To increase the efficiency and reproducibility of surface calculations by automating the generation of input files and processing of output files for density functional theory (DFT) codes
- To provide a toolbox of intuitive analytical tools to calculate performance-critical materials properties and directly generate publication-quality plots

The code makes extensive use of existing Python Materials Genomics (pymatgen) (Ong et al., 2013) surface modules with full functionality retained in `surfaxe`. As well as a fully flexible Python API, `surfaxe` has a lightweight command line interface. The modularity of `surfaxe` closely follows a best-practice workflow for the calculation of surface properties, with key features including:

- Automatic cleaving of slabs from the bulk crystal and organising them into a directory structure with all necessary calculation input files – generation module
- Analyses of atomic displacements and coordination environments, bond lengths and electrostatic potential through the slab (Figure 1a and b) – analysis module
- Processing of raw DFT outputs to determine surface energy variation with slab and vacuum thickness (Figure 1c) – convergence module
- Automatic extraction of surface energy, vacuum and core energy levels, along with the important calculation parameters – data module

In addition to pymatgen, existing packages related to surface calculations include the Atomic Simulation Environment (ase) (Hjorth Larsen et al., 2017), which is a large materials informatics library, and smaller packages to aid with specific post-processing tasks: MacroDensity for plotting of potentials (Butler, 2015), WulffPack for plotting of Wulff shapes (Rahm & Erhart, 2020), and bapt for plotting band alignments (Ganose, 2017). While these toolkits are extremely useful, `surfaxe` is distinct with its focus on the rigorous convergence of properties, the enabling of reproducible workflows, and the production of processed datasets and plots at the command line. Lastly, `surfaxe` is built on the pymatgen ecosystem, so full integration with the workflow packages FireWorks (Jain et al., 2015) and AiiDA (Uhrin et al., 2021) is possible for managing calculations on high-performance computing clusters.

Figure 1: Example analysis: a) average bond length, b) electrostatic potential as a function of lattice parameter perpendicular to the surface, and c) a typical surface energy convergence plot with respect to slab and vacuum thickness.

Acknowledgements

References

Butler, K. T. (2015). *MacroDensity*. http://www.github.com/keeeto/macrodensity

Ganose, A. M. (2017). *BAPT*. http://www.github.com/utf/bapt

Hjorth Larsen, A., Jørgen Mortensen, J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M., Friis, J., Groves, M. N., Hammer, B., Hargus, C., Hermes, E. D., Jennings, P. C., Bjerre Jensen, P., Kermode, J., Kitchin, J. R., Leonhard Kolsbjerg, E., Kubal, J., Kaasbjerg, K., Lysgaard, S., … Jacobsen, K. W. (2017). The atomic simulation environment—a Python library for working with atoms. *J. Phys. Condens. Matter*, *29*(27), 273002. https://doi.org/10.1088/1361-648X/aa680e

Jain, A., Ong, S. P., Chen, W., Medasani, B., Qu, X., Kocher, M., Brafman, M., Petretto, G., Rignanese, G.-M., Hautier, G., Gunter, D., & Persson, K. A. (2015). FireWorks: A dynamic workflow system designed for high-throughput applications. *Concurrency and Computation: Practice and Experience*, *27*(17), 5037–5059. https://doi.org/10.1002/cpe.3505

Ong, S. P., Richards, W. D., Jain, A., Hautier, G., Kocher, M., Cholia, S., Gunter, D., Chevrier, V. L., Persson, K. A., & Ceder, G. (2013). Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Comput. Mater. Sci.*, *68*, 314–319. https://doi.org/10.1016/j.commatsci.2012.10.028

Rahm, J., & Erhart, P. (2020). WulffPack: A Python package for Wulff constructions. *J. Open Source Softw.*, *5*(45), 1944. https://doi.org/10.21105/joss.01944

Uhrin, M., Huber, S. P., Yu, J., Marzari, N., & Pizzi, G. (2021). Workflows in AiiDA: Engineering a high-throughput, event-based engine for robust and modular computational workflows. *Comput. Mater. Sci.*, *187*, 110086. https://doi.org/10.1016/j.commatsci.2020.110086