py-sc-fermi: self-consistent Fermi energies and defect concentrations from electronic structure calculations

Alexander G. Squires 1,4, David O. Scanlon 1,3,4, and Benjamin J. Morgan 2,4

1 Department of Chemistry, University College London, London, United Kingdom 2 Department of Chemistry, University of Bath, Bath, United Kingdom 3 Thomas Young Centre, University College London, London, United Kingdom 4 The Faraday Institution, Didcot, United Kingdom

DOI: 10.21105/joss.04962

Summary

py-sc-fermi is a Python package for calculating point-defect concentrations in crystalline materials, under the constraint of net-charge-neutrality and the assumption of thermodynamic (quasi-)equilibrium. The required inputs are the formation energies of all point defects of interest and the electronic density of states. These can be obtained from electronic structure calculations, e.g., density functional theory.

Point defects are atomic-scale imperfections in crystalline materials. They can strongly affect a range of material properties, including electronic structure (Pastor et al., 2022), charge transport (Maier, 2015), and energy-conversion processes (Kim et al., 2020). Point defect concentrations are generally sensitive to synthesis conditions and to the inclusion of extrinsic dopants. By understanding the thermodynamics of point-defect formation, it is possible to predict the response of specific material properties to changes in synthesis or doping protocols (Park et al., 2018; Qu et al., 2022; Squires et al., 2019). The quantitative modelling of point-defect populations in functional materials has become common practice in the materials modelling community, often undertaken with the aim of informing the design of novel, highly performant materials (Ganose et al., 2018; Jackson et al., 2022; Toriyama et al., 2021) or to rationalise and optimise the properties of known materials (Kim et al., 2018; Shimoda et al., 2022; Squires et al., 2022).

While the formation energies of charge-neutral point defects can be calculated simply, e.g., from electronic structure calculations, the formation energies of charged defects depend on the electronic chemical potential (the Fermi energy), which, in turn, depends on the net population of all charged defects in the system. The equilibrium concentrations of charged point defects are therefore mutually dependent, and must be solved for as a unified self-consistent system.

In practice, self-consistent defect concentrations are calculated by imposing the constraint of net-charge-neutrality (Buckeridge, 2019),

\[ 0 = \sum_{X,q} q[X,q] - n_0 + p_0. \] (1)

Here, the first term is a sum over charge contributions from each defect \( X \) in each of its accessible charge states \( q \), and \( n_0 \) and \( p_0 \) are the concentrations of free electrons and electron holes respectively.

py-sc-fermi provides a Python implementation of an iterative numerical approach for calculating the self-consistent solution for an arbitrary set of point-defects. The resulting self-consistent Fermi energy can be used as a general descriptor for the electronic-transport properties of a material (Jackson et al., 2022)\textemdash;the calculated electronic charge-carrier concentrations can be used to calculate electronic conductivities (Squires et al., 2022)\textemdash;and the

point-defect concentrations, and how these vary with synthesis conditions or doping protocol, can be used to model how the formation of competing defects affects material properties of interest (Ganose et al., 2018; Kavanagh et al., 2021) as well as quantifying the doping response of a material (Squires et al., 2019, 2021).

Statement of need

**py-sc-fermi** is a Python package for determining the self-consistent Fermi energy of a material from pre-calculated point-defect data. This code allows the quantification of point defect and electronic carrier concentrations. Other software exists that is written to perform similar calculations (Arrigoni & Madsen, 2021; Buckeridge, 2019; Neilson & Murphy, 2022; Ogawa et al., 2022). To the best of our knowledge **py-sc-fermi** is unique in providing all of the following features:

- It is built on a flexible Python API which allows for rapid prototyping and convergence testing of the solved-for Fermi energy.
- It is not part of a larger “point-defects workflow” package, making it as flexible as possible for the end user.
- It is agnostic towards the choice of simulation code used to generate the input data.
- It possess both command-line functionality for those with little Python experience and an object-oriented API that allows the user to easily construct objects representing individual defects or complete systems of defects.
- It allows for constraints on the concentrations of any combination of defects and specific charge states of that defect, including the simulation of the influence of effective dopants.
- It is fully documented with good unit-test coverage.

The code has been used in a number of studies, including some focussed on rationalising the properties of known materials (Squires et al., 2022, 2021) and others focussed on predicting the properties of novel materials (Jackson et al., 2022). The code can also assist with visualising defect energetics (Haouari et al., 2021).

One notable feature of **py-sc-fermi** is the ability to fix the concentrations of specific defects or individual defect charge states. Defect and electronic carrier concentrations are functions of temperature. The initial defect concentrations in a material are often determined by high-temperature synthesis conditions. Subsequent rapid cooling can result in some or all of these defect populations being “frozen in” (kinetically trapped) at their initial high-temperature values, while other defect populations remain free to vary, and establish new pseudo-equilibrium defect concentrations (Maier, 2003). **py-sc-fermi** allows this scenario to be modelled through a two-step calculation, where in the second step some defect concentrations are fixed to their high temperature “synthesis” values. The new pseudo-equilibrium defect concentrations can be quickly computed as a function of quenching temperature, allowing defect concentrations to be calculated under a range of synthesis and quench scenarios (Squires et al., 2022).

There is some minor convenience functionality for users of the Vienna Ab initio Software Package (VASP).

Acknowledgements

The authors are grateful for feature requests and user testing from Seán Kavanagh, Joe Willis, Jiayi Cen, Sabrine Hachmioune, and Lavan Ganeshkumar. This work was supported by the Faraday Institution grant number FIRG017. D.O.S. acknowledges support from the European Research Council, ERC (Grant 758345). B.J.M. acknowledges support from the Royal Society (UF13032 and URF\\191006).
References


