Extended software development workshop (WP2: electronic structure): Wannier90

12 – 16 September 2016, San Sebastian, Spain

Organizers: Arash A. Mostofi (Imperial College London, UK), Mike Payne (University of Cambridge UK)

1. State of the art

Wannier90 is a code for generating maximally-localized Wannier functions and using them to compute with high efficiency and accuracy a host of advanced materials properties. Wannier90 is a paradigmatic example of an interoperable software tool, achieved by ensuring that the quantities that need to be input into it are entirely independent of the underlying electronic structure code from which they are obtained.

All of the major electronic structure codes in the world have an interface to Wannier90, including Quantum-Espresso, AbInit, VASP, Siesta, Wein2k and Fleur. In this way, the developments that are released in Wannier90 are immediately available to a very large fraction of the electronic structure research community and the impact of this has been evident in the large number of publications resulting from the use of Wannier90 (around 900), many of them in high profile journals such as Nature on topics ranging from the strongly correlated physics of topological insulators and iron-based pnictide superconductors, to the chemistry and bonding of ionic liquids.

The aim of this workshop is to share recent developments related to the generation and use of maximally-localised Wannier functions and to either implement these developments in, or interface them to, the Wannier90 code. It is also an opportunity to improve and update existing interfaces to other codes and write new ones. The format was deliberately open, with the majority of the time allocated for coding and discussion.

This workshop was an activity of the E-CAM European Centre of Excellence (https://www.e-cam2020.eu) that supports Software, Training and Discussion in Simulation and Modeling, and MARVEL, the Swiss National Centre on Computational Design and Discover of Novel Materials (http://nccr-marvel.ch).

2. Training provided

This workshop was instrumental in catalysing the transition of Wannier90 from a code developed by a small handful of developers to a community code with a much wider developer base. This has been achieved in two principal ways through the workshop: (i) situating the source code and associated development efforts on a public GitHub repository (wannier-developers/wannier90); and (ii) building a community of connected Wannier90 developers by facilitating new and hopefully lasting personal interactions between individuals at the workshop. There were 25 attendees at the workshop.

The response to this workshop from the community was highly positive. Aside from the Wannier90 Developers Group, there were 20 attendees at the Workshop, actively participating and contributing to discussions and the code. An anonymous feedback form that they were asked to complete at the end of the workshop demonstrated its success. Of the 14 respondents: 93% agreed or strongly agreed that the workshop was useful; 96% said that it should be repeated either once a year or every other year; and 100% agreed or strongly agreed that it was a good occasion to meet with other researchers and that it was a good opportunity to learn something new. Further to the feedback of the participants of the workshop, our intention is to hold a community Wannier90 software development workshop once every two years, with the next one in the second half of 2018.

3. Software development projects

The major technical outcomes were as follows:

Migration of the Wannier90 repository to GitHub: the Wannier90 revision control repository was previously hosted on a server hosted by the Theory of Condensed Matter group in Cambridge. This was migrated to GitHub in order to make it much easier for the community to contribute developments. The GitHub repository can be found at https://github.com/wannier-developers/wannier90.

A Guide for Contributors: now that the code has a much wider developer base, a formal coding style guide and guide for developers is needed.

An interface to the Yambo code: Wwrite an interface between Yambo and Wannier90, taking into account the possibility that eigenvalues need to be resorted if GW corrections swap them.

Update to the interface to Quantum-Espresso (non-collinear spin + USPPs): add non-collinear support with ultrasoft pseudopotentials to the QE interface (pw2wannier90).

AiiDA plugin for Wannier90: development of a plugin for AiiDA to be able to automate Wannier90 runs.

SIESTA interface: Wannier90 Interface for the SIESTA code.

CP2K interface: Wannier90 Interface for the CP2K code.

Matrix elements of the position operator: output the matrix elements of the position operator between Wannier functions in a format similar to the one for the Hamiltonian (seedname_hr.dat file).

Gollum interface: Wannier90 Interface for the Gollum code.

Improve the current library mode to support new features: improve the current library mode to support new features such as spinors and symmetry-adapted Wannier functions.

Design a new library interface: major re-design a new library interface for enhanced usability.

Add support for automatic output of bibTeX files: add functionality in the Wannier90 code so that at the end of each run it produces a bibTeX .bib file with all the citations to be used according to the functionality used in the run.

Symmetry-adapted Wannier functions: implement the code to compute the symmetry-adapted Wannier functions.

Improvements to the interpolation routines: improve the interpolation routines (bands and operator interpolation) to shift WFs into the Wigner-Seitz cell centred on the other WF when computing matrix elements.

Test suite for Wannier90 and integration with Travis-CI: implement the infrastructure for having tests in Wannier90, integrate it both with the test farms used for the EPW code (nightly builds) and with Travis-CI.

Z2PACK interface: implement the necessary changes in Wannier90 to allow Z2PACK to be able to use the Wannier90 routines.

FORD infrastructure implementation for automatic code documentation : implement the core infrastructure to have FORD automatically document the Wannier90 Fortran codes and routines.

Some of the software modules developed during this workshop will also be available in the E-CAM repository for electronic structure. These modules will be part of deliverable 2.2 of WP2.