



D2.2

QE and SIESTA workflows for basic materials parameters

Daniele Tomerini, Alberto García, Pablo Ordejón, and Nicola Marzari

Document information

Project acronym:	INTERSECT
Project full title:	Interoperable Material-to-Device simulation box for disruptive electronics
Research Action Project type:	Accelerating the uptake of materials modelling software (IA)
EC Grant agreement no.:	814487
Project starting / end date:	1 st January 2019 (M1) / 31 st January 2022 (M37)
Website:	www.intersect-project.eu
Final version:	30/01/2020
Deliverable No.:	D2.2
Responsible participant:	EPFL (participant number 2)
Due date of deliverable:	31/01/2020
Actual submission date:	30/01/2020
Dissemination level:	PU - Public

Authors: Daniele Tomerini, Alberto García, Pablo Ordejón, and Nicola Marzari

To be cited as: D. Tomerini, A. García, P. Ordejón, and N. Marzari (2020): QE and SIESTA workflows for basic materials parameters. Deliverable D2.2 of the H2020 project INTERSECT (final version as of 30/01/2020). EC grant agreement no: 814487, EPFL, Lausanne, Switzerland.

Disclaimer:

This document's contents are not intended to replace consultation of any applicable legal sources or the necessary advice of a legal expert, where appropriate. All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user, therefore, uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors' view.

D2.2 QE and SIESTA workflows for basic materials parameters

Content

1 Executive Summary	4
2 Introduction	4
3 Workflows in AiiDA	5
3.1 AiiDA calculations	5
3.2 AiiDA work chains	6
3.3 AiiDA-Quantumespresso workflows	8
3.3.1 Band gap workchain	9
3.4 AiiDA-SIESTA workflows	10
3.4.1 Band Gap workchain	11
4 Other on-demand properties	11
References	11

1 Executive Summary

Materials properties in the context of the Interoperable Material-to-Device simulation box are either retrieved from external sources (database of properties) or calculated on-demand, through workflows handled by AiiDA, that manages the computations performed by Quantum ESPRESSO and SIESTA.

In this document, we detail the difference between calculation and workchains in AiiDA and their role and interconnection in the context of workflows for materials properties. The available workflows for materials properties are reported. In particular, we present the hierarchical structure of the main workchain for basic electronic structure calculations (e.g. atomic relaxation, scf, bandstructure, etc) and we focus on the workflow for the extraction of the direct/indirect bandgap of solid-state materials.

2 Introduction

The realization of robust, reliable, and fully automated software solution for materials-to-device simulations is based on the implementation of workflows for the calculation of materials' properties on-demand and for a back-and-forth data exchange from/to electronic and continuum model. Here, we exploit the capability of the AiiDA infrastructure to automate the input/output of Quantum ESPRESSO and SIESTA for DFT calculations. This task relies on the definition of algorithms to setup the workflows and automatically choose the physical parameters for each case and initiate data-on demand. The most relevant interexchange physical quantities includes bandgaps, workfunctions, effective masses, dielectric constants, formation enthalpies, energy-level distribution of defects and their aggregation, electrical and thermal conductivities, inelastic/dissipation scattering rates (see also deliverable D1.1). All these will be necessary for the implementation of the test and pilot cases proposed in the project. In this deliverable, we focus on the determination of the *basic* materials parameters (such as bandgap, band structure, effective mass, dielectric constant). Other advanced properties (e.g. defect formation enthalpy, diffusion barrier, thermal conductivity, etc) will be subject of deliverable D2.3 due at month M24.

The automatic determination of each of these materials parameters requires the implementation of a specific workflow. In this first stage of the implementation, we focused on the definition and the setup of the main hierarchical structure of the AiiDA workchains for the main electronic structure calculations and we report the status of the developed workflows for the calculation of basic materials properties as implemented in both Quantum ESPRESSO and SIESTA codes.

3 Workflows in AiiDA

3.1 AiiDA calculations

One of the main AiiDA [1] capabilities is to simplify life for materials modelers: it provides a platform to easily use simulation codes to perform calculations, and automatically create and store the full provenance of the results, by linking every step along the process in a logical and reproducible way.

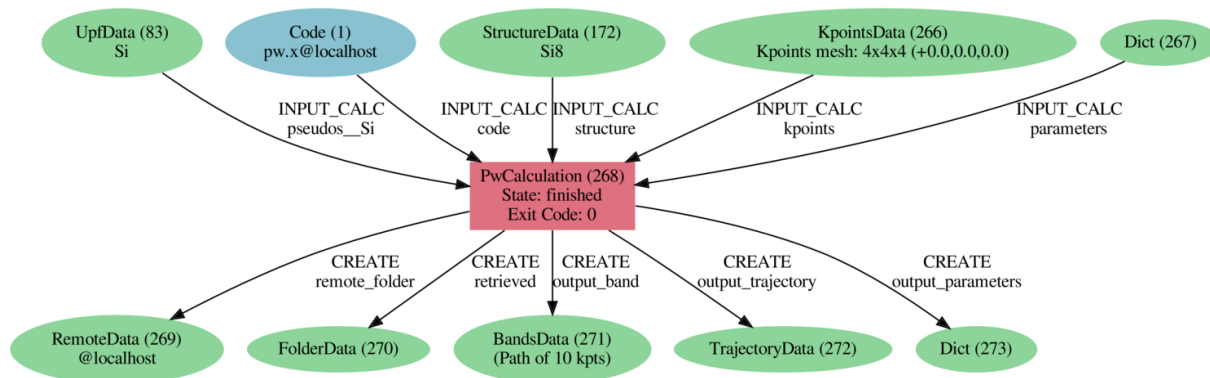


Figure 1. Example of the connectivity between the AiiDA database node for a single Quantum ESPRESSO calculation, from the documentation.

The support for external codes is provided by python packages that extend the functionality of AiiDA creating a “plugin ecosystem”.

A plugin can, therefore, support a simulation code, by defining how to generally create the text input file it needs to run from AiiDA objects. Figure 1 shows an example of interconnectivity between AiiDA and Quantum ESPRESSO (a similar scheme can be drawn for AiiDA-SIESTA plugin). In the example of Figure 1, the inputs are an external code, a crystal structure, an array of reciprocal k-points, pseudopotential files and a dictionary of parameters. The AiiDA plugin registry [2] contains a public list of codes that are supported (44 as of January 2020), with the possibility for developers to add their own (see also deliverable D2.1).

Since computationally demanding calculations (e.g. DFT cycles) are typically executed on external, dedicated computing centers, AiiDA takes care of the preparation of the submission scripts to submit a job on an external computer, and of the communication and transfer of

files; the most common schedulers (e.g. slurm, SGE, Torque) and communication mechanisms are available, but it is possible to extend them as needed.

The code plugin takes care of uploading the files it generated to execute the calculation, while AiiDA handles the rest (including the preparation of a submission script, according to the type of computing resource configured in advance), and wait for its execution.

The process node, e.g. the red node 268 of the database in the example of Figure 1, contains the details of the calculation, including its status, e.g. if it is waiting to be executed, running, or if it has failed for any reason -- with a related error code to meaningfully represent it. In AiiDA this is called a *calculation job*, that can either be run interactively in a python shell, but will more likely be submitted on a dedicated computer and monitored.

Once the calculation is terminated, AiiDA retains the information on the whereabouts of the calculation execution (*RemoteData* node 269 in Figure 1), the raw unprocessed results that can be inspected (*FolderData*, a repository that contains all the relevant files to be copied and saved locally, as per instruction of the plugin), and *Nodes* that contained parsed information from the raw outputs into AiiDA classes (Nodes 271 to 273 in Figure 1).

The links between the database nodes (*inputcalc* from the inputs to the calculation node, and *create* from the calculation to the results) are automatically created and stored in the database.

Calculations can also be created as generalized python subroutines, called **calculation functions**, that can be executed in a script, without having to call an external code. In this case, the inputs of the subroutine would be transformed inputs of a calculation node in the database, while any returned node will be saved as a result.

The graph describing the relation between nodes through calculation jobs and calculation function is strictly acyclic: there cannot be any closed loops, i.e. a node cannot be at the same time an *output* of a calculation and one of its *inputs*.

A python script that includes calculation function and calculation jobs can be used to generate automated material science workflows to handle as many steps as needed, integrating different codes, leaving AiiDA to handle the interconnections between the inputs and the generated data.

3.2 AiiDA work chains

Calculation functions can be as complicated and long as desired/necessary. In any case, they require a python script (either a script or a shell) to be run and executed on the machine where AiiDA is installed. This means that if -- for any reason -- the script is interrupted, the

progress on the calculation workflow would be lost. This could be a serious drawback for the automated execution of the entire workflow.

In order to overcome this problem, AiiDA implements the *workchain* class [3], so to gracefully tackle multi-step processes. Each one of the steps, along with the expected inputs and outputs, is defined in the workchain “define” python method. Along with sequential logical steps, the workchain implements logical handling (conditionals, *while* loops) and variables can be saved and passed between different steps in the work chain context, e.g. to keep track of the status of launched subprocesses, or to checkpoint important information.

When the workchain is submitted, an AiiDA daemon (a background process) takes care of it and frees the python script that launched it. At the end of each workchain step, the state of the calculation is saved on the machine where AiiDA is executed. If the machine is powered down (for any reason), the workchain can resume its status as soon as the AiiDA daemon is restarted. The status of the work chain can be inspected at any moment with the AiiDA *verdi* commands.

Workchain complements the AiiDA graph obtained solely by provenance (the created database links) by adding an additional logical on-top layer, which is called *return links*. Workflows cannot generate new nodes in the graph; rather the outputs of a workflow can only be nodes of the graph that already exist in the database. The workflow connects a series of low-level tasks (e.g. calculation jobs), leaving the management of node creation to the calculation functions, but keeping trace of the hierarchical connections that generated them. In this way, a user can either decide to focus on the full provenance of a workflow (Figure 2), or hide its complexity, allowing to highlight only the higher-level decisions (see Figure 3) or the data provenance (Figure 4).

A work function can *call* any type of process including other work functions; this allows the generation of complex workflows by building on lower level workchain that focuses on specific tasks.

The AiiDA documentation and tutorials provide more detailed information about the different process types and their implementation.

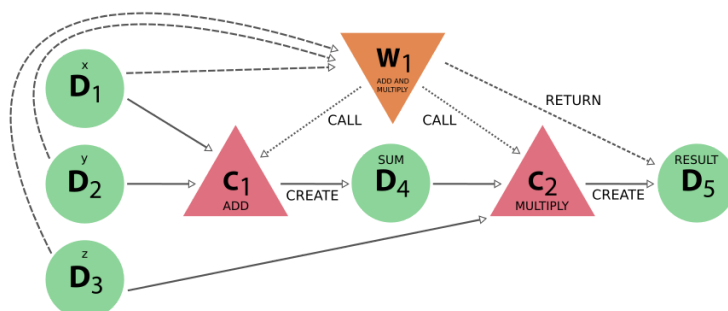


Figure 2. A workflow that takes 2 integers (D1 and D2), sums them and multiplies them by a third number (D3). C1 and C2 are calculation functions that add and multiply 2 numbers, respectively. Note that the work function W1 calls the C2, which creates the node that is returned by the work function. From the AiiDA documentation.

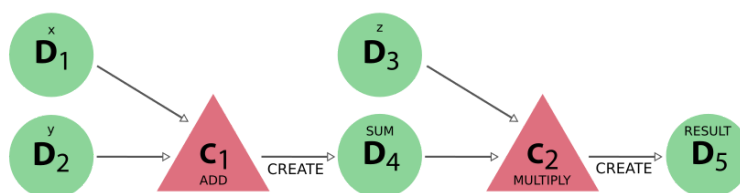


Figure 3. Data provenance of the work function W1 in Figure 2. The links from the work chain inputs, calls, and outputs have been hidden, leaving only the acyclic graph. From AiiDA documentation.

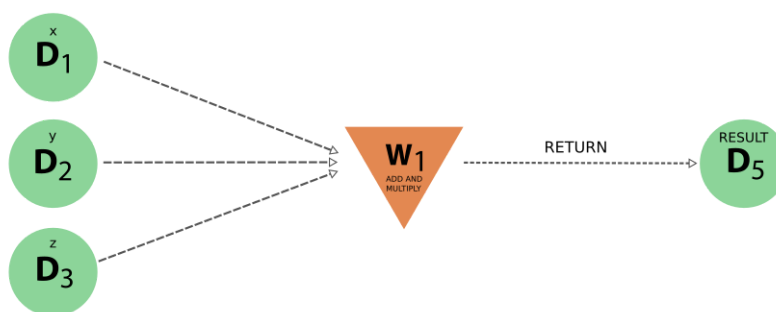


Figure 4. Logical provenance of the “add and multiply” work function in Figure 2, hiding all the intermediate steps; it is made clear that ‘x’, ‘y’ and ‘z’ are all inputs of the same workflow. From the AiiDA documentation.

3.3 AiiDA-Quantumpresso workflows

In the spirit of INTERSECT project, the realization of multi-physics, multi-code interoperable infrastructure (i.e. IM2D) requires the design and implementation of automatic workflows for

the calculation and the parsing of physical quantities (i.e. data-on-demand protocols) to be shared and distributed along the entire simulation chain. As a key example of *basic* materials parameters calculated by Quantum ESPRESSO and managed/curated by AiiDA, we considered the case of band gap on-demand.

The evaluation of the material band gap is based on the evaluation of ground state electronic structure, and in particular of the material band-structure. From the AiiDA point of view, this implies the design of a hierarchical series of *ad hoc* work-chains that, starting from the initial structure of the system, calculate the atomic and electronic ground state and the corresponding band-structure by using the Quantum ESPRESSO engine. Finally, the band gap value is obtained as a costless post-processing step.

3.3.1 Band gap workflow

The band gap workflow works around the capabilities of the PwCalculation calculation class, which handles the execution of the pw.x executable in Quantum ESPRESSO [4]. This process class has been greatly expanded to handle more than 20 different types of error reporting, in order to provide detailed information on the workchains built on top.

In particular, the band gap workflow is built on top of the following workchains:

1. **PwBaseWorkChain**, that handles the basic restart and error handling of the PwCalculation class, providing restart of the executable in case of failure or early recoverable problem, and adapting the output in case is needed (e.g., if the number of bands is insufficient). With this workchain it is possible to pass atomic pseudopotential based on families: rather than defining them individually for each atomic species, the workchain assigns them automatically from a set of pseudopotential inserted in AiiDA by the user. The k-points object can also be generated automatically by providing a spacing in reciprocal space, rather than explicitly defining the array of points.
2. **PwRelaxWorkChain**: this workchain handles the cell relaxation and provides additional error checking and restart capabilities, sanity checks (volume change between iterations) and final SCF calculation if requested.
3. **BandsWorkChain**: this workchain calculates the band structure of a crystalline material along high symmetry paths of its Brillouin zone. The calculation automatically performs a relaxation of the structure (if requested), calculates the symmetry of the crystal structure and generates a path for the calculation (using the SeeKPath K-points path analyzer available in AiiDA), runs a single point scf calculation of the relaxed structure, and finally calculates a band calculation along the standardized high symmetry paths, returning a BandStructure AiiDA object, along with the outputs of the scf and bands Quantum ESPRESSO calculations.

4. **BandStructureWorkChain**: this is an overlay of the BandsWorkChain class where the inputs are reduced to a minimum: the majority of the inputs that have to be defined in BandsWorkChain (parameters for the scf, bands and relax workflow, k-point spacing, pseudopotentials) are assigned according to a “protocol” name that contains and assigns reasonable parameters for the job, greatly reducing the complication, and making it easier to perform high-throughput calculations with no intervention from the user.

Finally, once the band-structure of material is obtained, band gap can be easily obtained and made available for further analysis/calculations: the BandGap workflow operates on top of the BandStructureWorkChain, and performs a simple analysis of the band structure AiiDA object, This further provides a dictionary with the definition of direct/indirect band gap type and a boolean value to indicate whether the structure is metallic or insulating.

3.4 AiiDA-SIESTA workflows

SIESTA’s [5] basic AiiDA workflows are built along the same lines as the Quantum ESPRESSO ones, with obvious differences regarding specification of input parameters (e.g., basis sets), and the internal architecture of the codes. Some elements (e.g. structure and k-point specification) use the same underlying AiiDA objects, so they are fully interoperable. Pseudopotentials in SIESTA are of the norm-conserving kind. Traditionally a SIESTA-specific file format (“psf”) has been used by the code. Recently we have developed a new interoperable format based on XML (PSML) that can be used to represent arbitrary norm-conserving pseudopotentials. Support for both the psf and psml pseudopotential types is available in AiiDA through special AiiDA-SIESTA data entry points. It is already planned that Quantum ESPRESSO could use the psml objects in the future to further increase the interoperability between the two codes. More information and details on the installation can be found in [6].

The SIESTA program is able to perform, in a single run, the computation of the electronic structure, the optional relaxation of the input structure, and a final analysis step in which a variety of magnitudes can be computed: band structures, projected densities of states, etc. The operations to be carried out are specified in a very flexible input format. Accordingly, the **SiestaBaseWorkchain** has been designed to be able to run the most general SIESTA calculation, with support for most of the available options (limited only by corresponding support in the parser plugin). In addition, the workflow is able to restart a calculation in case of failure (lack of electronic-structure or geometry relaxation convergence, termination due to walltime restrictions, etc).

The **SiestaBandsWorkchain** workflow can be used to compute and visualize the electronic band structure of the system of interest. Its inputs are a structure object and a specification

of the quality and cost level of the calculation. The latter is implemented internally, as in Quantum Espresso, as a set of protocols, which group operational parameters to offer the desired balance of accuracy and efficiency. Optionally, the workflow will relax the geometry of the system before computing the band structure. As discussed in the context of the Base workflow, the computation could be implemented as a single (restartable) SIESTA calculation, but it is instead segmented into different steps (optional relaxation followed by a final electronic-structure plus band calculation) to provide future support for different levels of accuracy in the two stages.

3.4.1 Band Gap workchain

Following the ideas developed for the Quantum ESPRESSO suite, the **SiestaBandGapWorkchain** is built around the **SiestaBandsWorkchain**: it uses the same input and output nodes by using the mechanism (new in AiiDA 1.0) of *port exposure*. Since both the Quantum ESPRESSO and SIESTA bands workflows produce as one their outputs the same type of band-structure AiiDA object, the SIESTA band-gap workflow uses basically the same underlying code to analyze the band eigenvalues, and it also outputs a dictionary object with the indirect band gap and a boolean value to indicate whether the structure is insulating or not.

The BandGap workflow is currently integrated into a feature branch of the AiiDA-SIESTA package which can be obtained from the development site [7].

4 Other on-demand properties

Other workflows relevant for the calculation of basic materials properties are in progress at various stages of development. The list of considered quantities includes the (projected) density of states, the effective mass, the dielectric constant and the work function. The implementation of the corresponding workflows will be completed in the next months. Future work will focus on the completion and thorough testing of these and other materials workflows relevant to the INTERSECT project (e.g. defect formation energy, diffusion barriers, etc), together with experimental and simulation validation in the context of device properties (deliverable D2.3).

References

[1] AiiDA documentation: <https://aiida-core.readthedocs.io/en/latest/>

[2] Registry for the public AiiDA plugins <https://github.com/aiida-team/aiida-registry>

[3] For more information about AiiDA workflows:

<https://aiida-core.readthedocs.io/en/latest/reference/index.html#workflows>

[4] P. Giannozzi *et al.*, *J.Phys.:Condens.Matter* **21**, 395502 (2009); P. Giannozzi *et al.*, *J.Phys.:Condens.Matter* **29**, 465901 (2017)

[5] SIESTA website <https://icmab.es/siesta/>

[6] AiiDA-SIESTA plugin documentation <https://aiida-siesta-plugin.readthedocs.io/>

[7] Code for the AiiDA-SIESTA plugin including BandGap workflow:

https://github.com/albgar/aiida_siesta_plugin/tree/bandgap. The workflow will be integrated in the master branch in the next release cycle, scheduled in the Spring of 2020.