Supplementary Material for *Efficient thermoelectricity in $Sr_2Nb_2O_7$ ⋯*

Giulio Casu, Andrea Bosin, and Vincenzo Fiorentini

Department of Physics, University of Cagliari, Italy

The Supplementary Material consists of the program package developed for, and used in, this paper, for free use on generic materials.

I. DISCLAIMER

This program is provided open-source, free, as-is, and with no warranty. The authors waive any responsibility for any damage that may result from its use in any context. Upon using this code to produce any publication or teaching material, please cite it as

G. Casu, `https://tinyurl.com/ycyvt4vk`;
G. Casu, A. Bosin, and V. Fiorentini, Phys. Rev. Materials **4,** 075404 (2020).

II. STRUCTURE

Upon unzipping a folder `pkg` should be found, which contains the three subdirectories `script`, `input`, `data`.

A. Contents of `script`

The `script` directory should contain:

- `CRT.py`: constant-relaxation-time thermoelectric coefficients
- `ART.py`: averaged-relaxation-time thermoelectric coefficients
- `NCRT.py`: full relaxation-time-model thermoelectric coefficients
- `TAU.py`: calculation of $\tau(E,T)$ for external use and plots.
- `plotter.py`: code doing the plotting
- `relaxtau.py`: library containing the relaxation time model $\tau(E,T)$

- `funzVar.py`: library containing various auxiliary functions
- `d2u.sh`: script converting CRLF-to-LF line ending

Execution times are insignificant for the auxiliary programs. For `*RT.py`, they depend on the input, the kind of material, and of course on the hardware available etc. For the present material a typical run may take up to roughly 15-20 minutes on a modern 4-core laptop.

B. Contents of `input`

The `input` directory should contain:

- `vasprun.xml`: the ab initio electronic structure calculation, which must be provided by you. The file in the original program package is for the current paper.
- `input`: input file with control and materials parameters. The one provided with the original program package is for the current paper.
- `latthcond`: file with lattice thermal conductivity (optional). The one provided with the original program package is a placeholder.

Details on the input format etc. are spelled out in the `input` file.

C. Contents of `data`

The `data` directory is used to dump output, which is then fetched by `plotter.py`. You will probably never need to look in here.

III. HOW TO INSTALL AND RUN

Given the minor disk occupation, the simplest way of using the code is unpacking and deploying it wherever it is needed. The requirements are

- `python 3.6` and its standard libraries (including `codecs`, `json`, `numpy`, `scipy`, and `matplotlib`) must be installed.

- The `BoltzTrap2` code must be installed for the present code to work. It can be obtained at `https://gitlab.com/sousaw/BoltzTraP2` or (on most systems) installed directly via `pip`. The current version as used in this work is `20.2.1`.

- The scripts are executed simply calling them as normal Unix-like commands. Should this fail, check the permissions. As well, some editors attach CR+LF line endings to files, and this may cause execution to fail: the simple script `d2u.sh` converts back to LF only.

To install and run:

1. unzip the whole package in your work dir (simple, negligible cost).
2. copy `vasprun.xml` to `input` dir;
3. prepare `input` file;
4. optionally, provide `latthcond` file (see `input` file for format);
5. in the `script` directory, run `CRT.py` and/or `ART.py` and/or `NCRT.py`, plus `TAU.py`, or alternatively run `do.sh`;
6. in the same dir, run `plotter.py` to analyze the results. You can easily adapt the plotting to your needs and taste.
7. To do convergence tests, repeat with different inputs. The usual parameters should be tested, so at least grid, multiplier, energy bins, energy range.