Computer Physics Communications 231 (2018) 140–145

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

BoltzTraP2, a program for interpolating band structures and calculating semi-classical transport coefficients☆

Georg K.H. Madsen [a,*], Jesús Carrete [a], Matthieu J. Verstraete [b,c]

[a] Institute of Materials Chemistry, TU Wien, A-1060 Vienna, Austria
[b] nanomat/QMAT/CESAM and Department of Physics, Université de Liège, allée du 6 août, 19, B-4000 Liège, Belgium
[c] European Theoretical Spectroscopy Facility, Belgium

ARTICLE INFO

Article history:
Received 19 December 2017
Received in revised form 30 April 2018
Accepted 4 May 2018
Available online 17 May 2018

Keywords:
Boltzmann transport equation
BoltzTraP

ABSTRACT

BoltzTraP2 is a software package for calculating a smoothed Fourier expression of periodic functions and the Onsager transport coefficients for extended systems using the linearized Boltzmann transport equation. It uses only the band and *k*-dependent quasi-particle energies, as well as the intra-band optical matrix elements and scattering rates, as input. The code can be used via a command-line interface and/or as a Python module. It is tested and illustrated on a simple parabolic band example as well as silicon. The positive Seebeck coefficient of lithium is reproduced in an example of going beyond the constant relaxation time approximation.

Program summary
Program Title: BoltzTraP2
Program Files doi: http://dx.doi.org/10.17632/bzb9byx8g8.1
Licensing provisions: GPLv3
Programming language: Python and C++
External routines/libraries: NumPy, SciPy, Matplotlib, spglib, ase, fftw, VTK, netCDF4, Eigen
Nature of problem: Calculating the transport coefficients using the linearized Boltzmann transport equation within the relaxation time approximation.
Solution method: Smoothed Fourier interpolation

1. Introduction

The first BoltzTraP program [1] provided a numerically stable and efficient method for obtaining analytic representations of quasi-particle energies. It has found broad adoption for the application of the Boltzmann transport equation (BTE) to such diverse fields as superconductors [2], transparent conductors [3] intermetallic phases [4] as well as thermoelectrics. Its application has been especially widespread for thermoelectrics research [5–12] for which it was originally conceived [13,14]. Furthermore, it has served as a reference for other methods for obtaining transport coefficients, such as maximally localized Wannier functions [15,16]. The numerical stability means that the approach can be automated [14] and the methodology has subsequently been used in several high-throughput studies [3,14,17–19].

The usefulness of the BoltzTraP approach can partly be attributed to the numerical efficiency of the procedure when the quasi-particle energies are approximated by the Kohn–Sham (KS) eigenvalues [20]. Once the multiplicative potential has been self-consistently calculated, calculating eigenvalues on a fine **k**-mesh is a comparatively simple computational step that can be trivially parallelized. An alternative approach calculates the derivatives necessary for the BTE directly from the intra-band momentum matrix elements [21]. However, within KS theory, it is often simpler to calculate a finer mesh of eigenvalues than to calculate the momentum matrix elements. When the quasi-particle energies cannot be calculated using a fixed multiplicative KS potential, as in beyond-KS methods such as hybrid functionals [22] or the GW approach [23], this argument no longer holds and calculating the momentum matrix elements [21] or using alternative interpolation methods [16,24,25] could be an advantage.

With the release of BoltzTraP2 we wish to achieve three objectives. First of all, a method to use both the eigenvalues and momentum matrix elements is introduced. This ensures that the interpolated manifolds exactly reproduce both the value and

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail address: georg.madsen@tuwien.ac.at (G.K.H. Madsen).

https://doi.org/10.1016/j.cpc.2018.05.010
0010-4655/

derivative at the calculated points. The advantages of the interpolation scheme of the original BoltzTraP approach [1], and the advantages of using the intra-band momentum matrix elements [21] are combined. Thereby the method becomes more suited for beyond-KS approaches. Secondly, we wish to make it more straightforward to avoid the constant relaxation time approximation (RTA) and handle e.g. a temperature-dependent transport distribution function due to electron–phonon coupling [26,27]. Finally, a further motivation for rewriting and rereleasing BoltzTraP is to provide a modular code based on the modern scripting language Python 3. While BoltzTraP is mainly thought of as a tool to evaluate the transport coefficients, the underlying algorithm can be generally useful for interpolating any periodic function. We hope that the new code can also serve as a library for further developments in this domain.

The paper is built as follows. First we present the interpolation scheme as well as the RTA-BTE. We discuss the interface to the code and provide an outlook and finally we use three examples to illustrate the methodology and results of the code.

2. Background

2.1. Band interpolation

The method is based on writing the quasi-particle energies and their derivatives, for each band, as Fourier sums

$$\tilde{\varepsilon}_{\mathbf{k}} = \sum_{\Lambda} c_{\Lambda} \sum_{R\in\Lambda} \exp(i\mathbf{k}\cdot\mathbf{R}) \tag{1}$$

$$\nabla\tilde{\varepsilon}_{\mathbf{k}} = i\sum_{\Lambda} c_{\Lambda} \sum_{R\in\Lambda} \mathbf{R}\exp(i\mathbf{k}\cdot\mathbf{R}) \tag{2}$$

where Λ are so-called stars representing a set of symmetry-equivalent lattice vectors. BoltzTraP was based on the idea by Shankland [28–30] that the coefficients should be obtained by minimizing a roughness function under the constraints that calculated quasi-particle energies should be exactly reproduced. This in turn means that the number of coefficients should be larger than the number of calculated points.

The derivatives can also be obtained from the intra-band optical matrix elements [21,31]

$$\nabla\varepsilon_{\mathbf{k}} = -\langle\psi_{\mathbf{k}}|\mathbf{p}|\psi_{\mathbf{k}}\rangle. \tag{3}$$

In BoltzTraP2 the Shankland algorithm [28–30] is extended so that the coefficients ensure that both the quasi-particle energies and their derivatives, Eq. (3), are exactly reproduced. This corresponds to minimizing the Lagrangian

$$I = \frac{1}{2}\sum_{\mathbf{R}} c_{\mathbf{R}}\rho_{\mathbf{R}} + \sum_{\mathbf{k}}\left[\lambda_{\mathbf{k}}\left(\varepsilon_{\mathbf{k}} - \tilde{\varepsilon}_{\mathbf{k}}\right) + \sum_{\alpha}\lambda_{\alpha,\mathbf{k}}\nabla_{\alpha}\left(\varepsilon_{\mathbf{k}} - \tilde{\varepsilon}_{\mathbf{k}}\right)\right] \tag{4}$$

with respect to the Fourier coefficient (c_R), and choosing the Lagrange multipliers ($\lambda_{\mathbf{k}}$ and $\lambda_{\alpha,\mathbf{k}}$) so that the constraints are fulfilled. The index α labels the three Cartesian directions and indicates that each calculated derivative, Eq. (3), adds three Lagrange multipliers. Like in the BoltzTraP code, we use the roughness function provided by Pickett et al. [32]

$$\rho_{\mathbf{R}} = \left(1 - c_1\frac{R}{R_{min}}\right)^2 + c_2\left(\frac{R}{R_{min}}\right)^6. \tag{5}$$

2.2. Boltzmann transport equation

BoltzTraP2 calculates transport coefficients based on the rigid-band approximation (RBA), which assumes that changing the temperature, or doping a system, does not change the band structure. In the RBA the carrier concentration, for a given T and μ, in a semiconductor can be obtained directly from the density of states (DOS)

$$n(\varepsilon) = \int\sum_{b}\delta(\varepsilon - \varepsilon_{b,\mathbf{k}})\frac{\mathrm{d}\mathbf{k}}{8\pi^3}, \tag{6}$$

where the subscript b runs over bands, by calculating the deviation from charge neutrality

$$c(\mu, T) = N - \int n(\varepsilon)f^{(0)}(\varepsilon;\mu,T)\mathrm{d}\varepsilon. \tag{7}$$

In Eq. (7), N is the nuclear charge and $f^{(0)}$ is the Fermi distribution function. In a semiconductor where charge neutrality would place the Fermi level in the band-gap, one can thus imagine how (at $T = 0$) moving μ into the conduction bands would produce a n-type material and moving μ into the valence bands would produce a p-type material.

The BTE describes the behavior of an out-of-equilibrium system in terms of a balance between scattering in and out of each possible state, with scalar scattering rates [33]. We have implemented the linearized version of the BTE under the RTA, where the transport distribution function

$$\sigma(\varepsilon, T) = \int\sum_{b}\mathbf{v}_{b,\mathbf{k}}\otimes\mathbf{v}_{b,\mathbf{k}}\tau_{b,\mathbf{k}}\delta(\varepsilon - \varepsilon_{b,\mathbf{k}})\frac{\mathrm{d}\mathbf{k}}{8\pi^3} \tag{8}$$

is used to calculate the moments of the generalized transport coefficients

$$\mathcal{L}^{(\alpha)}(\mu;T) = q^2\int\sigma(\varepsilon,T)(\varepsilon-\mu)^{\alpha}\left(-\frac{\partial f^{(0)}(\varepsilon;\mu,T)}{\partial\varepsilon}\right)\mathrm{d}\varepsilon \tag{9}$$

which give the charge and heat currents

$$j_e = \mathcal{L}^{(0)}\mathbf{E} + \frac{\mathcal{L}^{(1)}}{qT}(-\nabla T) \tag{10}$$

$$j_Q = \frac{\mathcal{L}^{(1)}}{q}\mathbf{E} + \frac{\mathcal{L}^{(2)}}{q^2T}(-\nabla T). \tag{11}$$

Identifying the two experimental situations of zero temperature gradient and zero electric current, we obtain the electrical conductivity, the Peltier coefficient, the Seebeck coefficient and the charge carrier contribution to the thermal conductivity as

$$\sigma = \mathcal{L}^{(0)} \tag{12}$$

$$\Pi = \frac{\mathcal{L}^{(1)}}{q\mathcal{L}^{(0)}} \tag{13}$$

$$S = \frac{1}{qT}\frac{\mathcal{L}^{(1)}}{\mathcal{L}^{(0)}} \tag{14}$$

$$\kappa_e = \frac{1}{q^2T}\left[\frac{(\mathcal{L}^{(1)})^2}{\mathcal{L}^{(0)}} - \mathcal{L}^{(2)}\right]. \tag{15}$$

The main advantage of the BoltzTraP procedure for evaluating the transport coefficients is that it is straightforward to obtain the group velocities from the $\mathbf{k}$-space derivatives of the quasi-particle energies, Eq. (2).

BoltzTraP is often associated with the constant relaxation time approximation (CRTA). The CRTA means that the Seebeck coefficient and Hall coefficient become independent of the scattering rate [34]. Therefore, they can be obtained on an absolute scale as a function of doping and temperature in a single scan. The CRTA in combination with the RBA, which makes the group velocities independent of μ and T, also has a computational advantage as it makes the transport distribution function, Eq. (8) independent of temperature and doping. The temperature and doping dependence of the transport coefficients $\mathcal{L}^{(\alpha)}$, Eq. (9), is solely due to the Fermi distribution function, and can be obtained via a scan over a fixed transport distribution function.

Clearly the CRTA will have limitations. It only delivers σ and κ_e dependent on τ as a parameter. Furthermore, the independence of S and R_H from τ is known to break down, even qualitatively, for specific cases [26]. While it is possible to run the original BoltzTraP with a temperature-, momentum- and band-dependent relaxation time, the structure of the code makes it inconvenient, and the functional form is quite limited. BoltzTraP2 makes it much more straightforward. The interpolated quasi-particle energies are usually considered to be independent from parameters such as temperature and Fermi level, and hence the interpolation does not need to be repeated (only the integration), for instance, to estimate thermoelectric coefficients for a different doping level or temperatures. The direct interface to the interpolation routines make it straightforward to interpolate the quasi-particle energies once and for all, and to avoid duplication of work when interpolating a temperature dependent τ.

3. Implementation and interface

3.1. General implementation aspects

BoltzTraP2 is implemented in Python 3, using syntax and standard library features that make it incompatible with Python 2. The Fortran code base of the original BoltzTraP was taken as a reference, but the new version was written from scratch. Numerical data is handled internally in the form of arrays, making extensive use of the NumPy and SciPy libraries [35,36]. Matplotlib [37] is used for plotting.

Efficiency is an important goal of this new implementation. Despite being implemented in a higher-level language, BoltzTraP2 achieves speeds comparable to the original BoltzTraP. There are several factors contributing to this. First, many of the most expensive operations are vectorized to be performed at a lower level by NumPy, avoiding expensive loops. Second, the symmetry-related code, heavy with loops over long lists of lattice points, is written in C++ and calls routines from the C API of the spglib library [38]. The C++ components are interfaced to Python by a Cython layer [39]. Third, fast Fourier transforms are delegated to optimized low-level libraries; specifically, the pyFFTW wrapper around FFTW [40] is used if available, with the default NumPy wrapper around FFTPACK [41] as a fallback. Finally, certain "embarrassingly parallel" loops can be run on several cores thanks to the multiprocessing module in the Python standard library.

BoltzTraP2 allows users to save results to files in JSON format, which is both human readable and parseable with a virtually limitless variety of tools and programming libraries. Specifically, there are two different sorts of JSON-formatted files at play. The first kind, bt2 files, contain the DFT input, all information about k points, the interpolation coefficients and a dictionary of metadata. The second category, btj files, contain the DFT input, the DOS, the thermoelectric coefficients for all values of temperature and chemical potential, and another metadata dictionary. Those dictionaries comprise several pieces of version information, a creation timestamp, and information about the scattering model used, if applicable. All JSON files are processed using the LZMA-based xz compressor, to drastically reduce the overhead of the text-based format.

The decision to stick with standard formats also affects other outputs of the calculations, stored in plain text with a column layout very similar to the one created by the original BoltTraP [1]. Most existing post-processing and analysis scripts can be adapted with minimal effort.

Regarding input formats, the current version of the code can read the native output of Wien2k [42,43], VASP [44] and ABINIT [45]. In the case of a VASP calculation, only the vasprun.xml file is required, while for Wien2k the necessary pieces of information are read from case.struct, case.energy and case.scf. If derivatives of the bands are to be used, the output-file case.mommat2 from the OPTIC [46] program is used. The code is modular enough that support for other DFT codes can be easily implemented. Alternatively, any DFT code can be adapted (or a translation script written) to create output in BoltzTraP2's own GENE format, designed for maximum simplicity. Examples of files in this format are provided with the source distribution, and it is only slightly modified compared to the original BoltTraP code GENE format.

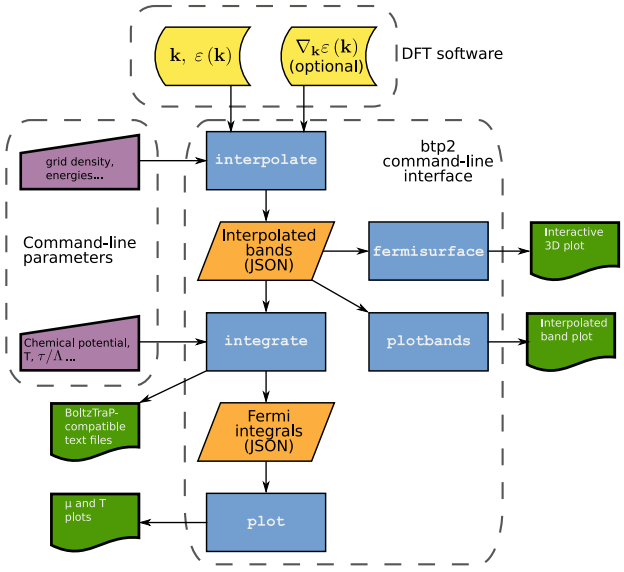

Fig. 1. Typical BoltzTraP2 workflow taking the user from the results of a DFT calculation to estimates of the thermoelectric coefficients for the system under study, and other related results, using the btp2 command-line interface.

BoltzTraP2 relies on Python's setuptools module for its build system. On many platforms, the program can be installed from source with a python setup.py install command, and no previous manual configuration. Moreover, we have uploaded it to the Python Package Index, so that even the downloading step can be bypassed in favor of a simple pip install BoltzTraP2. A copy of spglib is bundled with the BoltzTraP2, to avoid a separate installation process. Naturally, a C compiler and a C++-compliant C++ compiler are still needed when building from source.

3.2. Command-line interface

The most typical use case of BoltzTraP2 is the calculation of transport coefficients. This can be done directly through the btp2 command-line front-end, which implements the general workflow depicted in Fig. 1. It is designed to be self-documenting and controllable through command-line arguments, without the need for configuration files.

The process starts with a set of DFT results, typically from a non-self-consistent calculation using a dense k-point grid. The user first calls btp2 in "interpolate" mode to generate a representation of the bands interpolated to an even denser grid, which is stored in a JSON file. Optional command-line parameters to the "interpolate" step can be used to control the grid density, the number of bins used to compute the density of states, the minimum and maximum energies, etc. By saving the result of the interpolation to a file, we avoid repeating the calculation even if the user then wants to generate results for different situations.

To obtain a set of thermoelectric coefficients, the user needs to invoke btp2 a second time, now in "integrate" mode. In this mode

of operation, the command-line script accepts parameters such as the range of temperatures to scan, the dependence of scattering on electron energy, and so on. It then generates output in the form of text files, plus another compressed JSON file as described in the previous section.

The detailed list of command-line parameters, with their meanings, can be obtained by invoking `btp2` or one of its subcommands with the `-h` or `--help` flag.

In addition to the "integrate" and "interpolate" subcommands, Fig. 1 illustrates the use of the "fermisurface", "plotbands" and "plot" modes of the `btp2` command-line interface. Their job is to generate graphical representations of the `BoltzTraP2` output: an interactive 3D plot of the Fermi surface for different values of the energy, a plot of the interpolated electron energies along a specific path in reciprocal space, and plots of the thermoelectric coefficients as functions of temperature and chemical potential, respectively. 3D plotting will only be available if the optional `vtk` module is detected.

The code is documented, and a set of unit tests covering all the basic functionality is provided with the source distribution. The whole battery of tests can be run with `pytest`.

3.3. Using `BoltzTraP2` as a module

Advanced users may prefer to skip the command-line interface and access the full feature set of `BoltzTraP2` more directly. Those wanting to use the interpolation capabilities of `BoltzTraP2` in their own code, or using it as part of an automated workflow, will probably fall in this category. Furthermore, the `btp2` command-line interface only allows choosing between a uniform-relaxation-time model and a uniform-mean-free-path one. Users requiring custom parameterizations of the electronic scattering rates will need to bypass the interface. This is easily accomplished by calling the API of the `BoltzTraP2` Python module, either from a script or from an interactive Python shell, such as the `Jupyter` notebook [47]. Crystal structures are represented as `ase` atoms objects [48], which allows for easy interfacing with many other Python libraries and external software.

The best reference about the API of the `BoltzTraP2` is the source code of the `btp2` interface itself, and a set of documented examples that are provided with the source distribution of `BoltzTraP2`. The examples illustrate how to accomplish specific tasks and reproduce several results obtained with the original `BoltzTraP` code as well as the three examples in the following section.

4. Examples

The API of `BoltzTraP2` is illustrated through three examples. These represent "non-standard" uses of the code, which cannot be accessed through the command line interface. The python source codes reproducing the plots shown below and several others can be found in the `examples` directory.

4.1. Isotropic parabolic band model

The simplest way to illustrate the library functionality of `BoltzTraP2` is the parabolic band model. Consider a dispersion relation expressed as,

$$\varepsilon(k) = \frac{\hbar^2 k^2}{2m^*} \tag{16}$$

where m^* is the effective mass of the electrons in the band. For an isotropic parabolic band model we can replace the outer product of group velocities, Eq. (8), by k/m^* and the volume element $\mathrm{d}\mathbf{k}$ in the three dimensional volume integral in Eq. (8) by $4\pi k^2 \mathrm{d}k$, thereby obtaining analytic expressions for $n(\varepsilon)$ and $\sigma(\varepsilon)$

$$n(\varepsilon) = \frac{1}{4\pi^2}\left(\frac{2m^*}{\hbar^2}\right)^{3/2} \varepsilon^{1/2} \tag{17}$$

$$\sigma(\varepsilon) = \frac{1}{3\pi^2}\frac{\sqrt{2m^*}}{\hbar^3}\tau\varepsilon^{3/2}. \tag{18}$$

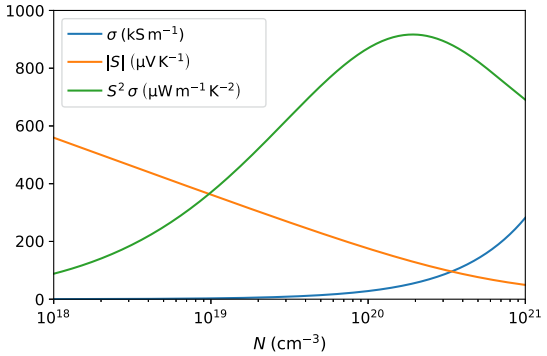

Fig. 2. S, σ, and the thermoelectric PF as a function of carrier concentration. The transport coefficients have been evaluated using a parabolic band model with $m^* = m_e$. The temperature and relaxation time were set to $T = 500$ K and $\tau = 10^{-14}$ s, respectively.

Evaluating the carrier concentration, Eq. (7), and the transport coefficients, Eqs. (9)–(14), leads directly to the famous plot Fig. 2. For comparison we have created a single parabolic band numerically on a $25 \times 25 \times 25$ **k**-mesh for a cubic unit cell with $a = 5$ Å. The band was interpolated onto a mesh containing 5 times the points. The resulting transport coefficients are indistinguishable from the analytic in Fig. 2.

Fig. 2 can be reproduced with the `parabolic.py` script in the `examples` directory of the `BoltzTraP2` distribution.

4.2. Inclusion of momentum matrix elements. Silicon

The band structure of silicon makes one of the simplest examples that is not trivial for an interpolation scheme. The conduction band minimum (CBM) of Si is made up of pockets found along the six-fold degenerate $\Gamma - X$ line. Furthermore, it has a non-symmorphic space group so that the bands can cross at the zone boundary. A crossing at the zone boundary will mean that the bands will not necessarily touch the zone-boundary "horizontally". A purely Fourier-based interpolation scheme, as the one used in `BoltzTraP2` can give false derivatives at these points, meaning that a very fine k-mesh can be necessary to obtain converged results.

The CBM pocket found along the $\Gamma - X$ line, which will dominate the transport properties of n-doped Si, is illustrated in Fig. 3. Fig. 3 compares the result of a usual DFT calculation of a band structure, with a fine set of k-points along a specific direction, with that obtained by the analytic interpolation of a coarse $9 \times 9 \times 9$ k-point mesh, Eq. (1). A $9 \times 9 \times 9$-mesh corresponds to only 35 k-points in the irreducible part of the Brillouin zone (IBZ). A $9 \times 9 \times 9$-mesh corresponds to a typical k-mesh used for a self-consistent DFT calculation and is obviously not a fine k-mesh, that would typically be calculated non-self-consistently for transport calculations [14] Furthermore, as the lowest conduction bands are degenerate at the X-point, Fig. 3, the non-symmorphic space group does result in the derivatives of interpolated bands being incorrect at this point. However, Fig. 3 illustrates how the modified Lagrangian, Eq. (4), forces the fit to reproduce the exact derivatives at the calculated points. Thereby, both the position and derivatives of the pocket

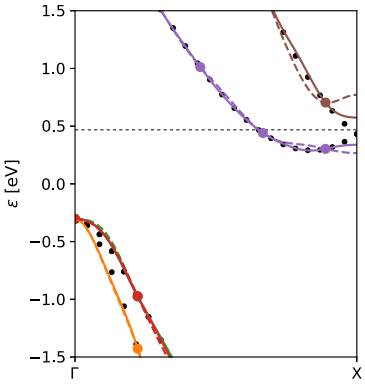

Fig. 3. Silicon band edges along the $\Gamma - X$ line. The black points are calculated points along this specific direction. The colored lines correspond to the interpolated bands based on a coarse $9 \times 9 \times 9$ k-point mesh. The points belonging to this mesh are marked with larger colored points. The full lines are obtained by including the momentum matrix elements in the fit and the dashed use only the eigenvalues. Thin dashed line: chemical potential used below in Fig. 4.

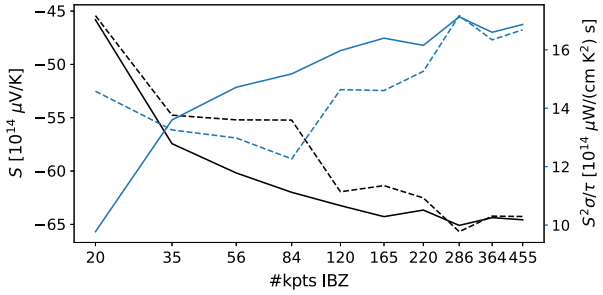

Fig. 4. Convergence of the Seebeck coefficient and thermoelectric power factor as a function of number k-points in the irreducible Brillouin zone. The full lines are obtained by including the momentum matrix elements in the fit whereas the dashed correspond to including only the eigenvalues.

are well reproduced. On the other hand, if only the eigenvalues are included in the fit, this mesh is obviously too coarse and the algorithm fails to reproduce either the position or the derivatives at the pocket (purple dashed line in Fig. 3).

The impression obtained graphically in Fig. 3 is quantified in Fig. 4. The Seebeck coefficient and the thermoelectric power factor, $S^2\sigma/\tau$, are calculated at a chemical potential close to the CBM (marked by the thin dashed line in Fig. 3) using the CRTA, Eqs. (8)–(14). It is seen how the results obtained by the modified Lagrangian show both a faster and more systematic trend, reaching convergence at about half the number of k-points needed when the derivatives are not included in the fit.

Figs. 3 and 4 can be reproduced with the `Si_pocket.py` and `Si_conv.py` scripts in the `examples` directory of the `BoltzTraP2` distribution. Before running these two scripts, the script `Si_btp.py` which does the Fourier interpolation and stores the results in JSON formatted `bt2` files.

4.3. State dependent relaxation time in Lithium

In `BoltzTraP2` the interpolation and integration steps are more explicitly decoupled than in `BoltzTraP`. This allows the interpolation capabilities of the code to be used to match quantities represented on different grids. We illustrate this possibility by considering the transport distribution function of bcc-Lithium.

First the KS eigenvalues were obtained on a $36 \times 36 \times 36$ **k**-point mesh and interpolated onto a grid containing 60 times the number of **k**-points. Fig. 5 illustrates how this leads to a positive

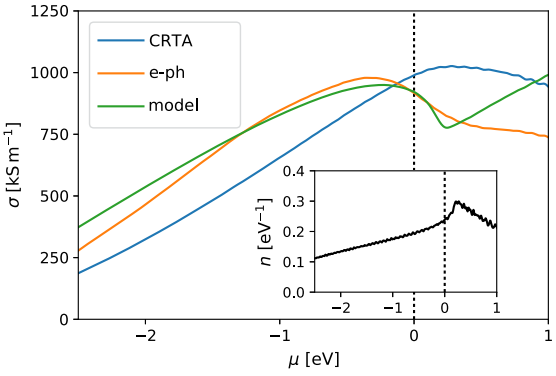

Fig. 5. Calculated conductivity of bcc-Li using a constant relaxation time, band and momentum dependent relaxation times due to electron–phonon coupling, and an energy dependent model $\tau^{-1}(\varepsilon) = cn(\varepsilon)$, relaxation time.

slope of the CRTA transport distribution function at the Fermi level. Consequently, Eqs. (9) and (14), we find a negative Seebeck coefficient of $S = -1.9$ μV/K at 300 K as in Ref. [26]. The obtained CRTA transport distribution function, Fig. 5, is in good agreement with a more "usual" `BoltzTraP` type calculation where the KS eigenvalues were calculated on $58 \times 58 \times 58$-grid and interpolated onto a grid containing four times as many **k**-points. This lends credibility to the interpolation of the scattering rates that we will now perform.

As a by-product of the calculations in Xu et al. [26], we can obtain the computationally costly relaxation times due to electron–phonon scattering, $\tau^{ep}_{n\mathbf{k}}$, on a relatively coarse $24 \times 24 \times 24$ **k**-point mesh. Using our interpolation scheme, the calculated band- and momentum-dependent relaxation times were interpolated onto the same fine grid as used for the KS eigenvalues. The inclusion of $\tilde{\tau}^{ep}_{n\mathbf{k}}$ leads to a change of slope at the Fermi level and consequently a positive Seebeck coefficient ($S = +2.9$ μV/K at 300 K).

As pointed out in the original work [26], the change of sign can be understood by a simple model where the scattering rate is proportional to the DOW. This is illustrated in the inset in Fig. 5 where a peak in the DOS is found above the Fermi level. Consequently, we fit a $\tau^{-1}(\varepsilon) = cn(\varepsilon)$ model to the calculated $(\tau^{ep}_{n\mathbf{k}})^{-1}$, thereby obtaining an energy dependent relaxation time. As seen in Fig. 5 a very good agreement, especially around the Fermi level, is obtained with the transport distribution calculated using the full $\tau^{ep}_{n\mathbf{k}}$.

Fig. 5 can be reproduced with the `Li_bcc.py` script in the `examples` directory of the `BoltzTraP2` distribution.

5. Conclusion

We have presented a new a software package, `BoltzTraP2`, based mainly on Python 3. The methodology is based on a smoothed Fourier expression for periodic functions and uses only the band and k-dependent quasi-particle energies as well as the intra-band optical matrix elements and scattering rates as input. The Onsager transport coefficients have been evaluated for a simple periodic band, as well as Silicon and Lithium using the linearized Boltzmann transport equation. The code can be used via a command-line interface and as a Python module.

Acknowledgments

MJV acknowledges support from the Communauté française de Belgique through an ARC grant (AIMED 15/19-09) and the Belgian Fonds National de la Recherche Scientifique FNRS, under Grant Number PDR T.1077.15-1/7. GKHM and JC acknowledge support

from M-era.net through the ICETS project (DFG: MA 5487/4-1) and the EU Horizon 2020 Grant No. 645776 (ALMA). Computational resources have been provided by the Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CECI), funded by FRS-FNRS G.A. 2.5020.11; the Tier-1 supercomputer of the Fédération Wallonie-Bruxelles, funded by the Walloon Region under G.A. 1117545; the PRACE-3IP DECI grants, on ARCHER and Salomon (ThermoSpin, ACEID, OPTOGEN, and INTERPHON 3IP G.A. RI-312763) and the Vienna Scientific Cluster (project number 70958: ALMA).

References

[1] G.K.H. Madsen, D.J. Singh, Comput. Phys. Comm. 175 (2006) 67–71.
[2] D.J. Singh, M.-H. Du, Phys. Rev. Lett. 100 (2008) 237003. http://dx.doi.org/10.1103/PhysRevLett.100.237003.
[3] G. Hautier, A. Miglio, G. Ceder, G.-M. Rignanese, X. Gonze, Nature Comm. 4 (2013) 2292.
[4] J. Dolinšek, M. Komelj, P. Jeglič, S. Vrtnik, D. Stanić, P. Popčević, J. Ivkov, A. Smontara, Z. Jagličić, P. Gille, Y. Grin, Phys. Rev. B 79 (2009) 184201. http://dx.doi.org/10.1103/PhysRevB.79.184201.
[5] A.F. May, D.J. Singh, G.J. Snyder, Phys. Rev. B 79 (2009) 153101. http://dx.doi.org/10.1103/PhysRevB.79.153101.
[6] S. Ouardi, G.H. Fecher, B. Balke, X. Kozina, G. Stryganyuk, C. Felser, S. Lowitzer, D. Ködderitzsch, H. Ebert, E. Ikenaga, Phys. Rev. B 82 (2010) 085108.
[7] D. Parker, X. Chen, D.J. Singh, Phys. Rev. Lett. 110 (2013) 146601. http://dx.doi.org/10.1103/PhysRevLett.110.146601.
[8] D.B. Luo, Y.X. Wang, Y.L. Yan, G. Yang, J.M. Yang, J. Mater. Chem. A 2 (36) (2014) 15159–15167. http://dx.doi.org/10.1039/c4ta02452d.
[9] J.Y. Kim, J.C. Grossman, Nano Lett. 15 (2015) 2830–2835. http://dx.doi.org/10.1021/nl504257q.
[10] A. Hong, L. Li, R. He, J. Gong, Z. Yan, K. Wang, J.-M. Liu, Z. Ren, Sci. Rep. 6 (2016) 22778.
[11] J. He, M. Amsler, Y. Xia, S.S. Naghavi, V.I. Hegde, S. Hao, S. Goedecker, V. Ozolins, C. Wolverton, Phys. Rev. Lett. 117 (2016) 046602. http://dx.doi.org/10.1103/PhysRevLett.117.046602.
[12] J. Zhang, L. Song, G.K.H. Madsen, K.F.F. Fischer, W. Zhang, X. Shi, B.B. Iversen, Nature Comm. 7 (2016) 10892.
[13] G.K.H. Madsen, K. Schwarz, P. Blaha, D.J. Singh, Phys. Rev. B 68 (2003) 125212.
[14] G.K.H. Madsen, J. Am. Chem. Soc. 128 (2006) 12140.
[15] J.R. Yates, X. Wang, D. Vanderbilt, I. Souza, Phys. Rev. B 75 (2007) 195121. http://dx.doi.org/10.1103/PhysRevB.75.195121.
[16] G. Pizzi, D. Volja, B. Kozinsky, M. Fornari, N. Marzari, Comput. Phys. Comm. 185 (2014) 422–429. http://dx.doi.org/10.1016/j.cpc.2013.09.015.
[17] J. Carrete, N. Mingo, S. Wang, S. Curtarolo, Adv. Funct. Mater. 24 (47) (2014) 7427–7432. http://dx.doi.org/10.1002/adfm.201401201.
[18] S. Bhattacharya, G.K.H. Madsen, Phys. Rev. B 92 (2015) 085205. http://dx.doi.org/10.1103/PhysRevB.92.085205.
[19] H. Zhu, G. Hautier, U. Aydemir, Z.M. Gibbs, G. Li, S. Bajaj, J.-H. Pöhls, D. Broberg, W. Chen, A. Jain, et al., J. Mater. Chem. C 3 (2015) 10554–10565.
[20] W. Kohn, L.J. Sham, Phys. Rev. 140 (1965) A1133–A1138.
[21] T.J. Scheidemantel, C. Ambrosch-Draxl, T. Thonhauser, J.V. Badding, J.O. Sofo, Phys. Rev. B 68 (2003) 125210. http://dx.doi.org/10.1103/PhysRevB.68.125210.
[22] A.D. Becke, J. Chem. Phys. 98 (7) (1993) 5648–5652. http://dx.doi.org/10.1063/1.464913.
[23] L. Hedin, Phys. Rev. 139 (1965) A796–A823. http://dx.doi.org/10.1103/PhysRev.139.A796.
[24] D. Prendergast, S.G. Louie, Phys. Rev. B 80 (2009) 235126.
[25] K. Berland, C. Persson, Comput. Mater. Sci. 134 (2017) 17–24.
[26] B. Xu, M.J. Verstraete, Phys. Rev. Lett. 112 (2014) 196603. http://dx.doi.org/10.1103/PhysRevLett.112.196603.
[27] W. Li, Phys. Rev. B 92 (2015) 075405. http://dx.doi.org/10.1103/PhysRevB.92.075405.
[28] R.N. Euwema, D.J. Stukel, T.C. Collins, J.S. DeWitt, D.G. Shankland, Phys. Rev. 178 (1969) 1419–1423. http://dx.doi.org/10.1103/PhysRev.178.1419.
[29] D.G. Shankland, Int. J. Quantum Chem. 5 (1971) 497–500. http://dx.doi.org/10.1002/qua.560050857.
[30] D.D. Koelling, J.H. Wood, J. Comput. Phys. 67 (1986) 253–262.
[31] E. Assmann, P. Wissgott, J. Kuneš, A. Toschi, P. Blaha, K. Held, Comput. Phys. Comm. 202 (2016) 1–11.
[32] W.E. Pickett, H. Krakauer, P.B. Allen, Phys. Rev. B 38 (1988) 2721–2726.
[33] J.M. Ziman, Electrons and Phonons: The Theory of Transport Phenomena in Solids, Oxford University Press, 2000.
[34] D.J. Singh, I.I. Mazin, Phys. Rev. B 56 (1997) R1650–R1653. http://dx.doi.org/10.1103/PhysRevB.56.R1650.
[35] S. van der Walt, S.C. Colbert, G. Varoquaux, Comput. Sci. Eng. 13 (2011) 22–30. http://dx.doi.org/10.1109/MCSE.2011.37.
[36] T.E. Oliphant, Comput. Sci. Eng. 9 (2007) 10–20. http://dx.doi.org/10.1109/MCSE.2007.58.
[37] J.D. Hunter, Sci. Eng. 9 (2007) 90–95. http://dx.doi.org/10.1109/MCSE.2007.55.
[38] A. Togo, spglib, a c library for finding and handling crystal symmetries, http://spglib.sourceforge.net/. (Accessed 26 July 2016).
[39] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. Seljebotn, K. Smith, Comput. Sci. Eng. 13 (2011) 31–39. http://dx.doi.org/10.1109/MCSE.2010.118.
[40] M. Frigo, SIGPLAN Not. 39 (2004) 642–655. http://dx.doi.org/10.1145/989393.989457.
[41] P.N. Swarztrauber, in: G. Rodrigue (Ed.), Parallel Computations, Academic Press, New York, 1982, pp. 51–83.
[42] K. Schwarz, P. Blaha, G.K.H. Madsen, Comput. Phys. Comm. 147 (2002) 71–76.
[43] P. Blaha, K. Schwarz, G.K.H. Madsen, D. Kvasnicka, J. Luitz, . WIEN2k, WIEN2k, an Augmented Plane Wave Plus Local Orbitals Program for Calculating Crystal Properties, Vienna University of Technology, Austria, ISBN: 3-9501031-1-2, 2001.
[44] G. Kresse, J. Furthmüller, Phys. Rev. B 54 (16) (1996) 11169–11186. http://dx.doi.org/10.1103/PhysRevB.54.11169.
[45] X. Gonze, F. Jollet, F. Abreu Araujo, D. Adams, B. Amadon, T. Applencourt, C. Audouze, J.-M. Beuken, J. Bieder, A. Bokhanchuk, E. Bousquet, F. Bruneval, D. Caliste, M. Côté, F. Dahm, F. Da Pieve, M. Delaveau, M. Di Gennaro, B. Dorado, C. Espejo, G. Geneste, L. Genovese, A. Gerossier, M. Giantomassi, Y. Gillet, D. Hamann, L. He, G. Jomard, J. Laflamme Janssen, S. Le Roux, A. Levitt, A. Lherbier, F. Liu, I. Lukačević, A. Martin, C. Martins, M. Oliveira, S. Poncé, Y. Pouillon, T. Rangel, G.-M. Rignanese, A. Romero, B. Rousseau, O. Rubel, A. Shukri, M. Stankovski, M. Torrent, M. Van Setten, B. Van Troeye, M. Verstraete, D. Waroquiers, J. Wiktor, B. Xu, A. Zhou, J. Zwanziger, Comput. Phys. Comm. 205 (2016) 106–131. http://dx.doi.org/10.1016/j.cpc.2016.04.003. URL https://doi.org/10.1016/j.cpc.2016.04.003.
[46] C. Ambrosch-Draxl, J.O. Sofo, Comput. Phys. Comm. 175 (2006) 1–14.
[47] F. Pérez, B.E. Granger, Comput. Sci. Eng. 9 (2007) 21–29. http://dx.doi.org/10.1109/MCSE.2007.53.
[48] A.H. Larsen, J.J. Mortensen, J. Blomqvist, I.E. Castelli, R. Christensen, M. Dułak, J. Friis, M.N. Groves, B. Hammer, C. Hargus, E.D. Hermes, P.C. Jennings, P.B. Jensen, J. Kermode, J.R. Kitchin, E.L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J.B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K.S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, K.W. Jacobsen, J. Phys.: Condens. Matter 29 (2017) 273002.