
BoltzTraP2Y

Release 0.0

Yi Wang, Zi-Kui Liu, and Long-Qing Chen

Feb 18, 2021

CONTENTS:

1	README	1
1.1	Quick run instruction	1
1.2	Run using Jupyter Notebook	2
1.3	Change logs in coding	2
2	Theoretical Brief	3
3	Installation	5
4	Basics to run BoltzTraP2	7
4.1	Example by Jupyter Notebook	7
5	BoltzTraP2	9
5.1	Prerequisites	9
5.2	Compiling and install BoltzTraP2	9
5.3	Running the tests	10
6	Utilities	11
6.1	Mixing two phases	11
7	Penn State's work	13
7.1	Thermodynamic respect of certain thermoelectric quantities	13
7.2	Revised output for the dope module	14
7.3	Thermodynamic formulations	14
7.4	Electrical conductivity	16
7.5	Separation of electron and hole contributions	16
8	Changelog	17
8.1	0.0 (2021-01-27)	17
9	Acknowledgements	19
10	Indices and tables	21

README

The long term goal of this package is to bridge the data between DFT results, either in its original format or those save in MongoDB database, and the other simulations, for example, the phase-field simulation of phase transition due to electronic structure change, typically, the metal insulator transition (MIT).

The above figure shows a schematic demonstration of the technical procedure for the data flow of the package. This package was mainly implemented through the combination of several distinct approaches for accounting the roles of electrons and lattice vibrations:

1. the kinetic properties by means of solving the Boltzmann transport equation. The code was mainly based on forking the [BoltzTraP2](#) by @georg.madsen et al. The major changes include the separation of the electron and hole contributions, in addition to technical improvements on the calculations of the chemical potential of electrons which makes it more efficient for calculation at low temperature that requires extremely refined meshes on electron energy.
2. the thermodynamic properties of electrons involving the mixing between two phases per our recent observations that certain eletron transfer related quantities, including:
 - the Seebeck coefficient - [First-principles thermodynamic theory of Seebeck coefficients](#); and
 - the Lorenz number - [Lorenz Number and Electronic Thermoelectric Figure of Merit: Thermodynamics and Direct DFT Calculations](#)

which were thought to be dependent on the scattering mechanism and therefore can only be calculated through the kinetic manner such as the Boltzmann transport equation, are in fact thermodynamic parameters and therefore can be predicted solely based on the electron density of states.

1. Our development for a package for the automation of first-principles calculations of thermodynamic properties of materials at finite temperature employing the quasi-hamonic phonon approach, as given in a separate documentaton (www.dfttk.org)

For more details, see the section **Penn State's work**.

1.1 Quick run instruction

1. perform DFT calculations for, including a. 0-K electron energetics. These data contain the lattice structure, 0-K total energy, electronic density of states, electron energies in each k -mesh. b. interatomic force constants through supercell or linear response approach. These data will be used to calculate the lattice contribution to the thermodynamic properties.
2. invoke the BoltzTrap2Y package to calculate the kinetic/thermodynamic properties based on 0-K DFT results
3. when needed, invoke DFTTK to postprocess the interatomic force constants to get lattice contribution to the thermodynamic properties.
4. convert the data calculated by BoltzTrap2Y/DFTTK into the input to phase field simulation

1.2 Run using Jupyter Notebook

To make ease of the package, we have prepared an extensive jupyter notebook script (which assumes that the step of DFT calculations have been completed) which can be executed by the following steps

1. clicking the link [BoltzTrap2Y.ipynb](#) followed by click the icon in the right hand side of the web page. By default, a file named like `BoltzTrap2Y.ipynb` will be saved in your `Downloads` folder in the case of Windows computer. After that, go back to this page and run the codes using the free google notebook server by
2. clicking the link [jupyter notebook google](#) followed by uploading the downloaded code through clicking `file->upload` in the jupyter notebook google page.
3. clicking `Runtime->Run all` in the jupyter notebook google page

For more details, see the section **Example by Jupyter Notebook**.

1.3 Change logs in coding

As a numerical demonstration, we have implimented the computational procedure by python code through forking the [BoltzTraP2](#) by @georg.madsen et al. The major extensions, which at present are only made for constant doping calculations, are briefed below:

1. revised `interface.py`
2. extended `bandlib.py` into `bandlibEXT.py`.
3. extended `io.py` into `ioEXT.py`
4. added `remesh.py` for calculations in low temperature (meshed increased to ~100,000 so that the calculated results can be good until a few of 10 K).
5. improved the computational efficiency on the calculations of the chemical potential of electrons (Fermi levels)
6. added `plotpng.py` for figure plot.

THEORETICAL BRIEF

In this section, we summarize the formulations for the Seebeck coefficient and Lorenz Number between the Boltzmann transport equation (BTE) and the proposed thermodynamic equation (TE). For more details, see the section of *Penn State's work*

- Seebeck coefficient

$$S_{BTE} = -\frac{1}{eT} \int \frac{\int [\varepsilon - \mu] f(1-f) \mathbf{X}(\varepsilon) d\varepsilon}{\int f(1-f) \mathbf{X}(\varepsilon) d\varepsilon}$$

vs

$$S_{TE} = -\frac{1}{eT} \int \frac{\int [\varepsilon - \mu] f(1-f) D(\varepsilon) d\varepsilon}{\int f(1-f) D(\varepsilon) d\varepsilon}$$

- Lorenz number

$$L_{BTE} = \frac{1}{e^2 T^2} \frac{\left\{ \int f(1-f) [\varepsilon - \mu]^2 \mathbf{X}(\varepsilon) d\varepsilon - \frac{[\int f(1-f) [\varepsilon - \mu] \mathbf{X}(\varepsilon) d\varepsilon]^2}{\int f(1-f) \mathbf{X}(\varepsilon) d\varepsilon} \right\}}{\int f(1-f) \mathbf{X}(\varepsilon) d\varepsilon}$$

vs

$$L_{TE} = \frac{1}{e^2 T^2} \frac{\left\{ \int f(1-f) [\varepsilon - \mu]^2 D(\varepsilon) d\varepsilon - \frac{[\int f(1-f) [\varepsilon - \mu] D(\varepsilon) d\varepsilon]^2}{\int f(1-f) D(\varepsilon) d\varepsilon} \right\}}{\int f(1-f) D(\varepsilon) d\varepsilon}; \text{ or}$$

$$L_{TE} = \frac{k_B}{e^2} \frac{C_{el}}{n_{eff}}$$

- Relation between $\mathbf{X}(\varepsilon)$ and $D(\varepsilon)$

$\mathbf{X}(\varepsilon)$ in the equation for S_{BTE} or L_{BTE} for the Boltzmann transport theory is called the transport distribution function [see Madsen, CPC 231, 140 (2018); Scheidemantel, PRB 68, 125210(2003)]

$$X^{\alpha\beta}(\varepsilon) = \frac{e^2}{V} \int \sum_i v_i^\alpha(\mathbf{k}) v_i^\beta(\mathbf{k}) \tau_{i,\mathbf{k}} \delta(\varepsilon - \varepsilon_i(\mathbf{k})) \frac{d\mathbf{k}}{8\pi^3}$$

where the electron group velocity is

$$v_i^\alpha(\mathbf{k}) = \frac{\partial \varepsilon_i(\mathbf{k})}{\partial k^\alpha}$$

and $\tau_{i,\mathbf{k}}$ is called the relaxation time.

$$D(\varepsilon) = \frac{1}{V} \int \sum_i \delta(\varepsilon - \varepsilon_i(\mathbf{k})) \frac{d\mathbf{k}}{8\pi^3}$$

$D(\varepsilon)$ in the equation for S_{TE} or L_{TE} for the **thermodynamic theory** represents the electron density of states

- It is observed that when the transport distribution function, $\mathbf{X}(\varepsilon)$, is replaced by the the electron density of states, $D(\varepsilon)$, the expressions for the Seebeck coefficient and Lorenz number due to Boltzmann transport equation completely reduced to those due to thermodynamic theory.

INSTALLATION

It is recommended to install the package under the [anaconda](#) environment. Under the anaconda prompt, one can create a preferred directory and then run

```
git clone https://gitlab.com/yiwang62/BoltzTraP2.git
cd BoltzTraP2/
setenv CC g++
```

Edit the `setup.py` file, depending on your CC environment, use

```
required_compile_flags = ["-std=c++11"], or
required_compile_flags = ["-std=c++0x"]
```

Last, run

```
pip install -e . #for develop version (recomended)
#if failed, try
python setup.py install #or
python setup.py develop #for develop version (recomended)
```


BASICS TO RUN BOLTZTRAP2

Once intalled, one can try `bt2 -h` for available command line options. This fork of BoltzTraP2 is mostly focused at the constant doping case, try `bt2 dope -h` for more details.

For the kinetic calculations, all the commands follow those from BoltzTraP2. For more details, check out the [BoltzTraP2 tutorial](<https://gitlab.com/sousaw/BoltzTraP2/-/wikis/tutorial>).

4.1 Example by Jupyter Notebook

Jupyter Notebook (previously IPython Notebook) files are used for interactive computing in many fields and contain a complete record of the user's sessions and include code, narrative text, equations, and rich output.

The advantage to use Jupyter Notebook is that it makes very easy for a reseracher lacking of experience software installation. For example, it takes quite significant time to figure this or that kind installation problem, even for the installation of the upstream package of this work, BoltzTraP2, in cmake and g++ settings

We have prepared an extensive jupyter notebook script which can be downloaded by

1. clicking the link [BoltzTrap2Y.ipynb](#) followed by click the  icon in the right hand side of the web page. By default, a file named like `BoltzTrap2Y.ipynb` will be saved in your Downloads folder in the case of Windows computer. After that, go back to this page and run the codes using the free google notebook server by
2. clicking the link [jupyter notebook google](#) followed by uploading the downloaded code through clicking `file->upload` in the jupyter notebook google page.
3. last, you can do one-click to run the code by clicking `Runtime->Run all` in the jupyter notebook google page

BOLTZTRAP2



BoltzTraP2 is a modern implementation of the [smoothed Fourier interpolation algorithm](#) for electronic bands that formed the base of the original and widely used [BoltzTraP](#) code. One of the most typical uses of BoltzTraP is the calculation of thermoelectric transport coefficients as functions of temperature and chemical potential in the rigid-band picture. However, many other features are available, including 3D plots of Fermi surfaces based on the reconstructed bands. For more information, check out the [BoltzTraP2 tutorial](#).

5.1 Prerequisites

BoltzTraP2 is a Python module, with a small performance-critical portion written in C++ and [Cython](#). BoltzTraP2's runtime requirements are Python version 3.5 or higher, and the Python libraries [NumPy](#), [SciPy](#), [matplotlib](#), [spglib](#), [NetCDF4](#) and [ASE](#). All of them can be easily obtained from the [Python Package Index \(PyPI\)](#), using tools such as [pip](#). They may also be bundled with Python distributions aimed at scientists, like [Anaconda](#), and with a number of Linux distributions. If [pip](#) is used to install BoltzTraP2, dependencies should be resolved automatically.

If available, BoltzTraP2 will also make use of [pyFFTW](#) (for faster Fourier transforms), [colorama](#) (to colorize some console output) and [VTK](#) (to generate 3D representations). Those packages are not required, but they are recommended to be able to access the full functionality of BoltzTraP2.

Furthermore, compiling BoltzTraP2 from its sources requires a C++ compiler, and the development headers and libraries for Python. [Cython](#) is **not** required for a regular compilation.

5.2 Compiling and install BoltzTraP2

The easiest way to get BoltzTraP2 is to run:

```
$ pip install BoltzTraP2
```

This should take care of downloading and installing the dependencies as well.

Users installing from source must install the dependencies first and then run:

```
$ python setup.py install
```

from the source directory. For finer-grained control, please see the output of these commands:

```
$ python setup.py --help
$ python setup.py --help-commands
$ python setup.py install --help
```

The BoltzTraP2 installer supports

```
$ python setup.py develop
```

which install the module through a set of symbolic links to the source directory, allowing users to immediately tests the effects of their changes to the code.

5.3 Running the tests

BoltzTraP2 comes with a comprehensive set of unit and integration tests of its core functionality. To run those, install `pytest` (also available through `pip`), change to the source directory and use the command

```
$ pytest -v tests
```

6.1 Mixing two phases

The mechanism to mix the DOSs between two structures and then calculate the chemical potential is given below

1. shift the DOS using Fermi energy as zero

$$D'_{AFM-b}(\varepsilon) = D_{AFM-b}(\varepsilon + \varepsilon_F^{AFM-b})$$

$$D'_{AFM-a}(\varepsilon) = D_{AFM-a}(\varepsilon + \varepsilon_F^{AFM-a})$$

2. mix the DOSs

$$D'_{\text{mix}}(\varepsilon) = (1 - x) * D'_{AFM-b}(\varepsilon) + x * D'_{AFM-a}(\varepsilon)$$

3. calculate the chemical potential of electrons

$$\int_{-\infty}^{\infty} \mathbf{f} D'_{\text{mix}}(\varepsilon) d\varepsilon = \int_{-\infty}^0 D'_{\text{mix}}(\varepsilon) d\varepsilon$$

under Fermi distribution

$$\mathbf{f} = \frac{1}{e^{\frac{\varepsilon - \mu}{k_B T}} + 1}$$

Usage:

```
python utilities/dosmixAPI.py -d0 dir0/ -d1 dir1/ -nC 11 -nT 101
```

The output to the text file `thermo.out` contains data as functions of phase composition x and T . These data can be plotted following the section [Example by Jupyter Notebook](#)

PENN STATE'S WORK

7.1 Thermodynamic respect of certain thermoelectric quantities

This branch is partially inspired from our recent understanding on the thermoelectric effects. The first is on the [thermodynamic understanding of the Seebeck coefficient](#) which shows that:

Thermoelectric effects, measured by the Seebeck coefficients, refer to the phenomena in which a temperature difference or gradient imposed across a thermoelectric material induces an electrical potential difference or gradient, and vice versa, enabling the direct conversion of thermal and electric energies. All existing first-principles calculations of Seebeck coefficients have been based on the Boltzmann kinetic transport theory. In this work, we present a fundamentally different method for the first-principles calculations of Seebeck coefficients without using any assumptions on the electron scattering mechanism, in contrast to the traditional theory by Cutler and Mott that shows the dependence of the Seebeck coefficient on the scattering mechanisms. It is shown that the Seebeck coefficient is a well-defined thermodynamic quantity that can be determined from the change in the chemical potential of electrons induced by the temperature change and thus can be computed solely based on the electronic density of states through first-principles calculations at different temperatures. The proposed approach is demonstrated using the prototype PbTe and SnSe thermoelectric materials.

The second is [Lorenz Number and Electronic Thermoelectric Figure of Merit: Thermodynamics and Direct DFT Calculations](#) which shows that:

The Lorenz number (L) contained in the Wiedemann-Franz law represents the ratio of two kinetic parameters of electronic charge carriers: the electronic contribution to the thermal conductivity (K_{el}) and the electrical conductivity (σ), and can be expressed as $LT=K_{el}/\sigma$ where T is temperature. We demonstrate that the Lorenz number simply equals to the ratio of two thermodynamic quantities: the electronic heat capacity (c_{el}) and the electrochemical capacitance (c_N) through $LT=c_{el}/c_N$, a purely thermodynamic quantity, and thus it can be calculated solely based on the electron density of states of a material. It is shown that our thermodynamic formulation for the Lorenz number leads to: i) the well-known Sommerfeld value $L=\pi^2/3(k_B/e)^2$ at the low temperature limit, ii) the Drude value $L=3/2(k_B/e)^2$ at the high temperature limit with the free electron gas model, and iii) possible higher values than the Sommerfeld limit for semiconductors. It is also demonstrated that the purely electronic contribution to the thermoelectric figure-of-merit can be directly computed using high-throughput DFT calculations without resorting to the computationally more expensive Boltzmann transport theory to the electronic thermal conductivity and electrical conductivity.

7.2 Revised output for the dope module

For the `interpolation.dope.trace` file, the columns are made of

- collum 1, $\mu - E_f(eV)$ - electron chemical potential
- collum 2, $T(K)$ - temperature
- collum 3, $N(e/uc)$ - number of charge carries due to doping
- collum 4, $DOS(ef)[1/(Ha * uc)]$ -
- collum 5, $S(V/K)$ - Seebeck coefficients by BTE theory
- collum 6, $\sigma/\tau_0[1/(ohm * m * s)]$ - trace of electrical conductivity
- collum 7, $RH[m * *3/C]$ -
- collum 8, $kappa_e/\tau_0[W/(m * K * s)]$ -
- collum 9, $C_\mu[J/(mole - atom * K)]$ - constant voltage heat capacity
- collum 10, $chi[m * *3/mol]$ -
- collum 11, $C_{el}[J/(mole - atom * K)]$ - heat capacity with constant number of eletrons
- collum 12, $S_e(V/K)$ - Seebeck coefficients by thermodynamic understanding, Phys. Rev. B, 98 (2018) 224101.
- collum 13, $n_{eff}(e/cm^3)$ - effective carrier concentration
- collum 14, $L(W * ohm/K * *2)$ - Lorenz number by thermodynamic understanding
- collum 15, σ_h - hole electrical conductivity
- collum 16, σ_e - electon electrical conductivity concentration
- collum 17, N_h - testing
- collum 18, N_e - testing
- collum 19, $n_h(e/cm^3)$ - hole carrier concentration
- collum 20, $n_e(e/cm^3)$ - electron carrier

7.3 Thermodynamic formulations

- Fermi distribution

$$f = \frac{1}{e^{\frac{\varepsilon - \mu}{k_B T}} + 1}$$

where μ is chemical potential, also called Fermi level. This quantity is temperature dependent, to be determined by the number of electrons in the system as

$$N_{el} = \int_{-\infty}^{+\infty} f D(\varepsilon) d\varepsilon = \int_{-\infty}^{\varepsilon_F} D(\varepsilon) d\varepsilon$$

where $D(\varepsilon)$ is the electron density of states defined as

$$D(\varepsilon) = \frac{1}{V} \int \sum_i \delta(\varepsilon - \varepsilon_i(\mathbf{k})) \frac{d\mathbf{k}}{8\pi^3}$$

- Formulations on the thermodynamic theory on the electronic contribution under constant doping conditions
 - internal energy

$$E_{el}(T) = \int f D(\varepsilon) d\varepsilon - \int_{-\infty}^{\varepsilon_F} D(\varepsilon) d\varepsilon$$

- entropy

$$S_{el} = -k_B \int [f \ln f + (1-f) \ln(1-f)] D(\varepsilon) d\varepsilon$$

- free energy

$$F_{el}(T) = E_{el}(T) - TS_{el}$$

- heat capacity

$$C_{el} = \frac{1}{k_B T^2} \left\{ \int f(1-f) [\varepsilon - \mu(T)]^2 D(\varepsilon) d\varepsilon - \frac{[\int f(1-f) [\varepsilon - \mu(T)] D(\varepsilon) d\varepsilon]^2}{\int f(1-f) D(\varepsilon) d\varepsilon} \right\}$$

- Heat capacity under constant voltage condition

$$C_\mu = T \left(\frac{\partial S_{el}}{\partial T} \right)_\mu = \frac{1}{k_B T^2} \left\{ \int f(1-f) [\varepsilon - \mu(T)]^2 D(\varepsilon) d\varepsilon \right\}$$

which is related to its constant doping counterpart by

$$C_\mu = C_{el} + \frac{n_{eff} e^2}{k_B} S_e^2$$

where the effective carrier density is defined as

$$n_{eff} = \int_{-\infty}^{\infty} f(1-f) D(\varepsilon) d\varepsilon$$

while the Seebeck coefficient is thermodynamically determined by

$$S_e = -\frac{1}{en_{eff} T} \int_{-\infty}^{\infty} (\varepsilon - \mu) (1-f) f D(\varepsilon) d\varepsilon$$

Finally, the Lorenz Number can thermodynamically be calculated by

$$L = \frac{k_B}{e^2} \frac{C_{el}}{n_{eff}}$$

7.4 Electrical conductivity

$$\sigma = \frac{1}{k_B T} \int_{-\infty}^{\infty} f(1-f) \mathbf{X}(\varepsilon) d\varepsilon$$

where \mathbf{X} is called transport distribution function, see Madsen, CPC 231}, 140 (2018) and Scheidemantel, PRB 68, 125210(2003)

$$X^{\alpha\beta}(\varepsilon) = \frac{e^2}{V} \int \sum_i v_i^\alpha(\mathbf{k}) v_i^\beta(\mathbf{k}) \tau_{i,\mathbf{k}} \delta(\varepsilon - \varepsilon_i(\mathbf{k})) \frac{d\mathbf{k}}{8\pi^3}$$

where $\tau_{i,\mathbf{k}}$ is relaxation time and the electron group velocity is

$$v_i^\alpha(\mathbf{k}) = \frac{\partial \varepsilon_i(\mathbf{k})}{\partial k^\alpha}$$

- effective carrier mobility

$$\mathbf{M} = \frac{\sigma}{n_{eff} e}$$

7.5 Separation of electron and hole contributions

- electron carrier concentration

$$n = \int_{\varepsilon_F}^{+\infty} f D d\varepsilon$$

- hole carrier concentration

$$p = \int_{-\infty}^{\varepsilon_F} f D d\varepsilon$$

- electrical conductivity

$$\sigma_h = \frac{1}{k_B T} \int_{-\infty}^{\varepsilon_F} f(1-f) \mathbf{X}(\varepsilon) d\varepsilon$$

$$\sigma_e = \frac{1}{k_B T} \int_{\varepsilon_F}^{\infty} f(1-f) \mathbf{X}(\varepsilon) d\varepsilon$$

- mobility

$$\mathbf{M}_h = \frac{\sigma_h}{ep}$$

$$\mathbf{M}_e = \frac{\sigma_e}{en}$$

CHANGELOG

8.1 0.0 (2021-01-27)

(Contributor: @YiWang)

- Change List:
- Changes are made only for constant doping calculations, partially according to Phys. Rev. B, 98 (2018) 224101. The major works include:
 1. revised interface.py
 2. extended bandlib.py into bandlibEXT.py.
 3. extended io.py into ioEXT.py
 4. added remesh.py for calculations in low temperature (meshed increased to ~100,000).
 5. improved the computational efficiency on the calculations of the chemical potential of electrons (Fermi levels). See line 332-335 in `bandlibEXT.py`.
 6. added plotpng.py for figure plot.

ACKNOWLEDGEMENTS

This work is supported the Computational Materials Sciences Program funded by the US Department of Energy, Office of Science, Basic Energy Sciences, under Award Number DE-SC0020145. First-principles calculations were carried out partially on the resources of NERSC supported by the Office of Science of the US Department of Energy under contract No. DE-AC02-05CH11231, and partially on the resources of Extreme Science and Engineering Discovery Environment (XSEDE) supported by NSF with Grant No. ACI-1053575.

INDICES AND TABLES

- genindex
- modindex
- search