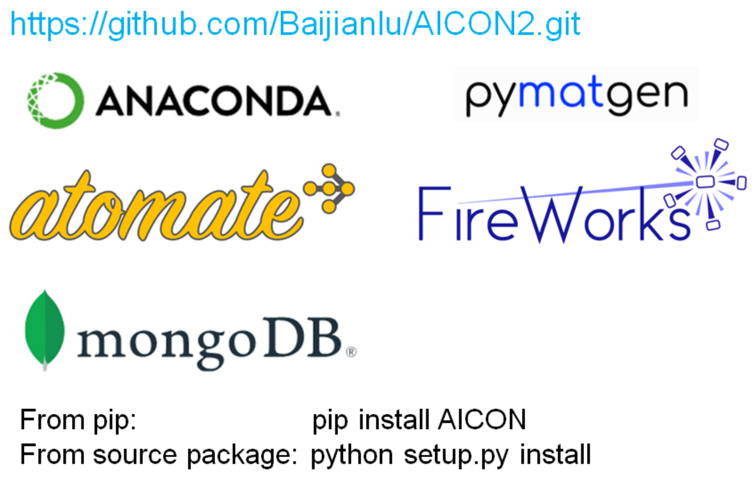
LECTURE 12. TUTORIAL WITH THE AICON CODE

Hello, everyone. Welcome to my third lecture. In this lecture, I am going to introduce how to use AICON2 Pro to calculate the transport properties and how to use BoltzTrap2 to calculate the transport properties. This presentation tells you how to do it. You can download it and try it yourselves.

AICON rely on several external programs to achieve its functions, such as scipy, pymatgen. The installation of AICON itself is very easy, the main difficulty comes from installing and configuring these libraries correctly. So before install AICON, these program should be installed first. The former two are used to do mathematical operation and data process, while the last three are used in automatic workflow control.



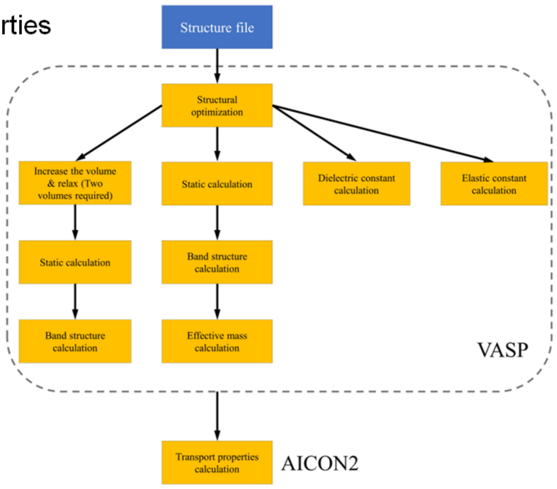
https://github.com/Baijianlu/AICON2.git

From pip: pip install AICON

From source package: python setup.py install

This picture shows a whole workflow of using AICON to calculate transport properties. Before calling AICON, these first-principles calculations should be finished first. As can be seen, at least 13 calculations need to be run. Some will be simple and will cost a short time, while some of them will be a little bit complex.

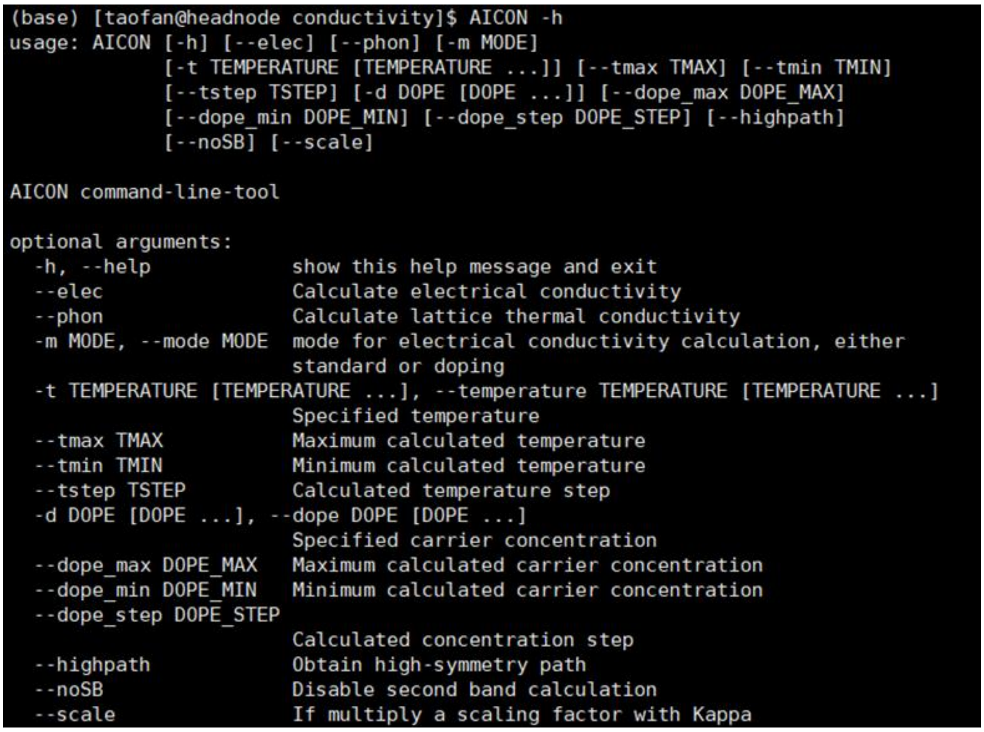
Electronic transport properties


Structure file
Structural optimization
Increase the volume & relax (Two volumes required)
Static calculation
Dielectric constant calculation
Elastic constant calculation
Static calculation
Band structure calculation
Band structure calculation
Effective mass calculation
VASP
Transport properties calculation
AICON2

After finishing all these first principles calculations, the results have to be collected into corresponding directories named exactly as this list shows. Basically, you need three band structures calculated at different volumes, effective mass at CBM and VBM, and second valence or conduction band if it exists. Dielectric constant and elastic constants.

equi	folder of equilibrium structure’s band structure
0.5per	folder of 0.1% larger structure’s band structure
1.0per	folder of 0.2% larger structure’s band structure
CBM	folder of conduction band minimum calculation
VBM	folder of valence band maximum calculation
VSB	folder of second valence band calculation
dielect	folder of dielectric constant calculation
elastic	folder of elastic constant calculation

You can use AICON –h to display the help information. I think I have given you clear explanation for each argument. There is one argument I may need to explain more, this –m mode. There are two kinds of mode for electrical conductivity calculation:



```
(base) [taofan@headnode conductivity]$ AICON -h
usage: AICON [-h] [--elec] [--phon] [-m MODE]
             [-t TEMPERATURE [TEMPERATURE ...]] [--tmax TMAX] [--tmin TMIN]
             [--tstep TSTEP] [-d DOPE [DOPE ...]] [--dope_max DOPE_MAX]
             [--dope_min DOPE_MIN] [--dope_step DOPE_STEP] [--highpath]
             [--noSB] [--scale]

AICON command-line-tool

optional arguments:
  -h, --help            show this help message and exit
  --elec                Calculate electrical conductivity
  --phon                Calculate lattice thermal conductivity
  -m MODE, --mode MODE  mode for electrical conductivity calculation, either
                        standard or doping
  -t TEMPERATURE [TEMPERATURE ...], --temperature TEMPERATURE [TEMPERATURE ...]
                        Specified temperature
  --tmax TMAX           Maximum calculated temperature
  --tmin TMIN           Minimum calculated temperature
  --tstep TSTEP         Calculated temperature step
  -d DOPE [DOPE ...], --dope DOPE [DOPE ...]
                        Specified carrier concentration
  --dope_max DOPE_MAX   Maximum calculated carrier concentration
  --dope_min DOPE_MIN   Minimum calculated carrier concentration
  --dope_step DOPE_STEP
                        Calculated concentration step
  --highpath            Obtain high-symmetry path
  --noSB                Disable second band calculation
  --scale               If multiply a scaling factor with Kappa
```

--mode (-m) “standard” or “doping”

standard	doping
➢ Properties as a function of temperature and chemical potential ➢ Chemical potential starts from the middle of the gap and till the 1 eV above with a step of 2 meV	➢ Properties as a function of temperature and doping concentration.

Here shows examples of using AICON under standard and doping mode respectively. This is using AICON by command-line way. AICON’s function can also be called by a python script.

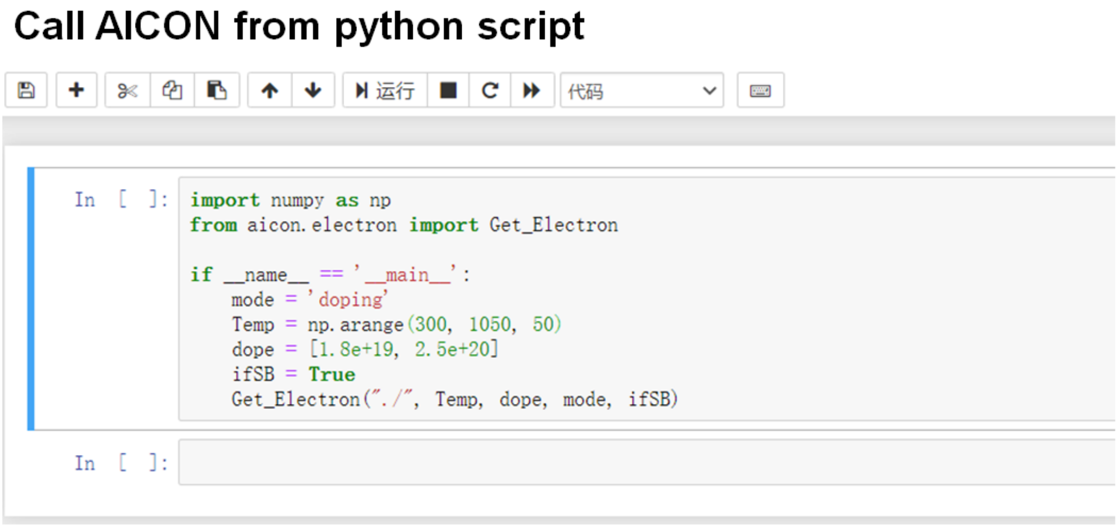
“standard” mode

```
AICON --elec -m standard --tmin 300 --tmax 1000 --tstep 100
```

“doping” mode

```
AICON --elec -m doping --tmin 300 --tmax 1000 --tstep 100 -d 1e+18 1e+19
```

Just several lines, clear and easy.



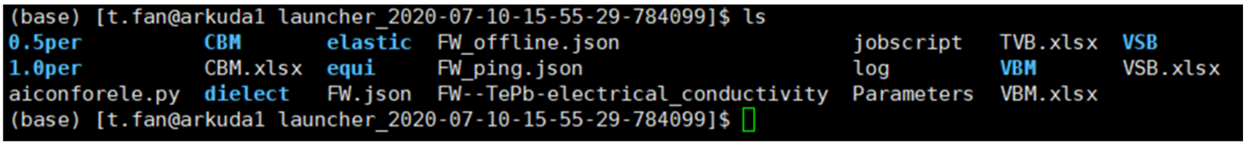
Call AICON from python script

```
In [ ]: import numpy as np
        from aicon.electron import Get_Electron

        if __name__ == ’__main__’:
            mode = ’doping’
            Temp = np.arange(300, 1050, 50)
            dope = [1.8e+19, 2.5e+20]
            ifSB = True
            Get_Electron(″./″, Temp, dope, mode, ifSB)

In [ ]:
```

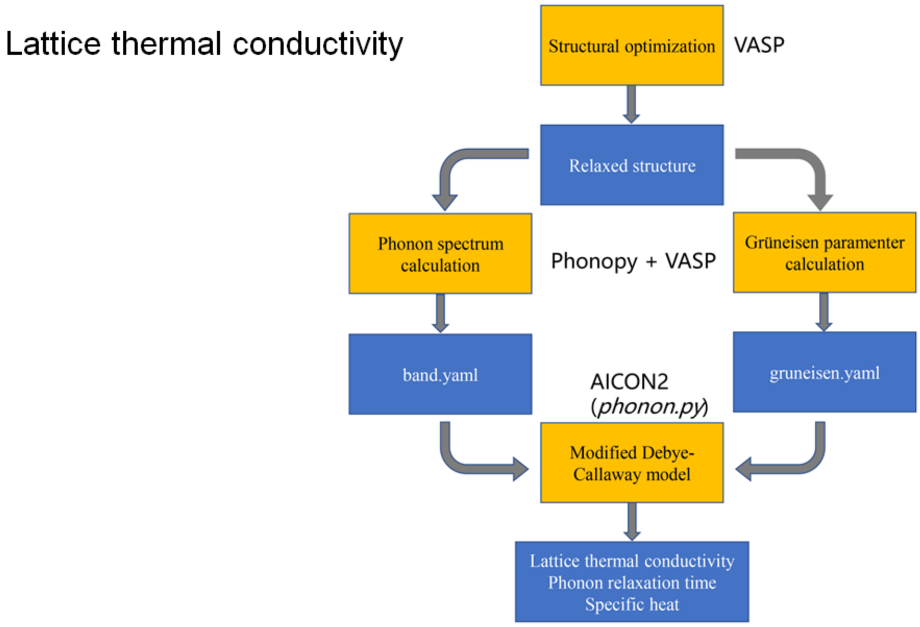
After AICON’s calculation are finished, there will be several files containing the results. This Parameter file contains all key parameters including extracted from first principles calculation. Including effective mass at CBM and VBM, band gap, deformation potential constants, etc. these excel files contain transport coefficients. However, since the data is first stored as a pandas.dataframe object, it can be saved as any file format pandas supports. Users can change it according to their needs. Let’s see an example of this file.



```
(base) [t.fan@arkudal launcher_2020-07-10-15-55-29-784099]$ ls
0.5per          CBM        elastic  FW_offline.json                 jobscript   TVB.xlsx  VSB
1.0per          CBM.xlsx   equi     FW_ping.json                    log         VBM       VSB.xlsx
aiconforele.py  dielect    FW.json  FW--TePb-electrical_conductivity  Parameters  VBM.xlsx
(base) [t.fan@arkudal launcher_2020-07-10-15-55-29-784099]$
```

❖ transport properties’s data are stored as a pandas.DataFrame object and it could be saved as any file format pandas supports.

For lattice thermal conductivity, this picture shows the process for calculating it. We start from a relaxed structure, calculate its phonon spectrum and gruneisen parameters with DFPT or finite displacement method by phonopy plus VASP. Then get the results files band.yaml and gruneisen.yaml. Use these two files with POSCAR file as inputs of AICON and get the lattice thermal conductivity values.


Lattice thermal conductivity
Structural optimization
VASP
Relaxed structure
Phonon spectrum calculation
Phonopy + VASP
Grüneisen paramenter calculation
band.yaml
gruneisen.yaml
AICON2 (phonon.py)
Modified Debye-Callaway model
Lattice thermal conductivity
Phonon relaxation time
Specific heat

During this process, you need to run AICON twice. The first time is for getting the high-symmetry path, which is used as input of Phonopy in order to calculate phonon spectrum and gruneisen parameters. The second time is calculating the kappa values. The command is simple and clear.

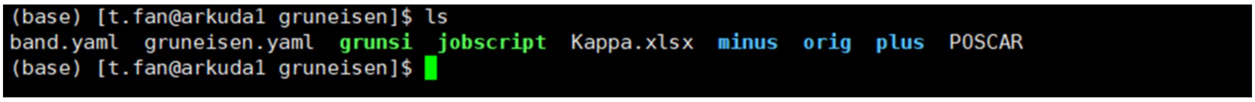
Step1: get high-symmetry path

```
AICON --phon --highpath
```

Step 2: calculate lattice thermal conductivity

```
AICON --phon --tmin 300 --tmax 1000 --tstep 50
```

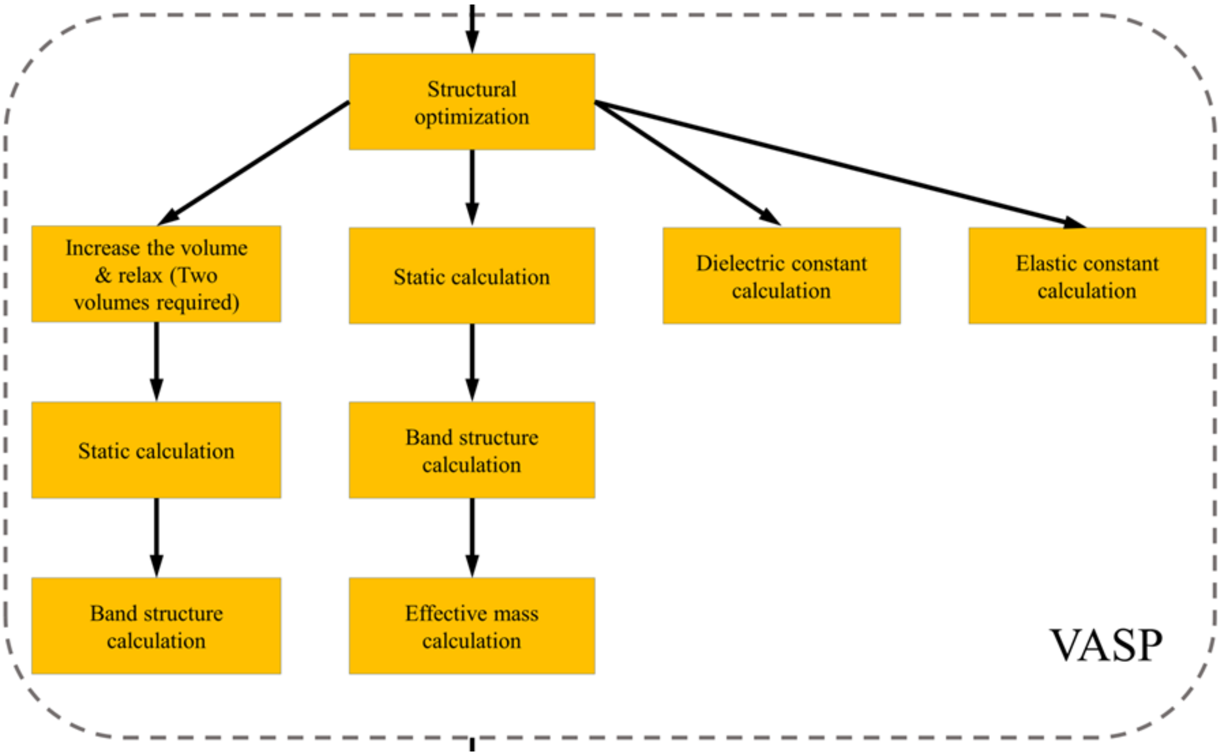
After the job finished, you will get an excel file named Kappa, which contains lattice thermal conductivity in the temperature range you want, also contains phonon relaxation time for three acoustic phonon branches and one optical phonon branch. Let’s see how it looks like.



```
(base) [t.fan@arkuda1 gruneisen]$ ls
band.yaml  gruneisen.yaml  grunsi  jobscript  Kappa.xlsx  minus  orig  plus  POSCAR
(base) [t.fan@arkuda1 gruneisen]$ 
```

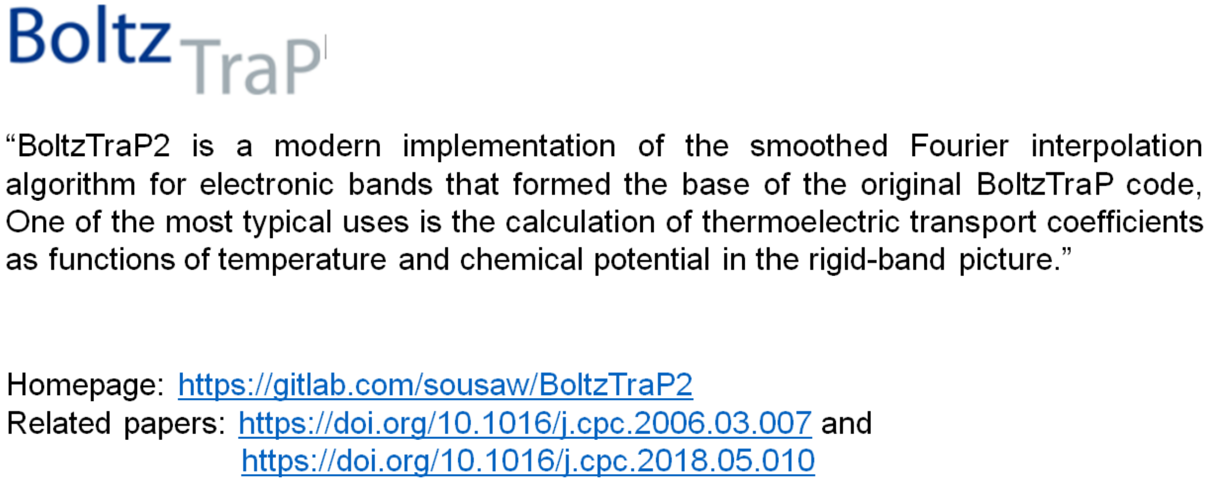
❖ transport properties's data are stored as a pandas.DataFrame object and it could be saved as any file format pandas supports.

As I already said, before calling AICON, users need to finish these first principles calculations. These calculations follow some order, have varying complexity and take different time. Managing the workflow could be painful, especially when doing high-throughput screens. That's why I also implement an automatic workflow control tool in AICON. I design a workflow specified for AICON's calculation. The users just need to submit the job once and the following calculation will run automatically. Let me show you how this is achieved.


Structural optimization
Increase the volume & relax (Two volumes required)
Static calculation
Dielectric constant calculation
Elastic constant calculation
Static calculation
Band structure calculation
Band structure calculation
Effective mass calculation
VASP

In this next section, I will briefly show you how to run BoltzTrap2 to calculate thermoelectric transport properties.

BoltzTraP2 is a modern implementation of the smoothed Fourier interpolation algorithm for electronic bands that formed the base of the original BoltzTraP code, One of the most typical uses is the calculation of thermoelectric transport coefficients as functions of temperature and chemical potential in the rigid-band picture. Here I need to emphasize again that BoltzTrap uses constant relaxation time approximation,, it assumes the electronic relaxation time is non-related with band index and wavevector. This however, will cause a problem that for some results of Boltztrap, such as electrical conductivity and electronic thermal conductivity, they are actually not the results we expect, since they contain this \tau implicitly. That’s why BoltzTrap calculation is much faster, but also means users need to get these \tau values from other methods. Here I listed the link of its homepage and two papers introduced its theory and utility.

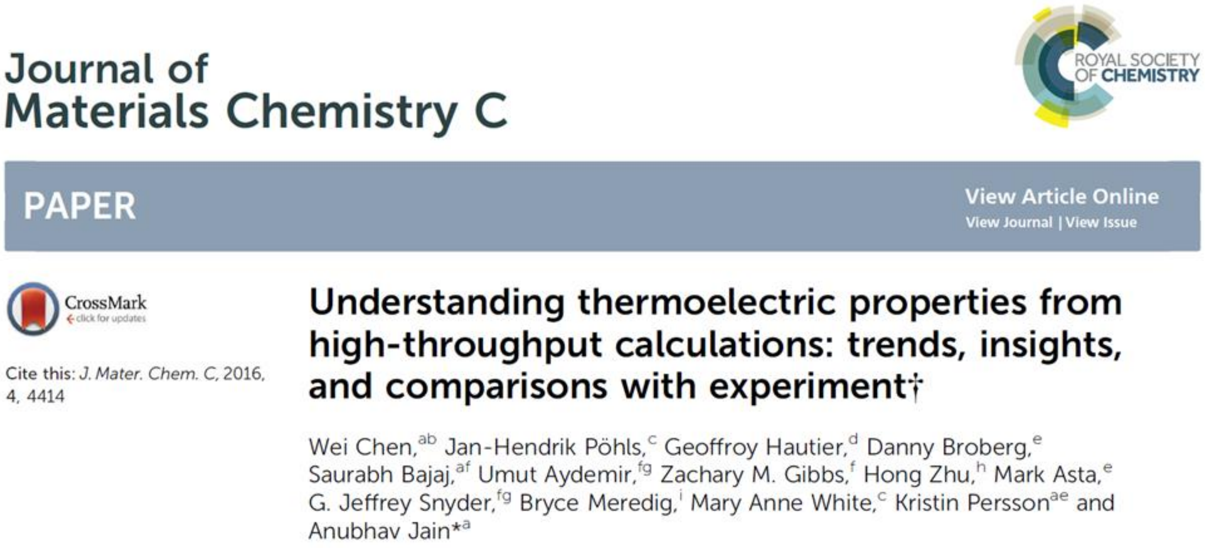


Boltz TraP

“BoltzTraP2 is a modern implementation of the smoothed Fourier interpolation algorithm for electronic bands that formed the base of the original BoltzTraP code, One of the most typical uses is the calculation of thermoelectric transport coefficients as functions of temperature and chemical potential in the rigid-band picture.”

Homepage: https://gitlab.com/sousaw/BoltzTraP2
Related papers: https://doi.org/10.1016/j.cpc.2006.03.007 and
https://doi.org/10.1016/j.cpc.2018.05.010

Since BoltzTrap being published, there has been a lot of high-throughput works using it to screen potential thermoelectric materials. The largest scale one is this work. This project is a part of the materials project database, they used BoltzTrap calculated nearly 50 thousand compounds’ transport coefficients. The data is open source and people can check and download these data.



Journal of Materials Chemistry C

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

CrossMark
click for updates

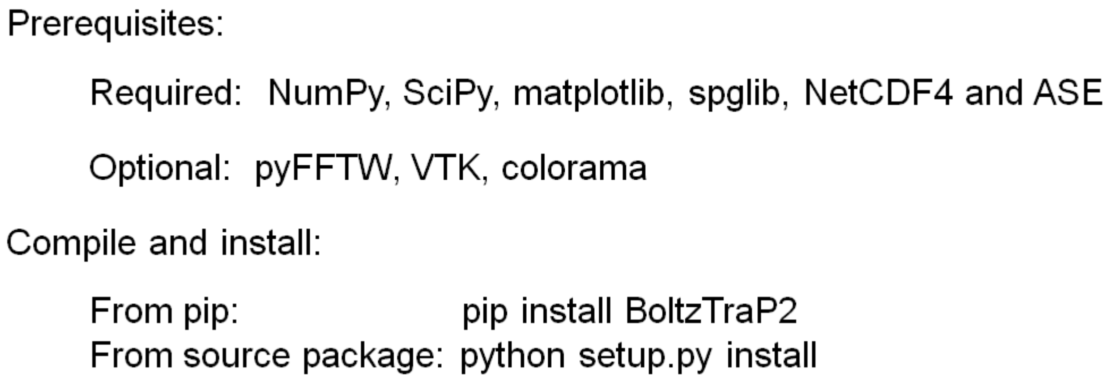
Cite this: *J. Mater. Chem. C*, 2016, 4, 4414

Understanding thermoelectric properties from high-throughput calculations: trends, insights, and comparisons with experiment†

Wei Chen,[ab] Jan-Hendrik Pöhls,[c] Geoffroy Hautier,[d] Danny Broberg,[e] Saurabh Bajaj,[af] Umut Aydemir,[fg] Zachary M. Gibbs,[f] Hong Zhu,[h] Mark Asta,[e] G. Jeffrey Snyder,[fg] Bryce Meredig,[i] Mary Anne White,[c] Kristin Persson[ae] and Anubhav Jain*[a]

https://contribs.materialsproject.org/projects/carrier_transport

The installation of BoltzTrap is relatively simple. The dependent packages are almost the same as those of AICON. You can install it either from pip or from the source package.



Prerequisites:

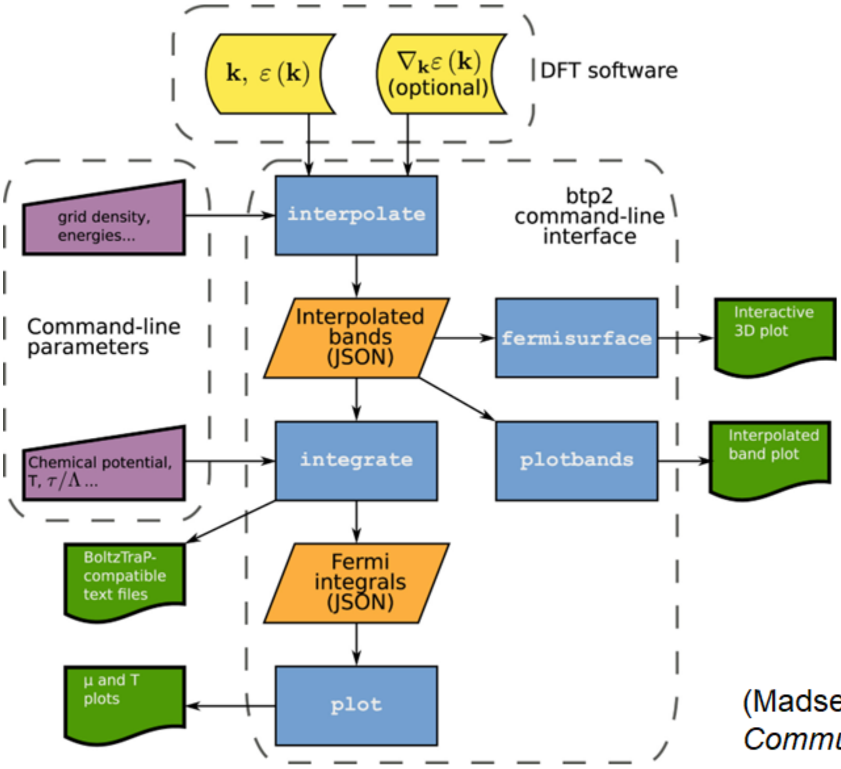
Required: NumPy, SciPy, matplotlib, spglib, NetCDF4 and ASE

Optional: pyFFTW, VTK, colorama

Compile and install:

```
From pip:                 pip install BoltzTraP2
From source package:      python setup.py install
```

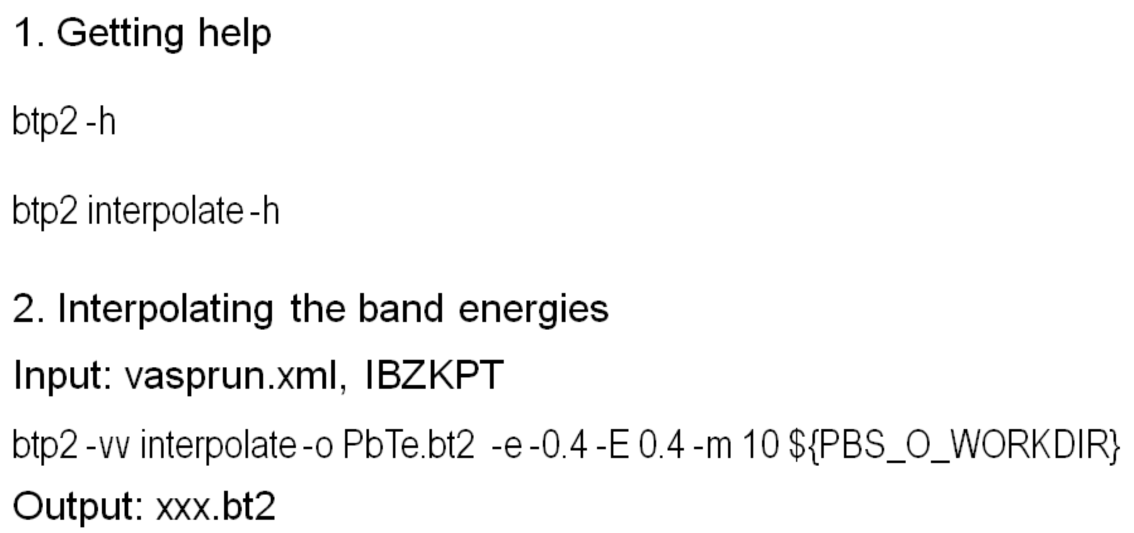
This is the workflow of BoltzTrap2, starting from DFT calculated wavevector, band energy and band velocity, first do the interpolation work to get the band energy on a extremely dense mesh. Then do the Fermi integrals to obtain transport coefficients. There are also other functions such as plotting fermi surface or plotting energy bands. Users who are interested in these functions can read its manual.



$\mathbf{k},\ \varepsilon(\mathbf{k})$
$\nabla_{\mathbf{k}}\varepsilon(\mathbf{k})$ (optional)
DFT software
grid density, energies...
interpolate
btp2 command-line interface
Command-line parameters
Interpolated bands (JSON)
fermisurface
Interactive 3D plot
Chemical potential, T, τ/Λ...
integrate
plotbands
Interpolated band plot
BoltzTraP-compatible text files
Fermi integrals (JSON)
µ and T plots
plot

(Madsen et al, *Computer Physics Communications*, 2018)

Here I show how to use BoltzTrap2 in a command-line way. Use btp2 –h get the help information, or btp2 –h get the help information of the interpolation function. Second, for interpolating the band energies the required inputs are vasprun.xml, IBZKPT if using VASP for DFT calculation. Then run the command, there are only several variables that users need to specify. –o specify the output file name. lowercase and uppercase E specify the energy window around fermi level in which bands are taken into consideration. The unit is Hartree. –m specify multiplying factor for the number of irreducible k points in the interpolation, so here 10 means the final interpolated mesh is ten times denser than that existing in IBZPKT file. Finally, the working directory should be given. The output file is a .bt2 file, .bt2 files are just xz-compressed JSON documents, which can be visualized and manipulated using a multitude of tools and libraries. Usually users just need to treat this file as a black box.



1. Getting help

```
btp2 -h
```

```
btp2 interpolate -h
```

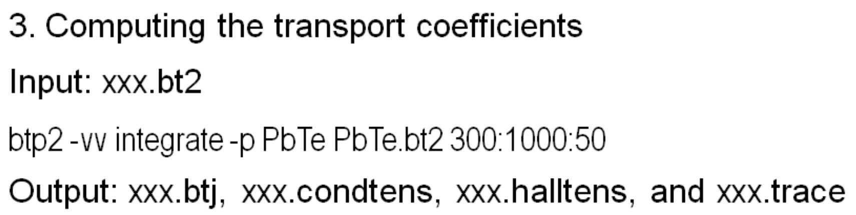
2. Interpolating the band energies

Input: vasprun.xml, IBZKPT

```
btp2 -vv interpolate -o PbTe.bt2  -e -0.4 -E 0.4 -m 10 ${PBS_O_WORKDIR}
```

Output: xxx.**bt2**

Third step, computing the transport coefficients, the required input file is the .bt2 file from the last step. Then run the command, here –p is specifying the name of output files, then followed by the interpolation file from the last step and the interested temperature range. The output files include a .btj file, .condtens, .halltens, and .trace files. The last three are closely following the formats defined by the original BoltzTraP. They can be visualized with any text editor, loaded into any data processing code or plotted with almost any software on the market. This table lists the first line of .trace file, it represents the meaning and unit of each column.

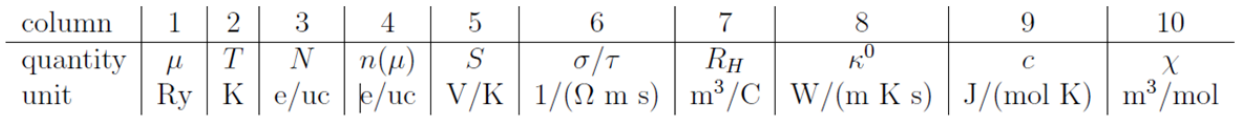


3. Computing the transport coefficients

Input: xxx.**bt2**

```
btp2 -vv integrate -p PbTe PbTe.bt2 300:1000:50
```

Output: xxx.**btj**, xxx.**condtens**, xxx.**halltens**, **and** xxx.**trace**



column	1	2	3	4	5	6	7	8	9	10
quantity	μ	T	N	$n(\mu)$	S	σ/τ	R_H	κ^0	c	χ
unit	Ry	K	e/uc	e/uc	V/K	$1/(\Omega$ m s)	m^3/C	W/(m K s)	J/(mol K)	$\mathrm{m}^3/\mathrm{mol}$

Next, I am gonna show you how to call BoltzTrap2 and analyze its results from a python script.