

# BoltzTraP2: A program for interpolating band structures and calculating semi-classical transport coefficients.

April 22, 2020

## 1 Authors Copyright

Copyright (C) 2017-2019 Georg K. H. Madsen: [georg.madsen@tuwien.ac.at](mailto:georg.madsen@tuwien.ac.at)  
Copyright (C) 2017-2019 Jesús Carrete: [jesus.carrete.montana@tuwien.ac.at](mailto:jesus.carrete.montana@tuwien.ac.at)  
Copyright (C) 2017-2019 Matthieu J. Verstraete: [matthieu.verstraete@ulg.ac.be](mailto:matthieu.verstraete@ulg.ac.be)  
Copyright (C) 2018-2019 Genadi Naydenov: [gan503@york.ac.uk](mailto:gan503@york.ac.uk)

## 2 License

This file is part of BoltzTraP2. BoltzTraP2 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

BoltzTraP2 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with BoltzTraP2. If not, see <http://www.gnu.org/licenses/>.

## 3 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 article](#).

## 4 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 Compiling and install BoltzTraP2

The easiest way to get BoltzTraP2 is to run:

```
1 pip3 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:

```
1 python setup.py install
```

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

```
1 python setup.py --help
2 python setup.py --help-commands
3 python setup.py install --help
```

The BoltzTraP2 installer supports

```
1 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.

## 6 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

```
1 pytest -v tests
```

## 7 Command lines help

```
1 btp2 -h
```

**usage:** `btp2 [-h] [-v] [-n NWORKERS] [-V] command [options] ... ..`

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

optional arguments:

-h, --help show this help message and exit

-v, --verbose increase the verbosity level (default: 0)

-n NWORKERS, --nworkers NWORKERS number of processes to span for parallel operations (default: 1)

-V, --version print version information

**available subcommands:**

`command [options] ...`

<b>interpolate</b>	interpolate the DFT input and save the results
<b>integrate</b>	compute transport coefficients based on the interpolated bands
<b>dope</b>	compute transport coefficients for particular carrier concentrations
<b>plotbands</b>	create a plot of the interpolated band structure
<b>plot</b>	create a plot of the integration results
<b>describe</b>	provide basic information about a .bt2 or a .btj file
<b>fermisurface</b>	create an interactive visualization of the Fermi surface