mDCThermalC User Manual

mDCThermalC, namely **m**odified **D**ebye-**C**allaway model for **Thermal C**onductivity calculation, is used for calculating lattice thermal conductivity with semi-empirical method. It can be obtained from https://github.com/Baijianlu/mDCThermalC.git. We have a published paper described the method behind this software detailly[1]. If you publish results obtained using the program then this paper should be cited.

For now, the program relies on external programs to generate input files it needs (POSCAR, band.yaml, gruneisen.yaml), we recommend VASP + Phonopy, but it can easily be interfaced to other codes. Please contribute if you make such an interface.

1 Getting started

1.1 Unpacking

Run the command: unzip mDCThermalC-1.1.1.zip
This will unpack several directories

Scripts	executable scripts of the program
examples	several examples for testing, including diamond, silicon, SnSe
mdcthermalc	source codes directory
doc	useful documents, including this manual
LICENSE	GPL-3.0 license
README.md	description of the program
setup.py	install script

1.2 Installation

mDCThermalC is a Python module. mDCThermalC's runtime requirements are Python version 3.5 or higher, and the Python libraries NumPy, SciPy, spglib and pymatgen. 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. Here we recommend to use Anaconda so that dependencies should be resolved automatically.
After the dependent libraries being installed, enter into the package directory and run:

```
python setup.py install
```

you can also specify the installing directory by running like this:

```
python setup.py install --prefix=/your_dir
```

However, this requires the directory you specified is on the python path or else it may report errors.
The installing process should be quick. If there is no error happening during installation and it ends up normally, you may find an executable named **mDCThermalC** in

directory like *~/anaconda3/bin/*, or the directory you specified.

1.3 Running test examples

The distribution includes four examples: diamond, Si, SnSe, Mg_2Si. The former three examples are also described in the mDCThermalC paper[1]. Here we use diamond as an example to show the calculating procedure step by step.

Suppose you have already gotten the second order force constant file using Phonopy, the next thing is obtaining band.yaml file which contains information of phonon frequency and phonon velocity. There **must** be a structural file in your directory named POSCAR(VASP format unit cell) and then run the command:

```
mDCThermalC -p
```

it will show some message like this,

```
Please set "BAND" parameter of phonopy as this:

0.3750 0.3750 0.7500  0.0000 0.0000 0.0000  0.5000 0.5000 0.5000  0.6250 0.2500 0.6250  0.0000 0.0000 0.0000  0.500
0 0.2500 0.7500  0.5000 0.0000 0.5000  0.0000 0.0000 0.0000

We notice your structure could have a primitive cell. Please set "PRIMITIVE_AXIS" parameter of phonopy as this:

0.00000000 0.50000000 0.50000000  -0.50000000 0.00000000 -0.50000000  -0.50000000 -0.50000000 0.00000000

(base) [Tao.Fan@frontend gruneisen]$
```

Set the Phonopy input file(such as band.conf) as the message said, for diamond:

```
ATOM_NAME = C
DIM = 2 2 2
PRIMITIVE_AXIS = 0.0 0.5 0.5  -0.5 0.0 -0.5  -0.5 -0.5 0.0
BAND = 0.3750 0.3750 0.7500  0.0000 0.0000 0.0000  0.5000 0.5000 0.5000  0.6250 0.2500 0.6250  0.0000 0.0000 0.0000
.5000 0.2500 0.7500  0.5000 0.0000 0.5000  0.0000 0.0000 0.0000
FORCE_CONSTANTS = READ
GROUP_VELOCITY = .TRUE.
```

Then run the command like this:

```
phonopy -c POSCAR-unitcell band.conf
```

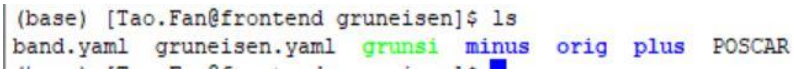
You will get the band.yaml file if no error occurs. The next thing is obtaining the gruneisen.yaml file. Suppose you have already finished all three needed calculation (one at the equilibrium volume, one at the slightly small volume, another at the slightly large volume. Details about how to calculate Grüneisen parameters can be seen here: http://atztogo.github.io/phonopy/gruneisen.html#phonopy-gruneisen). Then run:

```
phonopy-gruneisen orig plus minus --dim="2 2 2" --pa="0 1/2 1/2 -1/2 0 -1/2 -1/2 -1/2
0" --band="0.3750 0.3750 0.7500  0.0000 0.0000 0.0000  0.5000 0.5000 0.5000
0.6250 0.2500 0.6250  0.0000 0.0000 0.0000  0.5000 0.2500 0.7500  0.5000
0.0000 0.5000  0.0000 0.0000 0.0000" --readfc -c POSCAR-unitcell
```

Attention! The high symmetry path and primitive cell transform matrix should be exactly the same as in the band.conf file.

You will get gruneisen.yaml file if no error occurs. Here all input files needed by

mDCThermalC to calculate thermal conductivity have already been obtained. You should put them in the same directory like this:



Then run the command:

```
mDCThermalC -t 200:50:1000
```

Note "-t 200:50:1000" means temperature range from 200K to 1000K and sample every 50K. you can specify any temperature or temperature range you want to calculate by similar grammer.

Attention! Due to the methods we used, the results of thermal conductivity at very low temperature(< 100K) cannot be guaranteed. According to our test, the results more than 200K should be reliable.

The results are written into the file named "kappa", and this file is quite simple. The first column is temperature, the second column is thermal conductivity value, the third and fourth column is specific heat ratio defined in our paper[1], the rest columns are branch phonon relaxation time of different scattering processes (Normal, Umklapp and isotope defect).

2 Frequently asked questions

Reference

[1]