



Advanced workshop

Roberta Farris

Psi-k and CECAM Flagship Workshop
Barcelona, 2-5 June 2025



Institut Català
de Nanociència
i Nanotecnologia



CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS

CSIC



TDEP + SIESTA Tutorial

What do we need?

- Siesta installed
- TDEP installed



[Home](#) [What is SIESTA?](#) [Getting the code](#) [Documentation](#) [Support](#) [News](#) [Events](#) [The Team](#) [For Developers](#)

Advanced SIESTA Workshop 2025: Sessions unfolded

Software

The hands-on sessions will make use of the following software:

- [SIESTA 5.4](#)
- [TDEP](#)
- [TB2J](#)
- [sisl](#)
- [jupyter-notebooks/lab](#)
- [flos](#)
- [graph2mat](#)

Programme

All times are CEST (UTC+2). Color codes: [talk + hands-on](#), [talk / showcase](#), [discussion](#), [break / food event](#), [other](#)

Temperature Dependent Effective Potentials (TDEP)

joss 10.21105/joss.06150 release v25.03 license MIT Compile TDEP and test binaries passing DOI 10.5281/zenodo.15096483

Briefly summarized, the package provides all the tools you need to build accurate model Hamiltonians for finite temperature lattice dynamics from first principles. TDEP includes several programs for different tasks:

Installation

Please find installation instructions in the TDEP repository.

Table of contents

Tutorials

Installation

Report bugs and issues

Contribute

Other things to look at

How to cite

Troubleshooting

Symmetry errors

Anaconda

One convenient, (mostly) platform-agnostic way to install TDEP is to use [Anaconda](#).

Prepare environment

Create a `conda` environment with

```
conda create --prefix /path/to/where/you/want/to/create/the/conda_env python=3.10
```



Activate:

```
conda activate /path/to/where/you/want/to/create/the/conda_env/
```



Install dependencies

```
conda install -c conda-forge gfortran openmpi-mpifort scalapack fftw hdf5
```



Install

Copy `./examples/build/important_settings.conda` to `important_settings` and adjust the `PREFIX`, i.e.,

```
...  
PREFIX=/path/to/where/you/want/to/create/the/conda_env
```



Let's try!

MgO test case

Input files:

- `infile.ucposcar`
- `infile.forceconstant`
- `infile.forceconstant_thirdorder`

MgO test case

- **infile.ucposcar:** Defines the primitive cell of your system (lattice vectors and equilibrium positions), in the POSCAR format:

```
Mg O
1.0000000000000000
  0.0000000000000000  2.1194564144000001  2.1194564144000001
  2.1194564144000001  0.0000000000000000  2.1194564144000001
  2.1194564144000001  2.1194564144000001  0.0000000000000000
Mg O
1 1
Direct
0.0000000000000000 0.0000000000000000 -0.0000000000000000
0.5000000000000000 0.5000000000000000 0.5000000000000000
```

First step: Generate a supercell

```
$ tdep/bin/generate_structure -na 216
```

outfile.ssposcar

non-diagonal supercell

4.238912828800

0.000000000000000 0.000000000000000 3.000000000000000

0.000000000000000 3.000000000000000 0.000000000000000

-3.000000000000000 0.000000000000000 0.000000000000000

Mg O

108 108

Direct coordinates

0.833333333333333 0.500000000000000 0.000000000000000 site 1 species 1: Mg

0.833333333333333 0.333333333333333 0.166666666666667 site 2 species 1: Mg

0.833333333333333 0.166666666666667 0.333333333333333 site 3 species 1: Mg

First step: Generate a supercell

```
$ cp outfile.ssposcar infile.ssposcar
```

Second step: Create canonical configurations

```
$ tdep/bin/canonical_configuration --temperature 300 --quantum -n 4 --  
output_format 5
```

If you are curious about the different options:

```
$ tdep/bin/canonical_configuration -h
```

Second step: Create canonical configurations

siesta_conf0001
siesta_conf0002
siesta_conf0003
siesta_conf0004
siesta_conf0001.XV
siesta_conf0002.XV
siesta_conf0003.XV
siesta_conf0004.XV

Third step: Get the IFCs

We should run SIESTA and then, from the forces, we extract the IFCs :

```
$ ~/tdep/bin/extract_forceconstants -rc2 6 -rc3 4
```

We are not going to do this for lack of time!!!!

The needed inputs:

Input files:

- `infile.ucposcar`
- `infile.forceconstant`
- `infile.forceconstant_thirdorder`

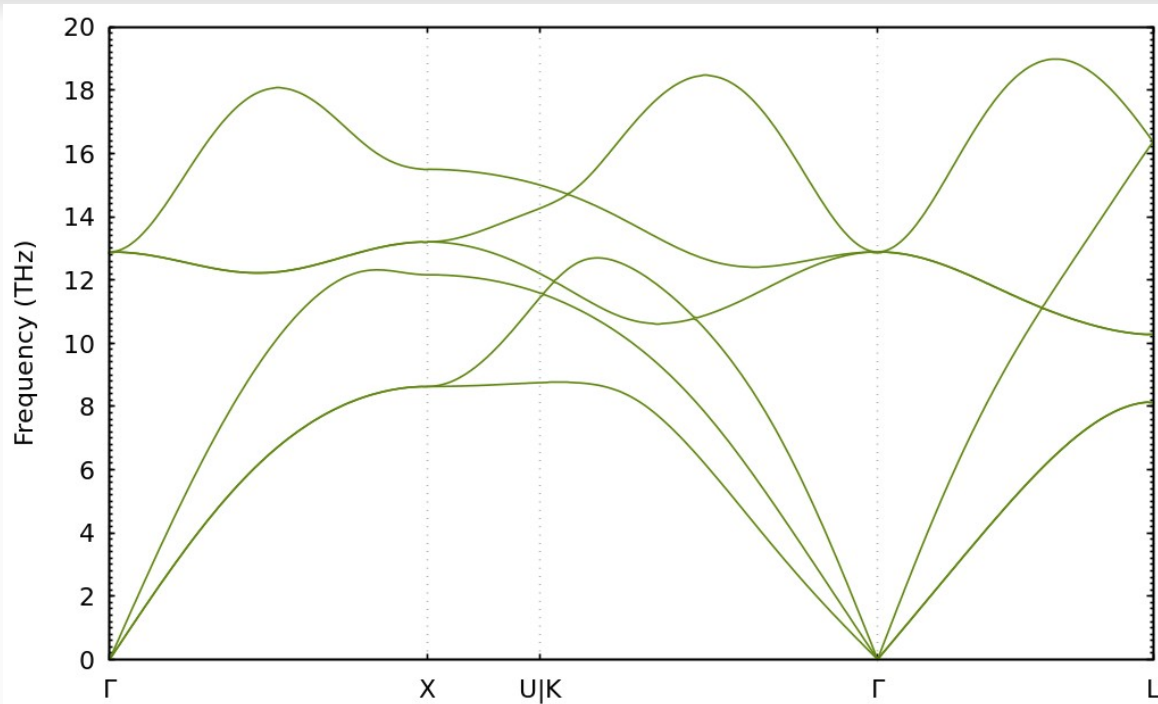
Fourth step: Phonons

```
$ tdep/bin/phonon_dispersion_relations
```

Plot as you wish, for example:

```
$ gnuplot --persist outfile.dispersion_relations.gnuplot
```


Fourth step: Phonons



Fifth step: Thermal conductivity

```
$ mpirun thermal_conductivity -qg 4 4 4 > log.thermal &
```

Inspect the optional flags :

```
$ thermal_conductivity -h
```

Fifth step: Thermal conductivity

\$ vi outfile.thermal_conductivity

```
Unit:          W/m/K
# Temperature:    0.300000000000E+03
# Single mode approximation
#      kxx      kyy      kzz      kxy      kxz      kyz
#      0.321196286146E+02  0.321196286146E+02  0.321196286146E+02  0.000000000000E+00  0.000000000000E+00  0.000000000000E+00
# Collective contribution
#      kxx      kyy      kzz      kxy      kxz      kyz
#      0.117930198065E+01  0.117930198065E+01  0.117930198065E+01  0.000000000000E+00  0.000000000000E+00  0.000000000000E+00
# Off diagonal (coherence) contribution
#      kxx      kyy      kzz      kxy      kxz      kyz
#      0.344889044599E+00  0.344889044599E+00  0.344889044599E+00  0.000000000000E+00  0.000000000000E+00  0.000000000000E+00
# Total thermal conductivity
#      kxx      kyy      kzz      kxy      kxz      kyz
#      0.336438196399E+02  0.336438196399E+02  0.336438196399E+02  0.000000000000E+00  0.000000000000E+00  0.000000000000E+00
```

For more examples:

https://github.com/tdep-developers/tdep-tutorials/tree/main/04_thermal_conductivity

Questions?