Guide to `clattices` 1.0

Daniel S. Koda[*]
Marcelo Marques[†]
Lara K. Teles[‡]
Instituto Tecnológico de Aeronáutica
Group of Semiconductor Materials and Nanotechnology

Friedhelm Bechstedt [§]
Friedrich-Schiller-Universität
Institut für Festkörpertheorie und -optik

June 2016

Abstract

`clattices` is a software for finding coincidence lattices of 2D crystals made in Python. Its goals are to simplify simulations of heterostructure combinations by optimizing supercell sizes, to allow for better investigations concerning interlayer twist, to improve experimental and theoretical contributions and to provide material for advances on theoretical investigations. This tool has been developed within the Group of Semiconductor Materials and Nanotechnology (GMSN) of Brazil's Instituto Tecnológico de Aeronáutica. This document describes the capabilities of the software and provides a beginner's guide to its utilization. We cover the installation of the script and demonstrate the use of this tool with examples accompanying the software package.

Contents

[*] `danielskoda@gmail.com`
[†] `mmarques@ita.br`
[‡] `lkteles@ita.br`
[§] `friedhelm.bechstedt@uni-jena.de`

1 Introduction

1.1 About `clattices`

Origins: `clattices` is a script developed to implement the coincidence lattice method, as described in [1].

Authors: Daniel S. Koda (MSc student in 2016), Prof. Dr. Friedhelm Bechstedt, Prof. Dr. Marcelo Marques and Prof. Dr. Lara K. Teles are responsible for the theory behind this software. The software was developed by Daniel S. Koda.

Software: The raw code and the installation package can be found at `https://github.com/gmsn-ita/CoincidenceLattices`. New versions will be constantly updated in this link.

1.2 Installation

Requirements: `clattices` uses Python 3 and C as standard languages. The packages `numpy`, `re`, `argparse` and `setuptools` are required in order to take full advantage of `clattices`. Please use your package manager to install these packages before proceeding with the installation.

Installing: Directly download the `clattices` code from `https://github.com/gmsn-ita/CoincidenceLattices` or simply execute:
`git clone https://github.com/gmsn-ita/CoincidenceLattices`.

Once downloaded, go to the directory `clattices/` and execute:
`# python setup.py install` or `$ sudo python setup.py install`

If Python 3 is not the default Python language in your operating system (confirm by executing `python --version`), you may want to try:
`# python3 setup.py install`

Then, install the `clattices_loop` module inside the root folder: `# python setup.py install` or `$ sudo python setup.py install`

From this point on, you can execute `clattices` as a normal command-line interface software.

2 Running `clattices`

2.1 Flags

Introduction: `clattices` was created to be easily tunable with its flags. Therefore, most of its features are activated by running the script with flags starting with a dash `-` and a mnemonic. Flags and tunable parameters are briefly listed here and relates to their counterparts in [1]. The usage of these flags is demonstrated in the examples section.

`input_files` Called without dashes, all input files describing crystals must be listed for `clattices`. Full directories also can be passed as argument by typing `directory/*` instead of every single file.

`-h` or `--help` Lists a help text containing a summary of flags and their descriptions.

`-o` or `--output_file` Specifies the filename for the output table. Set to `CoincidenceLattices.dat` by default

`-a` or `--angles` Specifies the range [`ANGLE_MIN`, `ANGLE_MAX`] of angles to be investigated. Set to [0°, 30°] by default.

`-s` or `--angles_step` Specifies the step for sampling the range [`ANGLE_MIN`, `ANGLE_MAX`] of angles to be investigated. Set to 0.1° by default.

`-N` Integer cutoff for solving the system of diophantine equations. Relates to the maximum area of the possible supercell investigated by Area $\propto$ `N`^{2}. Strongly influences the execution time. Set to 7 by default.

`-t` or `--tolerance` Maximum strain applied to one material in order to make the 2D bilayer commensurate. Set to 0.02 by default.

`--angle_tolerance` Maximum angle distortion (shearing strain) that must be applied to one material in order to make the 2D bilayer commensurate. Set to 0.05 by default.

`-n` or `--n_atoms` Specifies the maximum number of atoms inside the supercells found. Set to 100 (atoms) by default.

`-l` or `--label_size` Specifies the number of characters of the first column of the output table. Useful for text-formatting when long/short labels are used. Set to 20 (chars) by default.

`-f` or `--self_combinations` Allows combinations of the given materials with themselves be investigated. Otherwise, only heterobilayers will be investigated. Set as False by default, but calling this flag sets the investigation as True.

`-r` or `--first` Allows combinations of the given materials only with the first material passed as input. Otherwise, all combinations two-to-two of the crystals will be investigated. Set as False by default, but calling this flag sets the investigation as True.

`-q` or `--quiet` Whether or not to print textual information on screen. Set as False (verbose mode) by default, but calling this flag suppresses all text.

2.2 Standards for 2D crystal inputs

Crystal files: Each 2D crystal must be described using a plain text file with the following standard:

```
Label
Number of atoms
Bravais lattice
Relevant parameters
```

Specific examples are shown in section 3.

Label: Each 2D crystal will be handled by `clattices` by its label. The label will identify the 2D crystal on the output table. Any one-line text is accepted, but the user may have to deal with column sizes by himself/herself if the label is too long (see tag `-l` in section 2.1).

Number of atoms: Number of atoms inside the unit cell of the crystal. Just an integer must be typed.

Bravais lattice: One of the four Bravais lattices for 2D crystals: `square`, `rectangular`, `hexagonal` and `oblique`. This field is not case-sensitive.

Relevant parameters: Necessary parameters to describe the lattice. For `square` and `hexagonal` lattices, only one parameter is needed, namely the lattice parameter of these structures. In the case of `rectangular` lattices, two lattice parameters must be specified. Finally, to describe `oblique` lattices, three parameters must be typed: two lattice parameters and the angle between the two vectors. All numbers are treated as floating-point numbers.

3 Tutorial

In this section, actual examples of crystals will be supplied in order to clarify the use of `clattices`. The relevance of the information is already discussed in [1]. The files for this example are available in the folder `examples/` in the `clattices` directory.

3.1 Matching of a single 2D crystal with itself

Input: For this example, graphene will be used as input. It is possible to investigate small coincidence lattices of graphene from 0 to 30 degrees simply by running:

```
clattices Graphene -a 0 30 -o ex1_output1.dat --self_combinations
```

This leads to four combinations with less than 100 atoms inside the supercell.

Larger supercells: If larger supercells are allowed, it suffices to increase the `N` factor:

```
clattices Graphene -a 0 30 -o ex1_output2.dat --self_combinations
-N 15 -n 300
```

Execution time: This time, the execution time is greatly increased. This occurs because the system of diophantine equations is solved by brute force and the complexity of the algorithm is $\mathcal{O}(\text{nAngles} \cdot N^4)$. The number of atoms influences only the printing section.

3.2 Coincidences of two 2D crystals

Input: As a second example, graphene and hBN will be used as input. Coincidences between the two hexagonal lattices are found by running:

```
clattices Graphene hBN -o ex2_output1.dat
```

This finds coincidences between Gr and hBN. The output is exactly that reported in [1].

Single angles: If one is interested in running exactly one angle (e.g. for verifying favorable combinations to compare theoretical simulations to experimental results), the following line can be executed:
`clattices Graphene hBN -o ex2_output2.dat -a 21.8 21.8`

3.3 Coincidences of 2D crystals with different Bravais lattices

Input: In [1], coincidences between materials with different Bravais lattices are reported. We can find two 2D crystals and their coincidences by running, for example:
`clattices Phosphorene HfSe2 -o ex3_output1.dat --angle_tolerance 0.1`

Effect of `angle_tolerance`: Because two different Bravais lattices are involved in this coincidence, it is necessary to either increase the tolerance to make the systems commensurate or to refine the range of angles investigated by changing `angles_steps`.

On the difference of angles: It is important to note that the results differ a little from [1] because the latter uses another reference to the 0 degrees angle of the hexagonal lattice, while the former aligns the first vector of the first vector of the first lattice with the first vector of the second lattice. This is better visualized with Fig. 1.

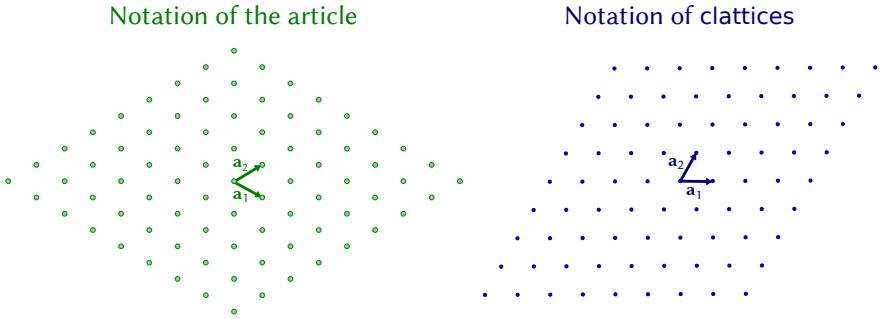

Figure 1: Differences between the notation for a 0° hexagonal lattice used in [1] (green, on the left) and in `clattices` (blue, on the right).

3.4 Coincidences between several different 2D crystals

Input: The fastest way to investigate coincidence between pairs of several different crystals is by creating several 2D crystals and putting them together in a directory, then executing `clattices` as in:
`clattices crystals/* -o ex4_output.dat`

3.5 Description of an oblique lattice

Input: An oblique lattice may be easily described with three parameters. In this example, graphene is described as an oblique lattice by the following three parameters:
`2.467 2.467 60`

in which the first two parameters are the lengths of the first and second lattice vectors, respectively, and the third parameter is the angle between the two vectors in degrees.

4 Conclusions

`clattices` is still evolving. If you feel like contributing with suggestions, coding or bug reports, please send an e-mail to the authors. Your help is appreciated.

Technical documentation is explicit in the `docs/` directory of the `clattices` package. If any doubt still persists, do not hesitate in contacting us!

Acknowledgements

We would like to thank F. L. Freitas, P. B. Arruda and I. Guilhon for tips and helps with the development of this software.

References

[1] D. S. Koda, F. Bechstedt, M. Marques, and L. Kühl Teles, "Coincidence Lattices of 2D Crystals: Heterostructure Predictions and Applications," *The Journal of Physical Chemistry C*, vol. 120, pp. 10895–10908, 2016.