

Number of equidistant neighbors on honeycomb lattice.

Supplementary scripts and data.

Zbigniew Koziol*

National Center for Nuclear Research, Materials Research Laboratory,
ul. Andrzeja Sołtana 7, 05-400 Otwock-Świerk, Poland

Supplementary scripts and data.

The scripts and data are provided as a basic illustration of concepts described in the original article [1],[2]. These examples can be used for *real* computation or may work as a starting point for developing more elaborated computational algorithms.

Scripts are written in Perl. Some basic familiarity with that language is needed for proper understanding of what is done there. If you need help with this - do not hesitate to contact the author.

Additional explanations are available within the source code of scripts. Figure 1 summarizes the transformations used for finding all possible pairs of indexes of integer numbers (m, n) preserving constant value of quantity defined as $\rho^2 = m^2 + mn + n^2$, described in details in [1],[2].

get_radiuses.pl

The script *get_radiuses.pl* finds radius for all possible pairs of natural numbers (m, n) , where $m > 0$, $n \geq 0$, $m \geq n$, sorts results by radius. Includes all pairs (m, n) that have the same radius, $\rho^2 = m^2 + mn + n^2$, in one record. It outputs results in a form of records (collection of lines) separate for every pair (m, n) .

get_radiuses.pl outputs data in the format stored in data file *MN_pairs.dat*, therefore this description of its output is also the description of the data format in *MN_pairs.dat*.

*e-mail: softquake@gmail.com

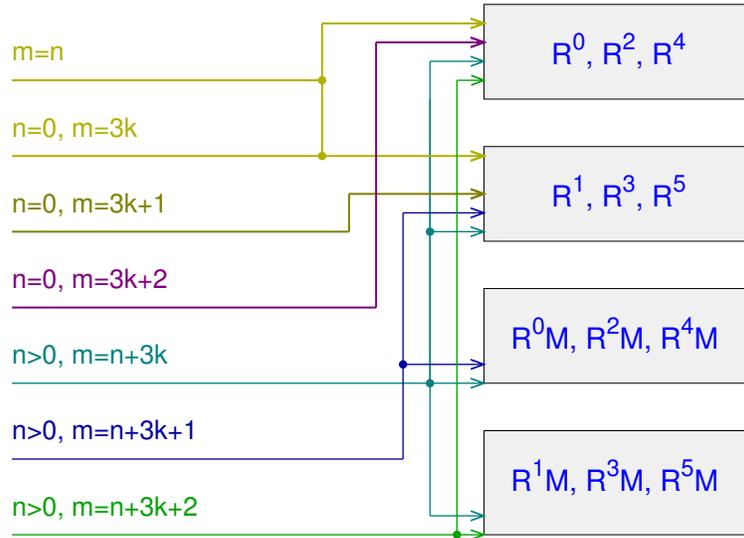


Figure 1: Transformations used for finding all possible pairs of indexes of integer numbers (m, n) preserving constant value of $\rho^2 = m^2 + mn + n^2$, as described in details in [1],[2], in table 4.

Here is an example output for $\rho^2 = 1$ (displayed at line beginning as *000001*, where *000001* is the squared radius of a pair, prefixed by certain number of *0*'s):

000001	1	0		
	-1	0	1	1
	-1	1	0	1
	0	-1	0	1
	0	1	1	1
	1	-1	1	1
	1	0	0	1

The first line contains (TAB separated) values of ρ^2 , m and n . The next 6 lines, starting with a TAB, contain values of m' and n' after a transformation (there is no information provided on which transformation is related to that given pair of indexes, but one may change that within the script). Next two

columns are results of checking for if point (m, n) belongs to graphene lattice for type I ordering (value is 0 or 1). The last column is if point $(m + 1, n)$ belongs to graphene lattice for type II ordering. Both these checkings are done by subroutine *isGraphen(m,n)* in script code.

When there is more than one pair of (m, n) producing the same ρ^2 , the first output line contains all these pairs listed, TAB separated. Here is an example output for $\rho^2 = 147$, when there are two valid pairs, $(7, 7)$ and $(11, 2)$. After the transformations are listed for $(7, 7)$, an empty line follows and the results for $(11, 2)$ are printed:

```
000147  7      7      11      2
        -14     7       1       0
        -7     -7       1       0
        -7     14       1       0
        14     -7       1       0
         7    -14       1       0
         7      7       1       0

        -11    -2       1       0
        -11    13       1       0
        -13    11       1       0
        -13     2       1       0
         -2   -11       1       0
         -2    13       1       0
         11   -13       1       0
         11     2       1       0
         13   -11       1       0
         13    -2       1       0
          2   -13       1       0
          2    11       1       0
```

multi_pairs.pl

The output of the script *multi_pairs.pl* is self-explanatory. It finds all possible pairs of (m', n') that have the same squared radius ρ^2 as the radius given by initial values of (m, n) (editable in the code of the script), $\rho^2 = m^2 + mn + n^2$. The script is based on considerations illustrated in figure 3 and summarized in table 4 of references [1],[2].

The lines of searched values of (m', n') are marked by *!!!* at ends of records.

References

- [1] Z Koziol, Number of equidistant neighbors on honeycomb lattice, 2020 arXiv:2004.11840 [cond-mat.mes-hall].
- [2] Z Koziol, Number of equidistant neighbors on honeycomb lattice 2020 *J. Phys. Comm.*