1 Introduction

hop is a key task of PiLab. It generates all tight-binding parameters of the system. This task will first assign the orbitals[1] of each sublattice. Then **hop** will automatically generates the hopping integrals using Slaster-Koster methods. If spin-orbit coupling is assumed, **hop** will also generate onsite spin-orbit coupling matrix. When all the interaction matrices are generated, **hop** can also transform their basis set. By default, it is in cubic harmonics. If needed, one can transform them to spherical harmonics, relativistic spherical harmonics and relativistic cubic harmonics[2]. Using correct basis set will help users to input the onsite energy based on their crystal field splitting easily.

In most tight-binding calculation, not all the suborbitals are needed. For example, people usually consider p_z only rather than the full p-orbital when studying graphene systems. In **hop**, there is also a keyword for users to select the suborbitals they want.

2 Dictionary

2.1 Input

hop.SiteOrb This parameter assigns the orbital on each sublattice. Each orbital is specified by two values [sublattice index, orbital quantum number]. Sublattice index is defined by the order of the sublattice in lat.Sublatt of the **lat** task. orbital quantum number are 0 for s, 1 for p, 2 for d and 3 for f orbitals.For example, [1,1;1,1;2,2] means there are two p orbitals on sublattice 1 and one d orbital on sublattice 2.

hop.Order This parameter sets the order of the hopping integrals. If hop.Order=1, **hop** will only calculates the hopping between first nearest sites. Similar, if hop.Order=2, it will calculate the hopping integrals between first and second nearest site. Obviously, this parameter has to be smaller or equal to lat.Order, i.e. **lat.Order** $\leq$ **hop.Order**, due to the lack of higher order lattice information. The benefits of using this parameter is for user to quickly compare the difference between 1st and 2nd hopping effects. Even though 2nd order hopping in input, **hop** will still set all 2nd order hopping integrals zero if this parameter is set to 1.

hop.SKint This parameter sets the hopping integrals. When two orbitals overlap to each other, their hopping integrals can be specified by just a few parameters, i.e the strength of the bonds with different symmetry. For example, when two p-orbitals couple to each other, σ-bond and π-bond will be formed. If these two bond strengths are given, the hopping integrals between arbitrary suborbitals as a function of bonding angles can be evaluated using Slaster-Koster formulation[3]. There are four Slaster-Koster integrals or, in other words, four different bonds which are called σ, π, δ and ϕ bond respectively. However, not all of them are needed to specify the coupling between two orbitals. Consider that each bond has the same symmetry as the $s(\sigma)$, $p(\pi)$, $d(\delta)$ and $f(\phi)$ orbital respectively. When two orbital couple to each other, only the minimal symmetry will be held. For example, $s-s$ has σ bond only, $p-p$ has both σ and π. However, $s-p$ will have σ only. Similarly, $d-d$ has σ, π, δ bonds, but $p-d$ has only σ, π. In this parameter, we will need 7 input values to specify the bond strength between two orbitals [orbital label 1, orbital label 2, hopping order, σ, π, δ, ϕ]. Orbital labels are the orbitals defined in hop.SiteOrb. 1 corresponds to first orbital defined in hop.SiteOrb and so on. Hopping order is the order of the two orbitals. σ, π, δ and ϕ are the bond strength. Even though not all bonds exist, you should still give the non-existed bonds values, e.g. zero. This is just to make the size of the input parameters consistent. No matter what are the values for the non-existed bonds, PiLab won't use them. For example, [1,2,1,2,1,0,0] means the orbital 1 and orbital 2 defined in hop.SiteOrb have σ bonding strength = 2 and π-bond strength = 1 if they are on first nearest sites. Of course, it is possible that two orbitals overlapped with second nearest sites. If so, set hopping order = 2. If you need to input more hopping integrals, separates each set via semicolon ";". An important restriction of this parameter is that **orbital label 1 must smaller or equal to orbital label 2** and PiLab will automatically generate the swapped case. For example, [2,1,1,2,1,0,0] is not allowed because once [1,2,1,2,1,0,0] is input, PiLab will generate [2,1,1,2,1,0,0] automatically and the user shoud not input it again.

[1]Here we use the term "orbital" to represent s, p, d, f and "suborbital" to represent a particular component of them. For example, p_x, p_y and p_z are suborbitals of the p orbital

[2]For the definition of each basis in PiLab, one should reference the "basis_set.pdf" file for details.

[3]see, Phys. Rev. 94 1498, J. Phys. C. 13 583 for details. Also see Slaster_Koster.pdf for the exact formulas used in PiLab.

hop.LS This parameter sets the strength of spin-orbit coupling. The Hamiltonian reads $H_{SO} = \alpha \mathbf{L} \cdot \mathbf{S}$. This parameter gives α.

hop.Filiter This parameters sets the value to filter out small hopping integrals. By default, this value is set to be 10^{-3}. Therefore, any hopping integrals whose absolute values are smaller than it will be set to zero. It helps user to make the hopping matrix more readable.

hop.Basis This parameter sets the orbital basis of the Hamiltonian. 'c' for cubic harmonics, 's' for spherical harmonics, 'rs' for relativistic spherical harmonics and 'rc' for relativistic cubic harmonics. Using appropriate basis will help you input the onsite energy based on crystal field splitting easier.

hop.SelState This parameter select the suborbitals. If one wants to select just a few suborbitals from the calculations, one should leave it empty and run **hop** task first. Once it is done, the output variable **hop.state_info_text** will show you all the suborbitals and their state labels. Check what suborbitals you want to keep and input their state labels in this parameter and the onsite energy of these states in **hop.OnsiteE**. Then run it again. PiLab will automatically kick out all non-selected suborbitals and regenerates the **hop** file again.

hop.OnsiteE This parameter sets the onsite energy of each suborbital. By default, PiLab will set all onsite energy as zero. Once hop.SelState is set, hop.OnsiteE will give the onsite energies of each selected state. For example, if we've picked state [1,3,5], then hop.OnsiteE=[-1,-2,-3] means their onsite energies are -1, -2, -3 respectively. If you leave it empty, then all the onsite energies of the selected states are assumed to be zero. Note that, even if you don't want kick out any suborbitals and just want to input onsite energies, you will still need to select all states. hop.OnsiteE will not be functional unless hop.SelState is set. **Here "..." is the keyword for line change in Scilab**. When this keyword occurs, Scilab will consider the current line and its next line as the same line. It is useful to deal with long input case.

2.2 Output

hop.state_info_text This variable shows all the quantum mechanical states (suborbitals) in the system. There are 5 strings in each row column [state_label, site, orbital identifier, orbital quantum number, text of suborbital]. State_label labels the number of each state. This number is also used to expand all the Hamiltonian matrices. In all later tasks, these numbers will also be used as the label of states. Site is the sublattice which this state belongs to. orbital identifiers are the order of each orbital input in hop.SiteOrb. So, two orbitals with the same orbital quantum numbers on the same site will still have different identifiers to distinguish them. Orbital quantum numbers are the same with the definition in hop.SiteOrb. Text of suborbital is a text description of the state. It contains three information: the suborbital number, the orbital quantum number in text form (s, p, d f) and the suborbital in text form. suborbital number is the number that PiLab used in the code to represent a suborbital. Since s, p, d f have 32 suborbitals in total (including spin), PiLab uses a number to represent each suborbital. This information is not that important if you are not going to develop PiLab project. Therefore, [1 # 1 # 1 # 2 # 9 D xy,d #] means state 1 is located in sublattice 1. Its orbital identifier is 1 and it is the spin-down xy component of a d orbital.[4]

hop.state_info This variable is the numerical format of hop.state_info. It is what PiLab really uses to deal with suborbitals in the code. For most users you can just ignored this part.

hop.LS_mat This variable is the LS coupling matrix. Since H_{SO} is a harmitian matrix, PiLab will output only the upper triangle part by sparse format.

hop.onsite_E This variable is the onsite energy matrix. Since H_{onsite} is a diagonal matrix, PiLab will output it by sparse format.

hop.hop_size This variable is the size of variable hop.hop_mat. It is just to help PiLab to preset the memory size if hop.hop_mat is loaded by other tasks. For most users, this variable can be ignored.

[4]For the definition and notation of each suborbital, see suborbital.pdf for details

hop.hop_mat(n)(:,:,m) This variable is the the hopping integrals between two orbitals. n represents the sublattice label and m represents the m-th neighbor site shown in lat.surr_site(n). Therefore, hop.hop_mat(1)(:,:,2) is the hopping between sublattice 1 and its 2nd neighbor, i.e the site defined by the 3rd row vector (1st row vector is the sublattice itself so it is not considered as a neighbor) of the of the variable lat.surr_site(1). Obviously, their hopping matrix is not hermitian because it just records the hopping from sublattice n to its m-th neighbor site. However if you combine the hopping matrix from m-th neighbor site to sublattice n, it will be hermitian. Since the hopping matrix is zero for most matrix elements, PiLab shows it in sparse format. Note that, this is the sparse matrix of the full hopping matrix rather than the upper triangle part only.

```
hop.SiteOrb=[1,2;2,1]         // specify orbital of each site, nx2, [site, l]
hop.Order=[1]                 // order of nearest order coupling, 1x1 integer, must <= lat.Order
hop.SKint=[1,2,1,2,1,0,0]         // specify SK parameters, nx7, [Orb1,Orb2,nn_order,ts,tp,td,tf]
hop.LS=[0]                    // strength of LS coupling, 1x1, real
hop.Filiter=[10^-3]               // fliter of small hopping elements, 1x1, real
hop.Basis=['c']               // Basis of the hopping matrix, 'c', 's', 'rc', 'rs'
hop.SelState=[1:16]               // Input state labels to pick states, 1xn, integer
hop.OnsiteE=...               // Onsite energy of picked states, 1xn, real
[-3,-3,-3,0,0,-3,-3,-3,0,0,-3,-3,-3,-3,-3,-3]

============= PiLib Variable =============
hop.state_info_text, @full, [state_label, site, identifier, l, SubOrb_text]
ORDER=    0, SIZE=[   16,    5], TYPE=STRING

1 # 1 # 1 # 2 # 9 D  xy,d #
2 # 1 # 1 # 2 # 10 D  yz,d #
3 # 1 # 1 # 2 # 11 D  zx,d #
4 # 1 # 1 # 2 # 12 D  x2-y2,d #
5 # 1 # 1 # 2 # 13 D  3z2-r2,d #
6 # 1 # 1 # 2 # 14 D  xy,u #
7 # 1 # 1 # 2 # 15 D  yz,u #
8 # 1 # 1 # 2 # 16 D  zx,u #
9 # 1 # 1 # 2 # 17 D  x2-y2,u #
10 # 1 # 1 # 2 # 18 D  3z2-r2,u #
11 # 2 # 2 # 1 # 3 P  x,d #
12 # 2 # 2 # 1 # 4 P  y,d #
13 # 2 # 2 # 1 # 5 P  z,d #
14 # 2 # 2 # 1 # 6 P  x,u #
15 # 2 # 2 # 1 # 7 P  y,u #
16 # 2 # 2 # 1 # 8 P  z,u #

============= PiLib Variable =============
hop.state_info, @full, [state_label, site, identifier, l, SubOrb]
ORDER=    0, SIZE=[   16,    5], TYPE=INTEGER

     1     2     3     4     5

     1     1     1     2     9
     2     1     1     2    10
     3     1     1     2    11
     4     1     1     2    12
     5     1     1     2    13
     6     1     1     2    14
     7     1     1     2    15
     8     1     1     2    16
     9     1     1     2    17
    10     1     1     2    18
    11     2     1     1     3
    12     2     1     1     4
    13     2     1     1     5
    14     2     1     1     6
    15     2     1     1     7
    16     2     1     1     8

============= PiLib Variable =============
hop.LS_mat, @t-sp, LS coupling matrix
ORDER=    1, SIZE=[    1,    3], TYPE=SPARSE
```

Figure 1: page 1 of NiO_hop.plb

```
      1      2                  3

     16     16    0.000000  0.000000

============== PiLib Variable ==============
hop.onsite_E, @t-sp, onsite energy matrix
ORDER=    1, SIZE=[   13,    3], TYPE=SPARSE

      1      2                  3

     16     16    0.000000  0.000000
      1      1   -0.300000  0.000000
      2      2   -0.300000  0.000000
      3      3   -0.300000  0.000000
      6      6   -0.300000  0.000000
      7      7   -0.300000  0.000000
      8      8   -0.300000  0.000000
     11     11   -0.300000  0.000000
     12     12   -0.300000  0.000000
     13     13   -0.300000  0.000000
     14     14   -0.300000  0.000000
     15     15   -0.300000  0.000000
     16     16   -0.300000  0.000000

============== PiLib Variable ==============
hop.hop_size, @full, size of hop.hop_mat, [sublatt, size(hop.hop_mat(n))]
ORDER=    0, SIZE=[    2,    4], TYPE=INTEGER

      1      2      3      4

      1     16     16      6
      2     16     16      6

============== PiLib Variable ==============
hop.hop_mat(1)(:,:,1), @a-sp, hop_mat between site-1 and its 1-th neighbor
ORDER=    1, SIZE=[    7,    3], TYPE=SPARSE

      1      2                  3

     16     16    0.000000  0.000000
      3     13    0.100000  0.000000
      4     11    0.173205  0.000000
      5     11   -0.100000  0.000000
      8     16    0.100000  0.000000
      9     14    0.173205  0.000000
     10     14   -0.100000  0.000000

============== PiLib Variable ==============
hop.hop_mat(1)(:,:,2), @a-sp, hop_mat between site-1 and its 2-th neighbor
ORDER=    1, SIZE=[    9,    3], TYPE=SPARSE

      1      2                  3

     16     16    0.000000  0.000000
      1     11    0.100000  0.000000
      2     13    0.100000  0.000000
```

Figure 2: page 2 of NiO_hop.plb

```
     4     12   -0.173205  0.000000
     5     12   -0.100000  0.000000
     6     14    0.100000  0.000000
     7     16    0.100000  0.000000
     9     15   -0.173205  0.000000
    10     15   -0.100000  0.000000

============== PiLib Variable ==============
hop.hop_mat(1)(:,:,3), @a-sp, hop_mat between site-1 and its 3-th neighbor
ORDER=    1, SIZE=[    7,    3], TYPE=SPARSE

     1      2                  3

    16     16    0.000000  0.000000
     2     12   -0.100000  0.000000
     3     11   -0.100000  0.000000
     5     13   -0.200000  0.000000
     7     15   -0.100000  0.000000
     8     14   -0.100000  0.000000
    10     16   -0.200000  0.000000

============== PiLib Variable ==============
hop.hop_mat(1)(:,:,4), @a-sp, hop_mat between site-1 and its 4-th neighbor
ORDER=    1, SIZE=[    7,    3], TYPE=SPARSE

     1      2                  3

    16     16    0.000000  0.000000
     2     12    0.100000  0.000000
     3     11    0.100000  0.000000
     5     13    0.200000  0.000000
     7     15    0.100000  0.000000
     8     14    0.100000  0.000000
    10     16    0.200000  0.000000

============== PiLib Variable ==============
hop.hop_mat(1)(:,:,5), @a-sp, hop_mat between site-1 and its 5-th neighbor
ORDER=    1, SIZE=[    9,    3], TYPE=SPARSE

     1      2                  3

    16     16    0.000000  0.000000
     1     11   -0.100000  0.000000
     2     13   -0.100000  0.000000
     4     12    0.173205  0.000000
     5     12    0.100000  0.000000
     6     14   -0.100000  0.000000
     7     16   -0.100000  0.000000
     9     15    0.173205  0.000000
    10     15    0.100000  0.000000

============== PiLib Variable ==============
hop.hop_mat(1)(:,:,6), @a-sp, hop_mat between site-1 and its 6-th neighbor
ORDER=    1, SIZE=[    7,    3], TYPE=SPARSE

     1      2                  3
```

Figure 3: page 3 of NiO_hop.plb

```
    16     16    0.000000  0.000000
     3     13   -0.100000  0.000000
     4     11   -0.173205  0.000000
     5     11    0.100000  0.000000
     8     16   -0.100000  0.000000
     9     14   -0.173205  0.000000
    10     14    0.100000  0.000000

============== PiLib Variable ==============
hop.hop_mat(2)(:,:,1), @a-sp, hop_mat between site-2 and its 1-th neighbor
ORDER=    1, SIZE=[    7,    3], TYPE=SPARSE

     1      2                   3

    16     16    0.000000  0.000000
    11      4   -0.173205  0.000000
    11      5    0.100000  0.000000
    13      3   -0.100000  0.000000
    14      9   -0.173205  0.000000
    14     10    0.100000  0.000000
    16      8   -0.100000  0.000000

============== PiLib Variable ==============
hop.hop_mat(2)(:,:,2), @a-sp, hop_mat between site-2 and its 2-th neighbor
ORDER=    1, SIZE=[    9,    3], TYPE=SPARSE

     1      2                   3

    16     16    0.000000  0.000000
    11      1   -0.100000  0.000000
    12      4    0.173205  0.000000
    12      5    0.100000  0.000000
    13      2   -0.100000  0.000000
    14      6   -0.100000  0.000000
    15      9    0.173205  0.000000
    15     10    0.100000  0.000000
    16      7   -0.100000  0.000000

============== PiLib Variable ==============
hop.hop_mat(2)(:,:,3), @a-sp, hop_mat between site-2 and its 3-th neighbor
ORDER=    1, SIZE=[    7,    3], TYPE=SPARSE

     1      2                   3

    16     16    0.000000  0.000000
    11      3    0.100000  0.000000
    12      2    0.100000  0.000000
    13      5    0.200000  0.000000
    14      8    0.100000  0.000000
    15      7    0.100000  0.000000
    16     10    0.200000  0.000000

============== PiLib Variable ==============
hop.hop_mat(2)(:,:,4), @a-sp, hop_mat between site-2 and its 4-th neighbor
ORDER=    1, SIZE=[    7,    3], TYPE=SPARSE

     1      2                   3
```

Figure 4: page 4 of NiO_hop.plb

```
    16      16    0.000000  0.000000
    11       3   -0.100000  0.000000
    12       2   -0.100000  0.000000
    13       5   -0.200000  0.000000
    14       8   -0.100000  0.000000
    15       7   -0.100000  0.000000
    16      10   -0.200000  0.000000

============= PiLib Variable =============
hop.hop_mat(2)(:,:,5), @a-sp, hop_mat between site-2 and its 5-th neighbor
ORDER=    1, SIZE=[    9,    3], TYPE=SPARSE

     1       2                   3

    16      16    0.000000  0.000000
    11       1    0.100000  0.000000
    12       4   -0.173205  0.000000
    12       5   -0.100000  0.000000
    13       2    0.100000  0.000000
    14       6    0.100000  0.000000
    15       9   -0.173205  0.000000
    15      10   -0.100000  0.000000
    16       7    0.100000  0.000000

============= PiLib Variable =============
hop.hop_mat(2)(:,:,6), @a-sp, hop_mat between site-2 and its 6-th neighbor
ORDER=    1, SIZE=[    7,    3], TYPE=SPARSE

     1       2                   3

    16      16    0.000000  0.000000
    11       4    0.173205  0.000000
    11       5   -0.100000  0.000000
    13       3    0.100000  0.000000
    14       9    0.173205  0.000000
    14      10   -0.100000  0.000000
    16       8    0.100000  0.000000
```

Figure 5: page 5 of NiO_hop.plb