MagneticTB manual

Zeying Zhang

February 14, 2023

Contents

1 Features

MagneticTB is an open-source software package developed based on the co-representation theory of magnetic groups. It can generate tight-binding models that satisfy any of the 1651 magnetic group symmetries. The specific features are as follows:

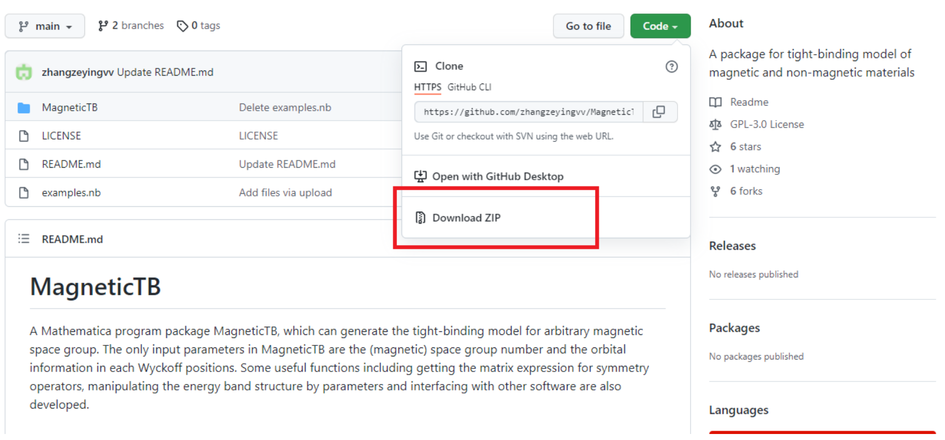
- The corresponding tight-binding model can be constructed simply by inputting the magnetic group number and the information of each Wyckoff position.
- Obtain the symmetry operations of the 1651 magnetic groups.
- Interface with other software.
- Plot the energy spectrum, and the spectrum can be dynamically displayed with changes in the parameters.
- Construct tight-binding models of non-magnetic materials.
- Consider non-collinear magnetic structures.
- Construct tight-binding models of single groups (without spin) and magnetic double groups (with spin).

2 Installation

First, log in to the homepage of the MagneticTB package on GitHub (no account or password required):

https://github.com/zhangzeyingvv/MagneticTB

Click ”Code”, ”Download ZIP”, and download the MagneticTB-main.zip file, as shown in the following figure:



zhangzeyingvv Update README.md

MagneticTB	Delete examples.nb
LICENSE	LICENSE
README.md	Update README.md
examples.nb	Add files via upload

https://github.com/zhangzeyingvv/MagneticT

Use Git or checkout with SVN using the web URL.

A package for tight-binding model of magnetic and non-magnetic materials

No releases published

No packages published

MagneticTB

A Mathematica program package MagneticTB, which can generate the tight-binding model for arbitrary magnetic space group. The only input parameters in MagneticTB are the (magnetic) space group number and the orbital information in each Wyckoff positions. Some useful functions including getting the matrix expression for symmetry operators, manipulating the energy band structure by parameters and interfacing with other software are also developed.

Open Mathematica and run:

```
$UserBaseDirectory
```

Click Shift+Enter to run. Open the displayed directory and go to the ”Applications” folder.

Unzip the previously downloaded MagneticTB-main.zip file to get a MagneticTB folder and three files. Copy the unzipped MagneticTB folder to the ”Applications” folder to complete the installation. Run in Mathematica:

```
Needs["MagneticTB‘"]
```

The package can be loaded by running the above command. Experienced users can install it anywhere in the `$Path`, but some directories may require administrator privileges. The above method is applicable to both Windows and Linux users. For more information, please refer to Comput. Phys. Commun., 270, 108153 (2022).

3 Capabilities of MagneticTB

3.1 Core module

The core functionality of MagneticTB is to construct symmetry-adapted tight-binding models, consisting of three main functions: `msgop`, `init`, and `symham`.

`msgop` can provide the symmetry operations for any of the 1651 magnetic groups, for example:

```
1 msgop[gray[191]]
2 msgop[bnsdict[{191, 236}]]
3 msgop[ogdict[{191, 8, 1470}]]
```

In the first line, `gray[191]` obtains the code for the gray group with number 191 (which is the space group with time-reversal symmetry); In the second line, `bnsdict [{191, 236}]` obtains the code for the magnetic group with BNS number 191.236; In the third line, `ogdict[{191, 8, 1470}]` obtains the code for the magnetic group with OG number 191.8.1470.

Using `msgop[code]` gives the symmetry operations for the corresponding magnetic group. The above three commands are actually equivalent, and their output is:

```
 1 Magnetic space group (BNS): {191.234,P6/mmm1'}
 2 Lattice : HexagonalP
 3 Primitive Lattice Vactor: {{a,0,0},{−(a/2),(Sqrt[3] a)/2,0},{0,0,c}}
 4 Conventional Lattice Vactor: {{a,0,0},{−(a/2),(Sqrt[3] a)/2,0},{0,0,c}}
 5
 6  {{"1",{{1,0,0},{0,1,0},{0,0,1}},{0,0,0}, F},
 7 {"6z ",{{1,−1,0},{1,0,0},{0,0,1}},{0,0,0}, F},
 8 {"3z ",{{0,−1,0},{1,−1,0},{0,0,1}},{0,0,0}, F},
 9 ...
10  {"1",{{1,0,0},{0,1,0},{0,0,1}},{0,0,0}, T},
11 {"6z ",{{1,−1,0},{1,0,0},{0,0,1}},{0,0,0}, T},
12 {"3z ",{{0,−1,0},{1,−1,0},{0,0,1}},{0,0,0}, T},
13 ...}
```

Lines 1-4 show the basic information of the magnetic group, including its symbol, lattice type, and lattice vectors. Lines 6-13 show the symmetry operations for the gray group with number 191, where the first four columns correspond to the name of the symmetry operation, its rotation part, its translation part, and whether it includes time-reversal operation.

With the symmetry operations, we can use the init function to initialize the structure:

```
sgop=msgop[gray[191]];
init [
  lattice  -> {{a, 0, 0}, {-(a/2), (Sqrt[3]  a)/2,  0},  {0,  0,  c}},
  lattpar  -> {a -> 1, c -> 3},
  wyckoffposition  -> {{{1/3, 2/3, 0}, {0, 0,  0}}},
 symminformation -> sgop,
  basisFunctions  -> {{"pz"}}];
```

The init function has five options that must be set. The lattice option corresponds to lattice vectors and the symminformation option corresponds to the set of symmetry operations, which can directly use the output of msgop as the option value or use user-defined symmetry operations. The lattpar option corresponds to the numerical values of the lattice vectors parameters, and the wyckoffposition option corresponds to the Wyckoff positions to be considered. Its format is:

$$\{\{\boldsymbol{a}_1, \boldsymbol{m}_1\}, \{\boldsymbol{a}_2, \boldsymbol{m}_2\}, ...\}$$

The $\boldsymbol{ai}$ and $\boldsymbol{mi}$ in the above text refer to the coordinates and magnetization directions of the ith Wyckoff position. For equivalent Wyckoff positions, only one of them needs to be input.

The basisFunctions option specifies the basis functions to be considered for each Wyckoff position. The format is:

$$\{b_1, b_2, ...\}$$

b_i is the list of basis functions to be considered for the ith Wyckoff point. The correspondence between basis functions and strings is shown in the following table:

basis function	code	basis function	code
s	"s"	p_x	"px"
p_y	"py"	p_z	"pz"
$p_x + ip_y$	"px+ipy"	$p_x - ip_y$	"px−ipy"
d_{z^2}	"dz2"	d_{xy}	"dxy"
d_{yz}	"dyz"	d_{xz}	"dxz"
$d_{x^2-y^2}$	"dx2−y2"		

To include spin, simply add "up" or "dn" after the basis function name, for example, {"pzup","dxydn"}. If a basis function is not listed in the above table, one can input its analytic expression directly. For example, if one wants to include the f_{xyz} orbital, they can input basisFunctions -> {{x*y*z}}.

To check if the generated structure is correct after running the initialization function, you can use the atompos function.

```
In:= atompos
Out:= {{{{1/3, 2/3, 0}, {0,  0,  0}}, {{2/3, 1/3,  0},  {0,  0,  0}}}}
```

It indicates that there are two atomic positions to be considered in the primitive cell of the structure, which are $(1/3, 2/3, 0)$ and $(2/3, 1/3, 0)$. The magnetic moment of both atoms is {0, 0, 0}, indicating that they are nonmagnetic. After

code	properties
atompos	The atomic coordinates and magnetic directions for each atom
wcc	Atom position of each orbital
reclatt	Reciprocal lattice vector
symmetryops	Representation matrix for symmetry operators
unsymham[n]	Generating the Hamiltonian without any symmetry constraints
symham[n]	Generating the Hamiltonian with symmetry constraints
symmcompile	Information about symmetry operations and representation matrices
bondclassify	Provides bond lengths for all neighbors
showbonds[n]	Show bond lengths for $(n-1)$th nearest neighbors

running init, the basic properties that can be obtained are shown in the following table:

After verifying the accuracy of the information, symham[n] can be used to generate a symmetrized tight-binding Hamiltonian considering the $(n-1)$th nearest neighbors. For example,

```
symham[1]
```

will generate the Hamiltonian with only on-site energies, while

```
Sum[symham[n], {n, 1, 3}];
```

will generate the Hamiltonian considering on-site energies, nearest and next-nearest neighbor interactions.

3.2 Plotting module

There are two functions in the plotting module, bandplot and bandManipulate. Usage is as follows:

```
bandplot[band path, number of points in each segment, symham−generated Hamiltonian,
    parameter values]
bandManipulate[band path, number of points in each segment, symham−generated Hamiltonian
    ]
```

bandplot can generate a beautiful band plot, which can be directly used as a figure in a paper. bandManipulate can generate a dynamic band plot that changes with the parameter values. See example in 4.1.

3.3 Interface module

The interface module can convert the Hamiltonian generated by symham into the wannier90_hr.dat file format. Usage is as follows:

```
hop[symham−generated Hamiltonian, parameter values]
```

See example in 4.1. Note that the Hamiltonian generated by symham should be used as input. **Do not use the Hamiltonian resolved by yourself as input. If you want certain parameters to be zero, specify them as zero in the parameter values.**

4 Example

4.1 Graphene

To generate the tight-binding model of graphene near the Fermi surface, one only needs to know the $2c$ Wyckoff point of two carbon atoms in the 191st space group and that the electronic states near the Fermi surface are composed of carbon atoms' p_z orbitals. The specific code is as follows:

```
1   Needs["MagneticTB`"];
2   sgop = msgop[gray[191]];
3    init [
4    lattice  -> {{a, 0, 0}, {-(a/2), (Sqrt[3]  a)/2,  0},  {0,  0,  c}},
5    lattpar  -> {a -> 1, c -> 3},
6    wyckoffposition  -> {{{1/3, 2/3, 0}, {0, 0,  0}}},
7   symminformation -> sgop,
8    basisFunctions  -> {{"pz"}}];
9   ham = Sum[symham[i], {i, 3}];
10  MatrixForm[ham]
```

The first line loads the MagneticTB package. The second line gets the symmetry operations of the 191st gray space group. The third line is the initialization function. The fourth to eighth lines are the parameters of the initialization function. The fourth line is the lattice vector. The fifth line is the value of the parameters in the lattice constant. The sixth line is the Wyckoff point coordinates and magnetic direction to be considered. The seventh line is the symmetry satisfied by the symmetrized tight-binding model. The eighth line is the atomic orbitals to be considered. The ninth line is the output Hamiltonian including the next-nearest neighbor.

To consider spin-orbit coupling, simply change the 8th line of the code from:

```
 basisFunctions  -> {{"pz"}}];
```

to:

```
 basisFunctions  -> {{"pzup","pzdn"}}];
```

Then,

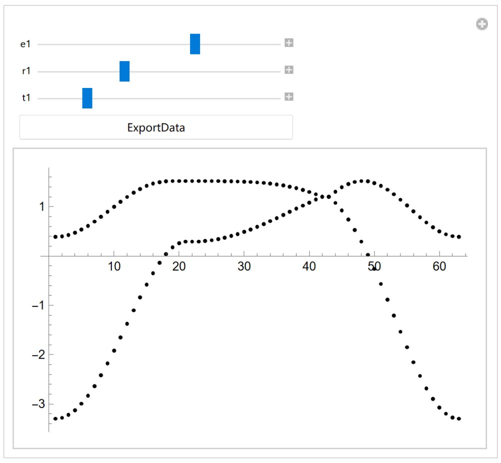
```
sgop = msgop[gray[191]];
 init [ lattice  -> {{a, 0, 0}, {-(a/2), (Sqrt[3]  a)/2,  0},  {0,  0,  c}},
 lattpar  -> {a -> 1, c -> 3},
 wyckoffposition  -> {{{1/3, 2/3, 0}, {0, 0,  0}}},
symminformation -> sgop,
 basisFunctions  -> {{"pzup", "pzdn"}}];
hamsoc = Sum[symham[i], {i, 3}];
MatrixForm[hamsoc]
```

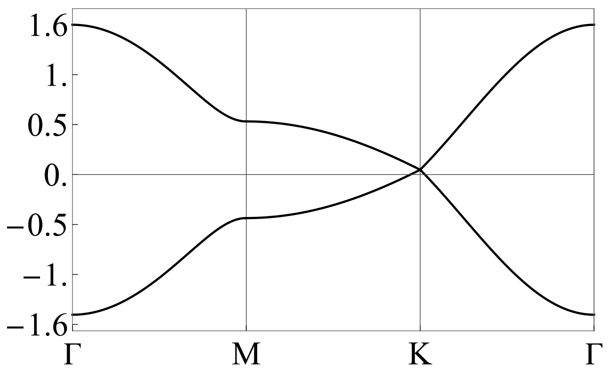
will generate the Hamiltonian that includes spin-orbit coupling. Once the Hamiltonian is obtained, band structures can be plotted and wannier90_hr.dat can be generated. The specific code for plotting the band structure is as follows:

```
path={
   {{{0,0,0},{0,1/2,0}},{"G","M"}},
   {{{0,1/2,0},{1/3,1/3,0}},{"M","K"}},
   {{{1/3,1/3,0},{0,0,0}},{"K","G"}}
};
```

```
6   bandManipulate[path, 20, ham]
7   bandplot[path, 200, ham, {e1 −> 0.05, t1 −> 0.5, r1−> 0}]
```

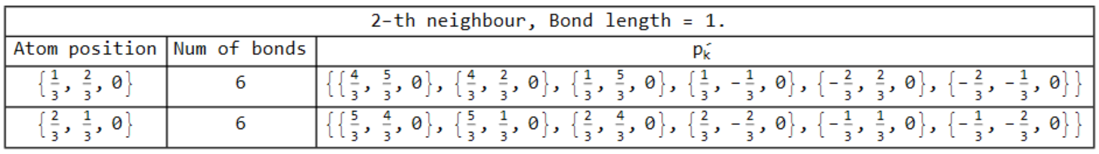
The first five lines define the $\boldsymbol{k}$-point path, the sixth line dynamically displays the band structure, and the seventh line plots the band structure for the given parameters. The results are shown in the following figure:





Clicking on ”ExportData” displays the size of the parameters.

The command showbonds can be used to display the bond lengths between neighboring atoms in graphene. For example, to display the bond lengths for the next-nearest neighbors, use showbonds[3]. The output is shown below:



2-th neighbour, Bond length = 1.		
Atom position	Num of bonds	$p_{\acute{k}}$
$\{\frac{1}{3}, \frac{2}{3}, 0\}$	6	$\{\{\frac{4}{3}, \frac{5}{3}, 0\}, \{\frac{4}{3}, \frac{2}{3}, 0\}, \{\frac{1}{3}, \frac{5}{3}, 0\}, \{\frac{1}{3}, -\frac{1}{3}, 0\}, \{-\frac{2}{3}, \frac{2}{3}, 0\}, \{-\frac{2}{3}, -\frac{1}{3}, 0\}\}$
$\{\frac{2}{3}, \frac{1}{3}, 0\}$	6	$\{\{\frac{5}{3}, \frac{4}{3}, 0\}, \{\frac{5}{3}, \frac{1}{3}, 0\}, \{\frac{2}{3}, \frac{4}{3}, 0\}, \{\frac{2}{3}, -\frac{2}{3}, 0\}, \{-\frac{1}{3}, \frac{1}{3}, 0\}, \{-\frac{1}{3}, -\frac{2}{3}, 0\}\}$

To generate the wannier90_hr.dat file, simply run the following command:

```
hop[ham, {e1 −> 0, r1 −> 0.02, t1 −> 0.5}]
```

The result is:

```
Generated by MagneticTB
2
7
1    1    1    1    1    1    1
−1   −1    0    1    1    0.02000000    0.00000000
−1   −1    0    2    1    0.00000000    0.00000000
−1   −1    0    1    2    0.00000000    0.00000000
−1   −1    0    2    2    0.02000000    0.00000000
−1    0    0    1    1    0.02000000    0.00000000
−1    0    0    2    1    0.00000000    0.00000000
−1    0    0    1    2    0.50000000    0.00000000
−1    0    0    2    2    0.02000000    0.00000000
0    −1    0    1    1    0.02000000    0.00000000
0    −1    0    2    1    0.50000000    0.00000000
0    −1    0    1    2    0.00000000    0.00000000
0    −1    0    2    2    0.02000000    0.00000000
0     0    0    1    1    0.00000000    0.00000000
0     0    0    2    1    0.50000000    0.00000000
```

```
0    0    0    1    2    0.50000000    0.00000000
0    0    0    2    2    0.00000000    0.00000000
0    1    0    1    1    0.02000000    0.00000000
0    1    0    2    1    0.00000000    0.00000000
0    1    0    1    2    0.50000000    0.00000000
0    1    0    2    2    0.02000000    0.00000000
1    0    0    1    1    0.02000000    0.00000000
1    0    0    2    1    0.50000000    0.00000000
1    0    0    1    2    0.00000000    0.00000000
1    0    0    2    2    0.02000000    0.00000000
1    1    0    1    1    0.02000000    0.00000000
1    1    0    2    1    0.00000000    0.00000000
1    1    0    1    2    0.00000000    0.00000000
1    1    0    2    2    0.02000000    0.00000000
```

5 FAQ

1. What should I do if there is no response after running the program after installation?

   Answer: Check the installation steps carefully, first run examples in the examples.nb file, and if you still can't run the program, you can open an issue on https://github.com/zhangzeyingvv/MagneticTB/issues.

2. Why do the Wyckoff points of molybdenum atoms in the example of molybdenum disulfide in the example.nb file not match those on the BCS website?

   Answer: This example is to completely reproduce the results of Phys. Rev. B 88, 085433 (2013). The default hexagonal lattice in the database has an angle of $\frac{2\pi}{3}$ between the $\boldsymbol{a}$ and $\boldsymbol{b}$ axes, while the lattice vectors used in Phys. Rev. B 88, 085433 (2013) have an angle of $\frac{\pi}{3}$ between the $\boldsymbol{a}$ and $\boldsymbol{b}$ axes. Therefore, to ensure that the symmetry operations are consistent with the original package, the symmetric operations in the database were converted using `sgoptr=MapAt[FullSimplify[tran . # . Inverse@tran] &, sgop, {;; , 2}]`, and the resulting Hamiltonian matches the literature completely. Refer to the article "Comput. Phys. Commun. 265, 107993 (2021)" of SpaceGroupIrep for the specific conversion method. In the future, MagneticTB will integrate automatic conversion functions. However, it should be noted that no matter how the original package is taken, the resulting Hamiltonian band degeneracy and number of parameters will remain the same, and only the form will differ.

3. If non-magnetic systems can be calculated, why is the software called MagneticTB??

   Answer: Because the 230 space groups are subsets of 1651 magnetic groups, MagneticTB can construct TB models of any of the 1651 magnetic groups, including non-magnetic materials.

4. What should I do if I find a bug?

   Answer: Ask questions in the WeChat group (recommended), ask questions at https://github.com/zhangzeyingvv/MagneticTB/issues, or send me an email at zhangzeyingvv@gmail.com.

5. In addition to wanting to construct TB models, how can I also calculate the topological properties of the models?

   Answer: The Wilson loop-related functions are in $UserBaseDirectory/MagneticTB/WilsonLoop.wl, which can be referred to. However, this part of the code has not been systematically tested, and this document will not provide instructions on how to use the Wilson loop-related functions. You can also convert the TB model to the wannier90_hr.dat file and use other software to calculate it.

6. Does MagneticTB support the construction of TB models for spin space groups?

   Answer: Not currently, the spin space group related code is still in testing and will be released in the future.

7. Can I consider spin but not spin-orbit coupling?

   Answer: Currently, no. Spin space group related code must be updated before this can be achieved.

8. Are there any reference books for Mathematica?

   Answer: I recommend ”Mathematica: A Practical Approach” or ”The Mathematica Book” written by Wolfram himself who created the WOLFRAM language.

9. Can I forward the manual?

   Answer: Sure! But do not make any changes to the manual when forwarding it.

10. How to cite MagneticTB?

Answer:Zeying Zhang, Zhi-Ming Yu, Gui-Bin Liu, Yugui Yao, Computer Physics Communications, 270, 108153 (2022).

Bibtex:

```
@article{ZHANG2022108153,
  title  = {MagneticTB: A package for tight—binding model of magnetic and non—
      magnetic materials},
  journal  = {Computer Physics Communications},
  volume = {270},
  pages = {108153},
  year  = {2022},
  doi  = {https://doi.org/10.1016/j.cpc.2021.108153},
  url  = {https://www.sciencedirect.com/science/article/pii/S0010465521002654},
  author = {Zeying Zhang and Zhi—Ming Yu and Gui—Bin Liu and Yugui Yao}
}
```