Project in Physics and Astronomy, 15c:
Tight Binding Modelling of Materials

Raquel Esteban Puyuelo

October 28, 2015

Abstract

In this project, a parametrized tight binding (TB) code is developed in order to describe the essential physics of real materials using a minimal model. We have used this code to compute the band structure of different materials where the primary inputs are the hopping parameters, obtained from a \(N\text{th}\) order muffin-tin orbital (NMTO) method. The code has been tested for a single atom in a unit cell having only one effective orbital (Li) as well as for many atoms having more than one orbital (NiS and IrO\(_2\)). A successful reproduction of the density functional theory band structure for all these three systems implies that this code can be applied in general to any real material. We have also analyzed the effect of various nearest neighbor interactions on the electronic structure of these systems.

Contents

1 Introduction 2

2 Theory 2

2.1 Description of the Tight Binding Model 2

2.2 Examples 3

2.2.1 One-dimensional chain: single s-band 3

2.2.2 One dimensional chain with two atoms 4

2.2.3 Three dimensional example: Lithium 5

2.2.4 Summary 5

2.3 Extraction of the Tight Binding parameters 6

2.3.1 Density Functional Theory 6

2.3.2 Fitting of TBM parameters 7

3 Code operation 7

3.1 Inputs 7

3.2 Working scheme 7

4 Applications 8

4.1 Li 8

4.2 NiS 9

4.3 IrO\(_2\) 10

5 Conclusion 11

6 Appendix 13
1 Introduction

Density functional theory is one of the most popular theories nowadays for the description of electronic structure properties of various materials. However, these calculations require large amounts of computing time in order to achieve the accuracy needed for the appropriate description of these complex systems. In addition, the results of the first-principles calculations are often not so straight forward to interpret. Thus, it is needed to have a simple model, which contains the essential physics necessary for a description of complex systems, and also provide the liberty to analyze the effect of various parameters separately which primarily dictate the nature of the systems.

In view of that, I have developed a code in the framework of parametrized tight-binding method taking the parameters from the NMTO downfolding calculation. This code has been tested successfully for various kinds of real systems. The thesis has been organized as follows. In section-II, I discuss the theory of the TB model. Section-III has been devoted in describing the code, Section-IV includes all the test cases, followed by summary in section-V.

2 Theory

2.1 Description of the Tight Binding Model

The tight binding approach to the electronic band structure calculation is used extensively in condensed matter physics and has as its starting point the wavefunctions of the free atoms.

In the tight-binding model (TBM) of electronic structures, single-electron wave functions are expanded in terms of atomic orbitals, centered around each atom. Thus this method is closely related to the linear combination of atomic orbitals (LCAO) method.

The simplest model consists in assuming that there is only one orbital in each atomic site (single band model), but it can be easily extended to more complex cases. The localized basis set is \( \phi(\vec{r} - \vec{R}_j) \), where \( \vec{R}_j \) is the position of the \( j \)th atom and \( \phi(\vec{r}) \) is the orbital wavefunction. In its usual form, the TBM is a single electron model where each electron is described by the Hamiltonian

\[
H_l = -\frac{\hbar^2}{2m} \nabla^2 + V(\vec{r}_l - \vec{R}_j). \tag{2.1}
\]

It can be rewritten in the following way:

\[
H_l = -\frac{\hbar^2}{2m} \nabla^2 + V(\vec{r}_l - \vec{R}_j) + \sum_j V(\vec{r}_l - \vec{R}_j). \tag{2.2}
\]

This Hamiltonian has two parts: the first and second can be regarded as on-site terms (\( H_l^0 \)), while the last accounts for the interactions between site \( l \) and its neighbors. As it is very small, it is usually treated as a perturbation (\( \delta V \)), so the Hamiltonian results in \( H_l = H_l^0 + \delta V \). The wavefunction of the TBM Hamiltonian must obey the Bloch’s theorem

\[
\psi_k(\vec{r}) = \sum_{\vec{R}_j} \phi^\dagger_{\vec{R}_j} \phi(\vec{r} - \vec{R}_j) \tag{2.3}
\]

in the form of one atom/unit cell. If there are two (A and B), it takes the form

\[
\psi_k(\vec{r}) = a_k \psi_k^A + b_k \psi_k^B, \tag{2.4}
\]

where

\[
\psi_k^\alpha(\vec{r}) = \sum_{\vec{R}_j} \phi^\dagger_{\vec{R}_j} \phi^\alpha(\vec{r} + \delta_\alpha - \vec{R}_j). \tag{2.5}
\]

The goal of this method is to calculate the band structure \( \epsilon_k \) by solving the Schrödinger equation

\[
H_k \psi_k = \epsilon_k \psi_k, \tag{2.6}
\]

where \( H_k \) is the Fourier transform of the Hamiltonian in the real space appearing in previous equations. The matrix equation is achieved by multiplying from the left with \( \psi_k^\dagger \)

\[
\psi_k^\dagger \hat{H}_k = \epsilon_k \psi_k^\dagger \psi_k, \tag{2.7}
\]

where

\[
\psi_k^\dagger = (a_k^*, b_k^*), \tag{2.8}
\]

the Hamiltonian matrix is

\[
H_k^{\alpha,\beta} = \left\langle \psi_k^\alpha(\vec{r}) \right| \hat{H}_k \left| \psi_k^\beta(\vec{r}) \right\rangle \tag{2.9}
\]
and the overlap matrix is
\[ S_{\alpha \beta}^k = \langle \psi^\alpha_k(r) | \psi^\beta_k(r) \rangle. \] (2.10)
The overlap corrections are often neglected as we will do in this project.

In general, the expression for the Hamiltonian is
\[ H^\alpha_\beta = \sum_{\vec{R}_1} \alpha \beta \sum_{\vec{R}_1} e^{i \vec{k} \cdot (\vec{R}_1 - \vec{R}_n)} \times \]
\[ \langle \phi^\alpha_k(r+\vec{\alpha}_n - \vec{R}_1) | H | \phi^\beta_k(r+\vec{\beta}_n - \vec{R}_k) \rangle. \] (2.11)
Using translational invariance and (2.2) the Hamiltonian results in:
\[ H^\alpha_\beta = N \sum_{\vec{R}_1} \alpha \beta \sum_{\vec{R}_1} e^{i \vec{k} \cdot \vec{R}_1} \times \]
\[ \langle \phi^\alpha_k(r) | H^0 + \delta V | \phi^\beta_k(r+\vec{\beta}_n - \vec{R}_1) \rangle = \]
\[ N(e^{i \alpha \beta}^{\vec{k}} + \epsilon^0_\alpha^{\vec{k}} + \epsilon^\alpha_\beta^{\vec{k}}), \] (2.12)
the overlap matrix in:
\[ S_{\alpha \beta}^k = \sum_{\vec{R}_1} \alpha \beta \sum_{\vec{R}_1} e^{i \vec{k} \cdot \vec{R}_1} \times \]
\[ \langle \psi^\alpha_k(r+\vec{\alpha}_n - \vec{R}_1) | \psi^\beta_k(r+\vec{\beta}_n - \vec{R}_1) \rangle \] (2.13)
and the hopping parameters can be expressed as
\[ t^\alpha_\beta = \sum_{\vec{R}_1} \alpha \beta \sum_{\vec{R}_1} e^{i \vec{k} \cdot \vec{R}_1} \times \]
\[ \langle \psi^\alpha_k(r+\vec{\beta}_n - \vec{R}_1) | \delta V | \psi^\beta_k(r+\vec{\beta}_n - \vec{R}_1) \rangle \] (2.14)

In practice, this eigenvalue problem should be addressed using the diagonalization of the matrix \( (H^0_k - \epsilon_k) \) at each point in the reciprocal space in the path along which we want to calculate the band structure, with
\[ H \approx \sum_n H_{\text{at}}(n) + \sum_{(n,n')} U_{n,n'}, \] (2.15)
where \( (n,n') \) means that only nearest-neighbor interaction \( U \) is considered (it is easily extensible to further neighbors).

2.2 Examples

2.2.1 One-dimensional chain: single \( s \)-band

Consider an infinite linear chain formed by identical atoms separated by a distance \( a \). Here we consider the simplest case, where each atom in the position \( n \) has only one \( s \)-type symmetrical orbital which interacts with the \( s \)-orbitals of the atoms located in sites \( n-1 \) and \( n+1 \) (only nearest neighbors interactions) to form the crystal states.

**Figure 1:** One dimensional chain with one atom in the basis

In order to calculate the band structure of the system we need to solve (2.6) applying the TB approximation explained above. In this case, we only consider the nearest neighbors interactions. Hence, the hopping interactions are zero unless \( n' = n-1 \) or \( n' = n+1 \):
\[ H \approx \sum_n [H_{\text{at}}(n) + U(n,n-1) + U(n,n+1)]. \] (2.16)
Also, the atomic Schrödinger equation in (2.6) can be written in Dirac’s notation
\[ H_{\text{at}} | \psi_n \rangle = \epsilon | \psi_n \rangle, \] (2.17)
where \( E_n = \epsilon \) is the energy of the atomic electron, which is known.

The energy dispersion can be calculated as following if we assume the orthogonality of the wavefunction and use the expressions in (2.17) and (2.16):
\[ E_k = \langle \Psi_k | H | \Psi_k \rangle = \langle \Psi_k | \sum_n H_{\text{at}}(n) | \Psi_k \rangle + \langle \Psi_k | \sum_n [U(n,n-1) + U(n,n+1)] | \Psi_k \rangle \] (2.18)
In the first neighbors approximation the interaction terms are nonzero in the following cases:
\[ \langle \psi_n U(n,n+1) | \psi_{n'} \rangle = \begin{cases} -t(n'' = n, n' = n + 1) \\ 0 \text{ otherwise} \end{cases} \] (2.19)
Then, taking in the Bloch factors the dispersion relation results in the following expression, which is plotted in figure 2
\[ E_k = \epsilon - t(e^{-ika} + e^{ika}) = \epsilon - 2t \cos(ka). \] (2.20)
Again, we want to calculate the band structure by solving a Schrödinger equation and we will use the Tight Binding method assumptions:

1. The Hamiltonian is as Equation (2.23).
2. The atomic problem is known:
   \[ H_{at} \psi_{\gamma n}(x) = E_{\gamma n} \psi_{\gamma n}(x) \]  
   (2.24)
3. The atomic wavefunction \( \psi_{\gamma n} \) is localized in each site.
4. The crucial TB assumption \( C_{\gamma} \) are unknown:
   \[ \Psi_k(x) = \sum_{n} e^{ikna} \sum_{\gamma} C_{\gamma} \psi_{\gamma n}(x) \]  
   (2.25)

To determine the dispersion relation we multiply the Schrödinger equation from the left by each of the two different atomic orbitals and integrate:

\[ \langle \psi_{An}|H|\Psi_k \rangle = E_k \langle \psi_{An}|\Psi_k \rangle \]  
(2.26)
\[ \langle \psi_{Bn}|H|\Psi_k \rangle = E_k \langle \psi_{Bn}|\Psi_k \rangle \]  
(2.27)

In the tight binding approach, only the on-site terms are retained on the left side of the equation and only the nearest neighbors’ terms in the right side:

\[ \varepsilon_A C_A - t(1 + e^{-ika}) = E_k C_A \]  
(2.28)
\[ \varepsilon_B C_B - t(1 + e^{ika}) = E_k C_B, \]  
(2.29)

where \( \varepsilon_\gamma = \langle \psi_{\gamma n}|H|\psi_{\gamma n} \rangle \) and \( t = \langle \psi_{\gamma n}|U(n,n-1)|\psi_{\gamma n-1} \rangle = \langle \psi_{\gamma n}|U(n,n+1)|\psi_{\gamma n+1} \rangle \), which is the same due to symmetry. This can be written in a matrix form:

\[ \begin{pmatrix} \varepsilon_A - E_k & -t(1 + e^{-ika}) \\ -t(1 + e^{ika}) & \varepsilon_B - E_k \end{pmatrix} = \begin{pmatrix} C_A \\ C_B \end{pmatrix} = 0 \]  
(2.30)

The equation will have solutions if the determinant of the first matrix equals zero:

\[ E_k - (\varepsilon_A + \varepsilon_B)E_k + \varepsilon_A \varepsilon_B - 2t^2[1 + \cos(ka)] = 0. \]  
(2.31)

This results in the dispersion relation for the energy:

\[ E_k = \frac{\varepsilon_A + \varepsilon_B}{2} \pm \sqrt{(\varepsilon_A - \varepsilon_B)^2 + 8t^2 \cos^2(ka)}. \]  
(2.32)

This means we will get two solutions: a valence and a conduction band separated by a "gap", as figure 4 shows.
The band structure of 1D chain formed by two atoms with 1s orbital each has two s-type bands separated by a gap. In this plot, \( \varepsilon_A = -\varepsilon_B = 1/2 \) and \( t = 1 \).

2.2.3 Three dimensional example: Lithium

The most complicated system that can still be easily solved analytically is a three-dimensional structure that has only one atom per unit cell, such as Lithium. This metal has a body centered cubic (bcc) Bravais lattice with a lattice constant \( a \) and has one atom per primitive cell, each of which having one valence electron in a 2s orbital.

The lattice vectors of the bcc structure are often taken as:

\[
\begin{align*}
\hat{a}_1 &= \frac{a}{2}(\hat{x} + \hat{y} - \hat{z}) \\
\hat{a}_2 &= \frac{a}{2}(-\hat{x} + \hat{y} + \hat{z}) \\
\hat{a}_3 &= \frac{a}{2}(\hat{x} - \hat{y} + \hat{z})
\end{align*}
\]

The number of bands a system will have, i.e. the dimension of the Hamiltonian matrix, can be calculated knowing the number of atoms per unit cell \( n \) and the number of orbitals each atom has \( m \): as \( n \times m \).
• One atom/unit cell with one orbital each: $1 \times 1 = 1$ band.
• One atom/unit cell with three orbitals each: $1 \times 3 = 3$ bands.
• Two atoms/unit cell with four orbitals each: $2 \times 4 = 8$ bands.
• Two atoms/unit cell, one has one orbital and the other has 3 orbitals: $1 \times 1 + 3 = 4$ bands.

2.3 Extraction of the Tight Binding parameters

In many cases the matrix elements $H_{mn}$ (on-site energies and hopping parameters) are found using either experimental data or to an independently calculated band structure using ab-initio methods and therefore it is important to have some basic knowledge about Density Functional Theory (DFT).

2.3.1 Density Functional Theory

DFT is an ab-initio theory of correlated many-body systems which is a primary tool for the calculation of electronic structure in condensed matter but can also be used to study molecules and other finite systems. Its fundamental idea is that any property of a many-body interacting system can be viewed as a functional of the ground state density, which reduces the number of degrees of freedom drastically. An overview to the Hohenberg-Kohn theorems and the Kohn-Sham equation follows.

The electronic properties of a system containing $N$ electrons can be obtained by solving the many-body Schrödinger equation (for the non-relativistic time-independent case):

$$ H \psi(\vec{r}_1, \vec{r}_2, ..., \vec{R}_1, \vec{R}_2, ...) = E \psi(\vec{r}_1, \vec{r}_2, ..., \vec{R}_1, \vec{R}_2, ...), $$

where $\psi$ is the all electron wave function which depends on the positions of the electrons ($\vec{r}_i$) and the ions ($\vec{R}_I$) and the full Hamiltonian is

$$ H = -\frac{\hbar^2}{2m_e} \sum_i \nabla_i^2 - \sum_I \frac{\hbar^2}{2m_I} \nabla_I^2 - \sum_i \sum_{I,I'} \frac{Z_I Z_{I'} e^2}{|\vec{r}_i - \vec{R}_I|} + \frac{\hbar^2}{2m_e} \sum_{i,j \neq i} \frac{e^2}{|\vec{r}_i - \vec{r}_j|} + \frac{1}{2} \sum_{i \neq j} \frac{Z_I Z_{I'} e^2}{|\vec{R}_I - \vec{R}_{I'}|}. $$

The first two terms in (2.39) are the kinetic energies of the electrons and the ions where $m_e$ and $m_I$ in are the mass of the electron and the Ith ion, respectively. The third term stands for the Coulomb interaction between the ions and electrons, with $Z_I$ being the charge of the first ones, and the two last terms represent the electron-electron and the ion-ion Coulomb interactions. Using the Born-Oppenheimer approximation [1] (frozen nucleus) relying on the fact that the ions are much heavier than electrons, results in a simpler Hamiltonian (in atomic units):

$$ H = -\frac{1}{2} \sum_i \nabla_i^2 - \sum_{i,I} \frac{Z_I}{|\vec{r}_i - \vec{R}_I|} + \frac{1}{2} \sum_{i \neq j} \frac{1}{|\vec{r}_i - \vec{r}_j|}. $$

Even with this approximation, a many body equation for a system of $N$ interacting system needs to be solved. A big step was made by the introduction of the Thomas-Fermi-Dirac approximation [2],[3],[4], when the many-body wave function was substituted by the electron density $n$ of the system, which theoretically simplifies the problem because the number of variables get reduced to just 3. The foundations of Density Functional Theory (DFT) were established on this approximation, from which follows that electronic properties can be calculated using $n(\vec{r})$ and that the total energy of the system is a functional of this density, $E[n(\vec{r})]$.

The formulation of DFT is based on the two Hohenberg-Kohn theorems, which state that any property of an interacting system can be obtained from the ground state electron density $n_0(\vec{r})$ via the minimization of the total energy functional, now $E[n_0(\vec{r})]$ even if for an interacting system. The many body equation in (2.38) is then replaced by a single particle equation, the Kohn-Sham (KS) equation:

$$ H_{eff}(\vec{r}) \phi_i(\vec{r}) = \left[ -\frac{\hbar^2}{2m_e} \nabla_i^2 + V_{eff}(\vec{r}) \right] \phi_i(\vec{r}) = \varepsilon_i \phi_i(\vec{r}), $$

where $V_{eff}$ includes the external potential due to the nuclei, the Hartree potential and the exchange-correlation potential $V_{xc} = \frac{\delta E_{xc} [n]}{\delta n[n]}$. Solving the
KS equation in a self-consistent way yields the total energy:

\[ E = \sum_{i}^{N} \epsilon_{i} - \frac{1}{2} \int d\vec{r} d\vec{r}' n(\vec{r}) n(\vec{r}') \frac{1}{|\vec{r} - \vec{r}'|} - \int d\vec{r} V_{xc}(\vec{r}) n(\vec{r}) + E_{xc}[n], \]  

(2.42)

where \( E_{xc} \) is the exchange-correlation functional. There is not an explicit form for \( E_{xc} \), so different approximations need to be implemented to solve the problem. Just to mention some, the Local Density Approximation (LDA) \[5\] takes the electron density to be locally the one of a uniform electron gas and the Generalized-Gradient Approximation (GGA) \[6\], \[7\] takes also into account the derivative of the density at the point.

In short, the problem of solving a many-body Schrödinger equation for an interacting system using the complete wave-function has been replaced by solving the non-interacting KS equation using a chosen basis. Then, through the minimization of the resulting energy functional, the ground state electronic properties can be obtained.

### 2.3.2 Fitting of TBM parameters

For this project, the hopping parameters and the on-site energies were obtained from a \( N^{th} \) order muffin-tin orbital (NMTO) method. For a mathematical derivation on how the fitting of the TBM parameters can be done, refer to \[8\].

### 3 Code operation

In order to obtain the band structure of more complex systems in which the analytic method can be too complicated to solve, I have written a general code (included in Section \[6\]) that calculates the band dispersion along various high symmetry directions using the tight binding method.

#### 3.1 Inputs

The following files are the inputs for the general TBM that we developed:

1. Nearest neighbor file (NNmap.in): it contains the position vectors of each atom, as well as its distance referred to the central atom, which allows a classification in terms of first, second... nearest neighbors.

2. TBM parameters file (HAMR.in): it contains the TB parameters in a matrix form (hopping parameters and on-site terms in Ryd) for each atom, obtained from a \( N^{th} \) order muffin-tin orbital (NMTO) method. This matrices have the dimension of the number of orbitals existing in each atomic sites and will be built up to bigger matrices in the body of the code.

3. k-points file (kpoints.txt): the k-points of the path along which the band structure will be calculated are provided, separated into segments, including the high symmetry points such as \( \Gamma \), K, L... that depend on the crystal structure in study. This file is the output of a small code that needs to be run before (kpoints.f), that needs to be provided with the number of symmetry points and its coordinates (SYML.in).

4. Dimension of the system: The program will ask for the number of atoms and orbitals the studied system has, in order to determine the dimension of the Hamiltonian matrices as it was explained in section \[2.2.4\].

#### 3.2 Working scheme

In each k-point the energy values are calculated in order to generate a file that is ready to plot. Before, the small tight binding matrices have to be built into matrices that have the same dimension as the Hamiltonian. When the neighbor vectors are introduced to the Hamiltonian, it is diagonalized at every k-point using a subroutine that uses the DGEEV diagonalization code from LAPACK \[9\], as Figure \[7\] depicts.
4 Applications

In this section we apply the general code to different systems in order to obtain their band structure.

4.1 Li

The first material we approach as a test is Lithium because we know the analytic solution for the TBM, calculated in 2.2.3. As Figure 5 shows, lithium has a body centered cubic structure and one atom in the basis, with a space group 194. Lithium’s electronic configuration is \([\text{He}]2s^1\), that is only one valence orbital. This means the dimension of the Hamiltonian matrix is \(1 \times 1 = 1\) and the exact solution can be calculated analytically without diagonalization as it was shown in Section 2.2.3. The path followed to plot the band structure is \(\Gamma-N-H-\Gamma\) (see Figure 8 and Table 1).

The comparison between the band structure calculated computationally and the analytic solution from Section 2.2.3 can be found in Figure 9. As expected, we obtain one band (one eigenvalue for each \(k\)-point) and both results agree. In this particular problem, and due to the fact that the Hamiltonian matrix has dimension 1, no diagonalization is needed to compute the eigenvalues, as it was stated before.
4.2 NiS

The next problem I address is hexagonal NiS, as whether its low-temperature (LT) phase is metallic or insulator has been very much debated. Its high-temperature (HT) phase is known to be a conducting Pauli paramagnetic state, but the antiferromagnetic and low conductivity LT phase has raised controversy due to its unusual transport/electronic properties. There have been as many experimental works claiming it to be an insulator as a metal, and this is why obtaining the TBM electronic structure is interesting.

Hexagonal NiS crystallizes in the NiAs structure with a space group of 194 (details can be found in [11]) and has only two inequivalent Ni atoms per unit cell, as Figure 10 shows. The path followed to plot the band structure is $\Gamma$-K-M-$\Gamma$-A-L-H-A. Table 2 contains the coordinates of the high symmetry points for NiS structure, which are plotted in Figure 11.

Although we will not calculate the band structure for Sulfur because it lies far from the Fermi level, its hybridization with Nickel atoms has already been taken into account in the hopping parameters.

Tables 3, 4 and 5 include some of the TMB parameters matrices used in the calculations as inputs.

---

Table 2: High symmetry points of NiS crystal structure, referred to the reciprocal lattice basis vectors ($\vec{k} = u\vec{b}_1 + v\vec{b}_2 + w\vec{b}_3$; $(u, v, w)$)

<table>
<thead>
<tr>
<th>Point</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma$</td>
<td>(0,0,0)</td>
</tr>
<tr>
<td>A</td>
<td>(0,0,$\frac{1}{2}$)</td>
</tr>
<tr>
<td>K</td>
<td>($\frac{1}{2}$,$\frac{1}{2}$,0)</td>
</tr>
<tr>
<td>H</td>
<td>($\frac{1}{2}$,$\frac{1}{2}$,$\frac{1}{2}$)</td>
</tr>
<tr>
<td>M</td>
<td>($\frac{1}{2}$,0,0)</td>
</tr>
<tr>
<td>L</td>
<td>($\frac{1}{2}$,0,$\frac{1}{2}$)</td>
</tr>
</tbody>
</table>

Table 3: On-site parameters matrix ($\epsilon$) used for NiS (in Ry)

<table>
<thead>
<tr>
<th>Orbitals</th>
<th>$dx_y$</th>
<th>$dy_z$</th>
<th>$dz^2 - r^2$</th>
<th>$dx_z$</th>
<th>$dx^2 - y^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$dx_y$</td>
<td>-0.1370</td>
<td>0.0425</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>$dy_z$</td>
<td>0.0425</td>
<td>-0.1099</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>$dz^2 - r^2$</td>
<td>0.0000</td>
<td>0.0000</td>
<td>-0.1587</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>$dx_z$</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>-0.0425</td>
</tr>
<tr>
<td>$dx^2 - y^2$</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>-0.1370</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Table 4: Example of the first nearest neighbor hopping parameters matrix ($t_1$) used for NiS (in Ry)

<table>
<thead>
<tr>
<th>Orbitals</th>
<th>$dx_y$</th>
<th>$dy_z$</th>
<th>$dz^2 - r^2$</th>
<th>$dx_z$</th>
<th>$dx^2 - y^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$dx_y$</td>
<td>0.0045</td>
<td>-0.0217</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>$dy_z$</td>
<td>-0.0084</td>
<td>0.0061</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>$dz^2 - r^2$</td>
<td>0.0000</td>
<td>0.0000</td>
<td>-0.0252</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>$dx_z$</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>-0.0217</td>
<td>-0.0084</td>
</tr>
<tr>
<td>$dx^2 - y^2$</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>-0.0061</td>
<td>-0.0015</td>
</tr>
</tbody>
</table>
Table 5: Example of the second nearest neighbor hopping parameters matrix (t₂) used for NiS (in Ry)

<table>
<thead>
<tr>
<th>orbitals</th>
<th>dxy</th>
<th>dyz</th>
<th>dz² - 1</th>
<th>dxz</th>
<th>dx² - y²</th>
</tr>
</thead>
<tbody>
<tr>
<td>dxy</td>
<td>0.0071</td>
<td>0.0023</td>
<td>-0.0029</td>
<td>-0.0047</td>
<td>0.0072</td>
</tr>
<tr>
<td>dyz</td>
<td>0.0083</td>
<td>-0.0057</td>
<td>-0.0098</td>
<td>0.0082</td>
<td>0.0012</td>
</tr>
<tr>
<td>dz² - 1</td>
<td>0.0019</td>
<td>-0.0085</td>
<td>0.0069</td>
<td>0.0064</td>
<td>-0.0044</td>
</tr>
<tr>
<td>dxz</td>
<td>-0.0009</td>
<td>-0.0009</td>
<td>0.0042</td>
<td>-0.0015</td>
<td>0.0010</td>
</tr>
<tr>
<td>dx² - y²</td>
<td>0.0022</td>
<td>0.0044</td>
<td>0.0039</td>
<td>-0.0050</td>
<td>0.0017</td>
</tr>
</tbody>
</table>

The band structure plotted in Figure 12 shows the results using our TBM (dots) over the DFT band structure.

Figure 12: Comparison between the bands obtained from the tight binding method developed in this project (red dots) and a self-consistent calculation (black lines) for hexagonal NiS

4.3 IrO₂

Our next system to study is IrO₂, which is more computationally challenging as it has more than one orbital per atom as well as more than one atom per unit cell. IrO₂ is the most simplest compound in the family of iridates which shows novel electronic and magnetic properties. Therefore a tight binding analysis of the bands near Fermi level will be quite significant to understand the novel optical and electronic properties of this system.

IrO₂ crystallizes in a tetragonal rutile structure with space group P4₂/mnm. There are two Ir atoms per unit cell and four O ions, so each Ir atom is surrounded by 6 O atoms in a distorted octahedral environment. The top panel of Figure 13 shows the crystal structure of IrO₂. Each Ir ion is octahedrally coordinated by O ions. In the bottom panel of Figure 13 we show how IrO₆ octahedral networks interact with each other. Details of the structure can be found at [12].

Figure 13: IrO₂ crystal structure (Ir atoms in violet and O in red) and tetragonal bondings

In a single transition metal ion, the five d-orbitals are energy degenerated, but when ligand atoms, such as oxygen, some of the Ir orbitals experience more interaction with the O ones depending on the geometry of the bonding, which leads to energy splitting due to the electrostatic environment. If we consider an octahedral geometry such as the IrO₂, the ligand atoms (O) approach the Ir center along the x, y and z axes and therefore the electrons in the orbitals lying along these axes will experience a greater repulsion than the other. This means that it is more energetically favorable to place an electron in the orbitals that do not lie along the axis, causing an energy splitting. Precisely, the orbitals dₓ² and dₓ² - y² (they are called e_g orbitals) have a higher energy than the dₓᵧ, dₓz and dᵧz (t₂g orbitals), which are more stable.

The k-path followed in this calculation will be Γ-

October 28, 2015 10 Project in Physics and Astronomy, 15c
X-M-Γ-Z-R-A-M. These symmetry points can be observed in Figure 14, which represents IrO$_2$ first Brillouin zone, and its coordinates are in Table 6.

![Figure 14: First Brillouin zone for IrO$_2$ structure with its high symmetry points](image)

**Table 6:** High symmetry points of IrO$_2$ crystal structure, referred to the reciprocal lattice basis vectors ($\mathbf{k} = u\mathbf{b}_1 + v\mathbf{b}_2 + w\mathbf{b}_3$: (u,v,w))

<table>
<thead>
<tr>
<th>Point</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Γ</td>
<td>(0,0,0)</td>
</tr>
<tr>
<td>X</td>
<td>(0,$\frac{1}{2}$,0)</td>
</tr>
<tr>
<td>M</td>
<td>($\frac{1}{2}$,$\frac{1}{2}$,0)</td>
</tr>
<tr>
<td>Z</td>
<td>(0,0,$\frac{1}{2}$)</td>
</tr>
<tr>
<td>R</td>
<td>(0,$\frac{1}{2}$,$\frac{1}{2}$)</td>
</tr>
<tr>
<td>A</td>
<td>($\frac{1}{2}$,$\frac{1}{2}$,$\frac{1}{2}$)</td>
</tr>
</tbody>
</table>

The band structure plotted in Figure 15 shows the results for a three-orbital model for IrO$_2$ ($t_{2g}$ orbitals) using our TBM (dots) over the DFT band structure calculated in [13]. Tables 7, 8, 9 and 10 include some of the TMB parameters matrices used in the calculations as inputs.

**Table 7:** On-site parameters matrix ($\varepsilon$) used for IrO$_2$ (in Ry)

<table>
<thead>
<tr>
<th>Orbitals</th>
<th>$3z^2-1$</th>
<th>dxz</th>
<th>$dx^2-y^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d3z^2-1$</td>
<td>-0.0930</td>
<td>0.0000</td>
<td>-0.0002</td>
</tr>
<tr>
<td>dxz</td>
<td>0.0000</td>
<td>-0.0507</td>
<td>0.0000</td>
</tr>
<tr>
<td>$dx^2-y^2$</td>
<td>-0.0002</td>
<td>0.0000</td>
<td>-0.0420</td>
</tr>
</tbody>
</table>

**Table 8:** Example of a first nearest neighbor parameter ($t_1$) matrix used for IrO$_2$ (in Ry)

<table>
<thead>
<tr>
<th>Orbitals</th>
<th>$3z^2-1$</th>
<th>dxz</th>
<th>$dx^2-y^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d3z^2-1$</td>
<td>0.0007</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>dxz</td>
<td>-0.0047</td>
<td>0.0006</td>
<td>0.0002</td>
</tr>
<tr>
<td>$dx^2-y^2$</td>
<td>0.0003</td>
<td>-0.0234</td>
<td>-0.0162</td>
</tr>
</tbody>
</table>

**Table 9:** Example of a second nearest neighbor parameter ($t_2$) matrix used for IrO$_2$ (in Ry)

<table>
<thead>
<tr>
<th>Orbitals</th>
<th>$3z^2-1$</th>
<th>dxz</th>
<th>$dx^2-y^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d3z^2-1$</td>
<td>0.0003</td>
<td>0.0000</td>
<td>-0.0010</td>
</tr>
<tr>
<td>dxz</td>
<td>0.0000</td>
<td>0.0057</td>
<td>0.0000</td>
</tr>
<tr>
<td>$dx^2-y^2$</td>
<td>-0.0010</td>
<td>0.0000</td>
<td>-0.0022</td>
</tr>
</tbody>
</table>

**Table 10:** Example of a third nearest neighbor parameter ($t_3$) matrix used for IrO$_2$ (in Ry)

**Figure 15:** Comparison between the bands obtained from the tight binding method developed in this project (red dots) and a self consistent calculation (black lines) for IrO$_2$

5 Conclusion

A tight binding model code has been written from scratch and it has been used to calculate the band structures of three different materials. Although the band structures that result from the TBM code developed for the project can not reproduce completely the ones obtained from the DFT based method, we can appreciate some of the important
features such as degeneracy of the bands and bandwidth. Better agreement can be achieved including more neighbors to the calculations. Also, the discrepancies between the DFT bands and the ones resulting from the code can be ascribed to a poor treatment of the inputs: more carefully calculations must be performed taking into account the non equivalence of some atoms in the cell, which therefore will have different hopping parameters. However, it is remarkable how such a basic model can provide good results for simple systems like lithium. Furthermore, it is possible to modify the hopping parameters and other terms in the model in order to try to study which interactions are the most important for a given material, which can complement DFT calculations.

It would be interesting to perform studies including spin polarization in order to be able to describe magnetic systems. Furthermore, one can also think about implementing spin orbit coupling and Hubbard-U terms in the Hamiltonian for the purpose of studying more complex interactions.

References


6 Appendix

This section includes the code that has been developed for the project.

```fortran
PROGRAM TBM
! tight binding method
! Band structure in the 1BZ
!
! info on the dsyev function used:
! http://www.netlib.org/lapack/explore-html/dd/d4c/dsyev_8f.html#a442c43fc
! a5493590f8f26cf42fed4044

implicit none
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! VARIABLES
!
! u*: all variables being u+number are useless
! typ: type of atom (not used here)
! dist: distance between neighbors (varies)
! dist1: 1NN distance
! dist2: 2NN distance
! RNN: 1st neighbors vectors
! R2N: 2nd neighbors vectors
! R3N: 3rd neighbors vectors
! NN: total number of nearest neighbors
! fNN: number of 1st NN
! sNN: number of 2nd NN
! tNN: number of 3rd NN
! epsi: on-site energy
! t1: 1st neighbors hopping parameter
! t2: 2nd neighbors hopping parameter
! t3: 3rd neighbors hopping parameter
!
! Nsp: number of symmetry points in the path
! path_name: segment of the path (examples: GM, MK...)
! div: number of divisions in the path
! kpoint: matrix of path points (x-y-z points for each path segment)
!
! a: lattice parameter (x-y plane)
! c: lattice parameter (z direction)
! dimh: dimension of the hamiltonian matrix
! Nat: number of atoms/cell
! RydeV: conversion from Rydbergs to eV
! i: imaginary unit
! pi: pi number
!
! hop1: sum of the 1NN hopping contributions
! hop2: sum of the 2NN hopping contributions
! hop3: sum of the 3NN hopping contributions
! expo: sum of the exponents of the exponentials for the 1NN
! expo2: sum of the exponents of the exponentials for the 2NN
! expo3: sum of the exponents of the exponentials for the 3NN
```

October 28, 2015 13 Project in Physics and Astronomy, 15c
H: hamiltonian, sum of epsi and hop1, hop2, hop3
E: energy, comes from the diagonalization of H at each k-point

integer n, NN, m, r, q, fNN, sNN, tNN, t, mc, mf, tc, tf, of, oc
integer dimh, Nat, dimorb
parameter (dimorb=5) ! change if needed
parameter (Nat=2) ! change if needed
parameter (dimh = dimorb * Nat)
real*8 pi, E(99,99, dimh), epsi(dimh, dimh)
real*8 hop2 (dimh, dimh), t3(dimh, dimh,99), dist3 , hop3(dimh, dimh)
real*8 t1(dimh, dimh,99), t2(dimh, dimh,99), H(dimh, dimh), eig(dimh)
real*8 Hg(dimh, dimh), prova3(dimh, dimh)
real*8 pepsi (dimorb, dimorb), pt1 (dimorb, dimorb,99)
real*8 pt2 (dimorb, dimorb,99), pt3 (dimorb, dimorb,99)
complex*8 i, expo, expo2, expo3, texpo(dimh, dimh,99)
character*8 u1, u3, u2
integer atom, typm (dimorb,10), j, typ (30)
real*8 dist, u4, u5, u6, dist1, RNN(4,30,30), R2N(4,30,30)
real*8 R3N(4,30,30), a, c, RydeV, dist2 , ca(999)
character*8 u7
integer Nsp, l, path_ind (99), maxdiv (99), p
character (len=2) path_name (99)
real*8 kpoint (99,3,99), div(99,3), vec(99,99), maxE, klength(99)
real*8 W(dimh), EV(dimh, dimh), change(dimh, dimh)

i=(0.0d0,1.0d0) ! imaginary unit
pi=4.0d0*datan(1.0d0) ! pi number
RydeV=13.6057d0

! READ FROM THE NN MAP FILE
vectors:
typm=0
open (unit=20, file=’Ir_NNmap.in’, status=’old’)
do m=1,Nat
t=1

if( m.eq.1) then
  do q=1,3
    read(20,*)
  end do
  read(20,*) u1, atom, dist1 ! read dist1 to compare later
  rewind(20)
end if

do q=1,3
  read(20,*)
end do

! read NN vector components until dist is bigger than dist1
j=1
do
read(20,*) u1, atom, dist, u2, RNN(4,j,m), u3, u4, u5, u6, &
RNN(1,j,m), RNN(2,j,m), RNN(3,j,m)
if(dist.gt.dist1) exit
j=j+1
fNN=j-1
dist2=dist

! same with dist2
backspace(20)
j=1
do
read(20,*) u1, atom, dist, u2, R2N(4,j,m), u3, u4, u5, u6, &
R2N(1,j,1), R2N(2,j,m), R2N(3,j,m)
if(dist.gt.dist2) exit
j=j+1
sNN=j-1
dist3=dist

! same with dist3
backspace(20)
j=1
do
read(20,*) u1, atom, dist, u2, R3N(4,j,m), u3, u4, u5, u6, &
R3N(1,j,m), R3N(2,j,m), R3N(3,j,m)
if(dist.gt.dist3) exit
j=j+1
tNN=j-1
end do
end do
end do
close(20)

NN=fNN+sNN+tNN ! number of nearest neighbors

! READ FROM THE HAMR FILE: TBM PARAMETERS
open(unit=30, file="HAMR.in", status="old")
do j=1,11
read(30,*)
end do
do j=1,dimorb
read (30,*) u7, pepsi(j,:
end do
do n=1,fNN
do j=1,8
read(30,*)
end do
do j=1,2
read(30,*) u7, pt1(j,:,n)
end do
do n=1,sNN
do j=1,8
read(30,*)
end do

! 
! CONSTRUCTION OF BIG TMB MATRICES
!

do mf=1,Nat
    do mc=1,Nat
        ! on-site term
        if(mf.eq.mc) then
            do oc=1,dimorb
                do of=1,dimorb
                    epsi(oc+dimorb*(mc-1),of+dimorb*(mf-1))=pepsi(oc,of)
                end do
            end do
        end if
    end do
end do
read(40,*)
end do

7 format(I3,3(F8.5))
read kx,ky,kz for each path (here 7)
do j=1,30
  read (40,*) ((kpoint(j,m,1),m=1,3), l=1,Nsp-1)
end do
close(40)

kpoint(:,1,:)=kpoint(:,1,:)*2.0d0*pi
kpoint(:,2,:)=kpoint(:,2,:)*2.0d0*pi
kpoint(:,3,:)=kpoint(:,3,:)*2.0d0*pi

! CALCULATE ENERGY VALUES FOR EACH K-POINT IN EACH PATH
! BUILD THE DIM x DIM MATRIX AND DIAGONALIZE
do l=1,Nsp-1
  maxE=maxE+maxdiv(l) !total number of kpoints adding paths
end do

do l=1,Nsp-1 !path loop
do j=1,30 !kpoint loop
  hop1=0
  do n=1,fNN
    do tc=1,Nat
      do tf=1,Nat
        if ((RNN(4,n,tc)).eq.tf) then
          expo=0
          do m=1,3
            expo=expo-i*(kpoint(j,m,l)*RNN(m,n,tc))
          end do
          do oc=1,dimorb
            do of=1,dimorb
              t1(oc+dimorb*(tc-1),of+dimorb*(tf-1),n)= &
              pt1(oc,of,n)
              texpo(oc+dimorb*(tc-1),of+dimorb*(tf-1),n)= &
              t1(oc+dimorb*(tc-1),of+dimorb*(tf-1),n) &
              *exp(expo)
            end do
          end do
        end if
      end do
    end do
  end do
  hop2=0
  do n=1,sNN
    do tc=1,Nat
      do tf=1,Nat
        hop1(r,q)=hop1(r,q)+texpo(r,q,n)
      end do
    end do
    texpo=0
  end do
end do
if ((R2N(4,n,tc)).eq.tf) then
  expo=0
  do m=1,3
    expo=expo-i*(kpoint(j,m,1)*R2N(m,n,tc))
  end do
  do oc=1,dimorb
    do of=1,dimorb
      t2(oc+dimorb*(tc-1),of+dimorb*(tf-1),n) = &
      pt2(oc,of,n)
      texpo(oc+dimorb*(tc-1),of+dimorb*(tf-1),n) = &
      t2(oc+dimorb*(tc-1),of+dimorb*(tf-1),n)* &
      exp(expo)
    end do
  end do
end if
end do
end do
!add all texpo matrices
do r=1,dimh
  do q=1,dimh
    hop2(r,q)=hop2(r,q)+texpo(r,q,n)
  end do
end do
  texpo=0
end do

hop3=0
do n=1,tNN
  do tc=1,Nat
    do tf=1,Nat
      if ((R3N(4,n,tc)).eq.tf) then
        expo=0
        do m=1,3
          expo=expo-i*(kpoint(j,m,1)*R3N(m,n,tc))
        end do
        do oc=1,dimorb
          do of=1,dimorb
            t3(oc+dimorb*(tc-1),of+dimorb*(tf-1),n) = &
            pt3(oc,of,n)
            texpo(oc+dimorb*(tc-1),of+dimorb*(tf-1),n) = &
            t3(oc+dimorb*(tc-1),of+dimorb*(tf-1),n)* &
            exp(expo)
          end do
        end do
      end if
    end do
  end do
  !add all texpo matrices
  do r=1,dimh
    do q=1,dimh
      hop3(r,q)=hop3(r,q)+texpo(r,q,n)
    end do
  end do
  texpo=0
end do

end do

do r=1,dimh
do q=1,dimh

  \[ H(r,q) = \epsilon_{r,q} + h_{o1}(r,q) + h_{o2}(r,q) + h_{o3}(r,q) \]  

end do

eV=H !copy H to the input matrix for the function !
DIAGONALIZE H

! call diagen(EV,eig,dimh)
!
energy values are the eigenvalues
do q=1,dimh
E(j,l,q)=eig(q)
end do

end do
end do

p=1
do l=1,Nsp-1 !path loop
do j=1,30 !kpoint loop
  if (p.eq.1) then
    ca(p )=0.0 d0
  else
    ca(p)=klength(l)/maxdiv(l)+ca(p-1)
  end if
  p=p+1
end do
end do

! WRITE ENERGY IN FILE, SO THEY CAN BE PLOTTED
p=0
2 format (I3,i12(F13.7))
5 format (A,i3(3,F8.4))
open (50, file="E_k3.out", status="unknown")
open (80, file="E_k.out", status="unknown")
do l=1,Nsp-1
  do j=1,30
    write (50,2) p,E(j,l,:)-0.2483709 d0 !correct fermi level
  write (80,1) ca(p+1), E(j,l,:)-0.2483709 d0
  p=p+1
  end do
end do
close (50)

! WRITE MATRICES IN FILE

1 format (12(F8.4))
3 format (A,3(F8.4))
format(A,I3,A,I3,A,3(F8.3))
open(70,file="HDIAO.out",status=unknown)
write(70,*) "FINAL HAMILTONIAN MATRICES (eV)"
write(70,*) "===============================================
write(70,*) "ON - SITE TERM ( EPSI )"
do q=1,dimh
   write (70,1) sngl (eps1(q,:))
end do
write(70,*) "==============================================="
write(70,*) ! blank line
write(70,*) "FIRST NEAREST NEIGHBORS HOPPING (T1)"
do n=1,fNN
   write(70,*) "-----------------------------------------------"
   write(70,3) "Connecting vectors:"
   do q=1, Nat
      write(70,4) "Atom ",q," to ",int(RNN(4,n,q))," : ",RNN(1,n,q),&
                  RNN(2,n,q),RNN(3,n,q)
   end do
   write(70,*) !blank line
   do q=1, dimh
      write(70,1) sngl (t1(q,:,n))
   end do
end do
write(70,*) "==============================================="
write(70,*) ! blank line
write(70,*) "SECOND NEAREST NEIGHBORS HOPPING (T2)"
do n=1,sNN
   write(70,*) "-----------------------------------------------"
   write(70,3) "Connecting vectors:"
   do q=1, Nat
      write(70,4) "Atom ",q," to ",int(R2N(4,n,q))," : ",R2N(1,n,q),&
                  R2N(2,n,q),R2N(3,n,q)
   end do
   write(70,*) !blank line
   do q=1, dimh
      write(70,1) sngl (t2(q,:,n))
   end do
end do
!
write(70,*) "==============================================="
write(70,*) !blank line
write(70,*) "THIRD NEAREST NEIGHBORS HOPPING (T3)"
do n=1,tNN
   write(70,*) "-----------------------------------------------"
   write(70,3) "Connecting vectors:"
   do q=1, Nat
      write(70,4) "Atom ",q," to ",int(R3N(4,n,q))," : ",R3N(1,n,q),&
                  R3N(2,n,q),R3N(3,n,q)
   end do
end do
write(70,*), 'blank line'
do q=1,dimh
    write(70,1) sngl(t3(q,:,:))
end do
!
end do
write(70,*), '=============================================

END PROGRAM

!-------------------!
!---------------------------------------------------------!
! Calls the LAPACK diagonalization subroutine DSYEV !
! input: a(n,n) = real symmetric matrix to be diagonalized !
! n = size of a !
! output: a(n,n) = orthonormal eigenvectors of a !
! eig(n) = eigenvalues of a in ascending order !
!---------------------------------------------------------!
subroutine diasym(a,eig,n)
  implicit none

  integer n,l, inf
  real*8 a(n,n), eig(n), work(n*(3+n/2))

  l=n*(3+n/2)
call dsyev('V','U',n,a,n,eig,work,l,inf)

end subroutine diasym

!---------------------!

October 28, 2015 21 Project in Physics and Astronomy, 15c