

# Tight Binding Notes

Tyler C. Sterling<sup>1,\*</sup>

<sup>1</sup>*Department of Physics, University of Colorado at Boulder, Boulder, Colorado 80309, USA*

(Dated: August 17, 2021)

This is a **work-in-progress** document intended as a clean way to keep notes while writing my tight binding (TB) code. I am writing a PYTHON software package that will be used build tight binding models by fitting to density functional theory (DFT) outputs. It can then be used to solve for the valence band energies at arbitrary points in the Brillouin zone and (eventually) to build electron-phonon Hamiltonians using first-principles electron-phonon coupling parameters. But first, I need to figure out how to write the TB code; that's what these notes are for!

## CONTENTS

I. Introduction	1
II. Introduction to tight binding	3
III. Further reading	4
References	4

## I. INTRODUCTION

These notes are intended to track my progress and record the relevant theory while building a tight binding (TB) code. I am writing a PYTHON software package that will be used build tight binding models by fitting to density functional theory (DFT) outputs. It can then be used to solve for the valence band energies at arbitrary points in the Brillouin zone and (eventually) to build electron-phonon Hamiltonians using first-principles electron-phonon coupling parameters. The code will be *modular* in the sense that a script will return the Hamiltonian that can be solved at arbitrary  $\mathbf{k}$ -points. The Hamiltonian can be put into other scripts that tune the parameters to fit density functional theory (DFT) outputs, add more model terms (e.g. Hubbard U or spin-orbit coupling), calculate properties from the energies, etc. It should be more-or-less plug and play.

---

\* ty.sterling@colorado.edu

First-things-first, I need to actually figure out how to build the TB part of the code. I need matrix elements between orbitals in a unitcell. I can then transform the Hamiltonian to  $\mathbf{k}$ -space, diagonalize to get the energies and eigenvalues, and profit. I want it to work with very little user input: mainly just give positions of atoms, what orbitals are valence on each, and what other atoms they couple to. The code should then fit the energies to DFT outputs, write the parameters, then return a model that can be used willy-nilly. The hard part as of now is calculating the matrix elements for arbitrary angular momentum. I need to figure out how to represent the matrix elements between arbitrary orbitals. It seems like the most reasonable way to implement in a code is to use rotation matrices and rotate orbitals to be quantized along a common z-axis; i.e. the vector that separates them. Then their matrix elements are simpler and have a common form that can be tabulated for all  $Y_{l',m'}^\dagger Y_{l,m}$  pairs. I also need the radial functions, but these will probably be Gaussians and their integrals are easy.

I plan to fit the hopping parameters to DFT calculations: either to band-structures or, somehow, to the potential. Fitting to the potential seems optimal since we don't have to care about entangled bands or band crossings, etc. We just set up and diagonalize the Hamiltonian and get the bands and calculate stuff from them. However, the potential from a DFT code usually includes the core and exchange-correlation (XC) potential which are (generally) not present in TB. Since we are only including the valence bands, we would have to subtract the core and XC from the total DFT potential OR maybe include the core potential in the TB calculation. Alternatively, we just fit the bands which is the usual technique. This is the simplest way: we just define the orbitals we care about, solve the model to get the valence bands, and fit these to the DFT bands. Either way, this code needs to set up the model, return it, and different modules can be used to solve and fit it to different DFT calculations.

A broad outline (*to be updated as I go!*) is as follows (note, this is NOT a table of contents but merely what I plan to put in this document):

1. **Introduction to tight binding:** Introduce the tight-binding method, derive the equations, etc.
2. **tight binding matrix elements:** Derive the tight binding matrix elements for arbitrary angular momentum. We will use the general rotation matrix formalism valid for any angular momentum.
3. **Example?:** I dunno

That is enough rambling for now. Let's get to work!

## II. INTRODUCTION TO TIGHT BINDING

The idea of tight binding (TB) is to have a minimal model that is easy enough to solve and that we can still gain physical intuition from. Originally, Slater and Koster's idea was that a model should be easily solvable by hand. Now that we have modern computers, we really just want a model that can be very quickly solved on a consumer PC. Regarding physical intuition, we don't really care that much about the rigour of our model so much as that the terms in the Hamiltonian are physically transparent (e.g. Hubbard  $U$ ) and that they can be rapidly added to a code and, tuned, and many calculations performed. This is, in principle, possible with DFT. For example, Hubbard- $U$  like corrections to DFT are already available. However, the terms are *not* transparent but are rather black-box like. Similarly implementation is *not* straightforward and running many calculations is *not* cheap.

To keep TB as transparent and cheap as possible, we have to accept it for what it is: a cheap and simple approximation scheme (Slater and Koster called it an 'interpolation' technique, but I don't really care about interpolating band structure). To that end, we ignore all the terms that add complexity (e.g. 3-center integrals) and simply treat the remaining terms as fitting parameters that are tuned to best reproduce some physical properties: e.g. DFT band structures, experimental optical gaps, etc.

We also work in real space (at least, we parameterize our Hamiltonian in real space) so that we can use localized, atomic-like orbitals that we are comfortable with. We usually assume some radial function times a spherical harmonic. This is transparent in the sense that we can easily investigate angular momentum related properties like spin-orbit coupling. Moreover, most correlated calculations focus on highly localized states like  $d$  electrons.

Our starting point is atomic-like orbitals  $\phi_{\{i,l,m\}}(\mathbf{r} - (\mathbf{R}_p + \boldsymbol{\tau}_i))$ .  $\phi$  is an atomic-like orbital which we will take to be the product of a localized radial function  $\chi_l$  and a spherical harmonic  $Y_{l,m}$ . The indices  $l$  and  $m$  denote the orbital angular momentum quantum numbers ( $l(l+1)\hbar^2$  is the eigenvalue of  $\mathbf{L}^2$  and  $m\hbar$  is the eigenvalue of  $L_z$ ).  $i$  labels which atom in the unitcell we are looking at. The curly brace  $\{i,l,m\}$  is to indicate that a particular  $l,m$  orbital is associated with atom  $i$ .  $\mathbf{R}_p$  is a lattice vector with  $p$  indicating the unitcell and  $\boldsymbol{\tau}_i$  is the basis vector of atom  $i$  relative to the unitcell origin.

The Hamiltonian can be written:

$$\hat{H} = \frac{\hbar^2}{2m} \nabla^2 + V(\mathbf{r}) \quad (1)$$

In a crystal, the potential satisfies  $V(\mathbf{r}) = V(\mathbf{r} + \mathbf{R}_p)$  and we know that the solutions of this Hamiltonian are Bloch functions.

### III. FURTHER READING

- *Modern Quantum Mechanics* by Jun Sakurai
  - *Computational Physics* by Jos Thijsen
  - *Berry Phases in Electronic Structure Theory* by David Vanderbilt
  - *Electronic Structure: Basic Theory and Practical Methods* by Richard Martin
-