COMPUTATIONAL TECHNIQUES FOR ACCELERATED MATERIALS DISCOVERY

Franklin T. Cerasoli

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

December 2021

APPROVED:

Marco Buongiorno Nardelli, Major Professor
Oliviero Andreussi, Committee Member
Yuri Rostovtsev, Committee Member
Arrigo Calzolari, Committee Member
Jingbiao Cui, Chair of the Department of Physics
Pamela Padilla, Dean of the College of Science
Victor Prybutok, Dean of the Toulouse Graduate School

Cerasoli, Franklin T. *Computational Techniques for Accelerated Materials Discovery*. Doctor of Philosophy (Physics), December 2021, 185 pp., 1 table, 34 figures, 194 numbered references.

Increasing ubiquity of computational resources has enabled simulation of complex electronic systems and modern materials. The PAOFLOW software package is a tool designed to construct and analyze tight binding Hamiltonians from the solutions of DFT calculations. PAOFLOW leverages localized basis sets to greatly reduce computational costs of post-processing QE simulation results, enabling efficient determination of properties such as electronic density, band structures in the presence of electric or magnetic fields, magnetic or spin circular dichroism, spin-texture, Fermi surfaces, spin or anomalous Hall conductivity (SHC or AHC), electronic transport, and more. PAOFLOW's broad functionality is detailed in this work, and several independent studies where PAOFLOW's capabilities directly enabled research on promising candidates for ferroelectric and spintronic based technologies are described. Today, Quantum computers are at the forefront of computational information science. Materials scientists and quantum chemists can use quantum computers to simulate interacting systems of fermions, without having to perform the iterative methods of classical computing. This dissertation also describes a study where the band structure for silicon is simulated for the first time on quantum hardware and broadens this concept for simulating band structures of generic crystalline structures on quantum machines.

ACKNOWLEDGMENTS

I would like to thank my research supervisor, Dr. Marco Buongiorno Nardelli, who's endless support has afforded me many opportunities for scientific and artistic endeavors. His guidance, patience, and understanding have been steadfast during my experience in this graduate physics program, and his influence on me has extended well beyond the academic setting. In addition, I extend gratitude to the members of my defense committee, Dr. Oliviero Andreussi, Dr. Yuri Rostovtsev, and Dr. Arrigo Calzolari, for their willingness to participate in the production and defense of this dissertation. My coworkers certainly participated in my progress at UNT, and I am especially thankful for the opportunities to work alongside Dr. Jagoda Sławińska, Dr. Ilaria Siloi, Dr. Haihang Wang, Kyle Sherbert, and Anooja Jayaraj. Many interesting ideas, conversations, software implementations, and publications have come from the aforementioned relationships. I would also like to express gratitude to my friends and family for believing in my success and showing interest in my work.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS iii

LIST OF TABLES AND FIGURES vii

CHAPTER 1. INTRODUCTION 1

CHAPTER 2. CONDENSED MATTER THEORY 4

2.1 Theory of Crystalline Structures 4

2.2 The Many Body Problem 5

2.3 Bloch's Theorem 7

2.4 The Density Functional Theory 8

2.5 Pseudopotentials and LCAO 9

CHAPTER 3. PAOFLOW 11

3.1 Introduction 11

3.2 Software Design 13

3.3 Description of the Code 16

3.4 Installation and Input File Structure 24

3.5 Performance 41

3.6 Examples, Testing and Continuous Integration 42

3.7 Externals 44

3.8 Conclusions 47

3.9 Acknowledgments 47

CHAPTER 4. ADVANCED MATERIALS MODELING WITH PAOFLOW 2.0 48

4.1 Introduction 49

4.2 Software Design and Installation 50

4.3 Code Description and Package Usage 52

4.4 Tight Binding Models 75

4.5 Scattering Models 79

4.6 Performance 85

4.7 Conclusion 86
4.8 Acknowledgments 86

CHAPTER 5. GIANT SPIN HALL EFFECT IN TWO-DIMENSIONAL MONOCHALCOGENIDES 87
5.1 Introduction 87
5.2 Methods 89
5.3 Spin Hall Effect in Unstrained Monolayers 91
5.4 Spin Hall Effect in Strained Monochalcogenides 93
5.5 Summary and Conclusions 98
5.6 Acknowledgments 99

CHAPTER 6. ULTRATHIN SNTE FILMS AS A ROUTE TOWARD ALL-IN-ONE SPINTRONICS DEVICES 101
6.1 Introduction 101
6.2 SHC and PSH in 2D SnTe 104
6.3 All-in-One Spin Transistor 107
6.4 Conclusions 113
6.5 Methods 114
6.6 Acknowledgments 115

CHAPTER 7. QUANTUM INFORMATION IN MATERIALS SCIENCE 117
7.1 Quantum Computing 117
7.2 Quantum Computing in Materials Science 120

CHAPTER 8. QUANTUM COMPUTATION OF SILICON ELECTRONIC BAND STRUCTURE 122
8.1 Introduction 122
8.2 Hamiltonian Representation 124
8.3 Variational Quantum Eigensolver 126
8.4 Energies Beyond the Ground State 127
8.5 Data Acquisition 128
8.6 Quantum Experiments 129
8.7 Additional Remarks on Measuring Expectation Values 133
8.8 Error Analysis and Mitigation 135
8.9 Summary and Perspectives 138

8.10 Acknowledgements 139

CHAPTER 9. A SYSTEMATIC VARIATIONAL APPROACH TO BAND THEORY IN A QUANTUM COMPUTER 140

9.1 Introduction 140

9.2 Background 143

9.3 Method 146

9.4 Results 157

9.5 Discussion 164

9.6 Conclusion 167

9.7 Acknowledgements 167

CHAPTER 10. CONCLUSION 169

REFERENCES 171

LIST OF TABLES AND FIGURES

Page

Tables

Table 4.1: Symbol, units, and corresponding key in tau_dict for the parameters required in various scattering models. 84

Figures

Figure 3.1: Comparison between fixed width and adaptive smearing for the density of state of FCC Si. 14

Figure 3.2: Top panel: Band structure of Fe including spin-orbit interaction. Bottom panel: Berry curvature 14

Figure 3.3: Energy resolved anomalous Hall conductivity in Fe 14

Figure 3.4: Magnetic circular dichroism spectrum of Fe 15

Figure 3.5: Energy resolved spin Hall conductivity in Pt 15

Figure 3.6: Spin circular dichroism spectrum of Pt 15

Figure 3.7: Schematic illustration of the zero padding algorithm for fast FFT interpolation 20

Figure 3.8: Parallelized routines provide run time scaling proportional to the number of cores used in a calculation, closely approaching the speed increase limit of Amdahl's Law. 41

Figure 3.9: Executing Fast Fourier Transforms in parallel on GPUs yields a significant increase in run time performance on calculations with a large MP mesh. 42

Figure 3.10: Si(001) surface projected band structure 46

Figure 4.1: Fermi surface of FeP calculated on a ultra-dense k-grid in PAOFLOW and visualized in FermiSurfer [49] 64

Figure 4.2: Spin texture ($\boldsymbol{Sz}$) of two-dimensional ferroelectric SnTe along high-symmetry lines (`example08`) (see chapter 6). 67

Figure 4.3: Electronic density for diamond structure of silicon on the $\langle \mathbf{1}, \mathbf{0}, -\mathbf{1} \rangle$ plane cut, calculated from the real space PAO wavefunctions (`example01`). 71

Figure 4.4: Select examples (see `examples/` on GitHub) performed on an increasing number of processors. 85

Figure 4.5: Memory scaling per core (in GB), for select examples 86

Figure 5.1: Structure of 2D group IV monochalcogenides 91

Figure 5.2: Relativistic electronic structures of group IV monochalcogenides GeX (a) and SnX (b), X = S, Se 94

Figure 5.3: Electronic properties and spin Hall conductivities of SnTe as a function of strain 97

Figure 5.4: Spin Hall conductivity for SnTe as a function of strain and doping 98

Figure 6.1: Structural and electronic properties of SnTe monolayer 106

Figure 6.2: Properties and applications of biased 1ML-SnTe 108

Figure 6.3: Geometry and electronic structure of the 2ML-SnTe 110

Figure 6.4: Properties of multilayer SnTe 112

Figure 8.1: Mean-field circuit acting on a single qubit has been employed to determine the lowest bands of silicon. 130

Figure 8.2: The two-band electronic structure of silicon computed along X-Γ-L line 131

Figure 8.3: The circuit used to diagonalize the 8x8 Hamiltonian 132

Figure 8.4: Electronic structure of silicon computed via hybrid classical/quantum algorithm on QSS 132

Figure 8.5: The transition rates estimated for a qubit on Rigetti's QPU 137

Figure 9.1: (a) The ansatz $\boldsymbol{V(\theta)}$ suitable for any band structure calculation. (b) One constituent $\boldsymbol{A(\theta, \phi)}$ gate 152

Figure 9.2: A schematic of our algorithm and its relation to VQE 154

Figure 9.3: Statevector Simulator – The band structure of a simple cubic lattice with $\boldsymbol{s}$ and $\boldsymbol{p}$ orbitals (right inset) along the high-symmetry path XMΓ through the lattice's First Brillouin Zone (left inset) 158

Figure 9.4: Sampling Simulator – Our method applied in the presence of sampling noise (high-fidelity qubits). 160

Figure 9.5: Noisy Simulator – Our method applied while simulating low-fidelity qubits, without calibration. 161

Figure 9.6: Calibrated Simulator – Our method applied while simulating low-fidelity qubits, along with rudimentary calibration 163

CHAPTER 1

INTRODUCTION

Modern condensed matter physics was birthed from the unification of several independent investigations into the atomic properties of matter. J.J. Thompson's discovery of the electron led to Drude developing a model for electrons in metallic solids [1,2]. The experimental discovery of superconductivity in 1911 by Heike Onnes incited heavy research/curiosity and wouldn't be adequately explained until forty years later, when the quantum mechanics formalism was more widely adopted [3]. With Schrödinger's wave interpretation of quantum mechanics, Feliks Bloch established the dynamics of electrons within periodic systems (1928), fueling development of modern solid-state physics. Researchers raced to explain many new observed phenomena, such as the quantum Hall effect or the BCS theory of superconductors [4,5], and developed curious new theories to describe them.

Increasing ubiquity of computational resources in the 1950s enabled simulation of electronic systems which were too complex to perform by hand. The Hartree-Fock method became a feasible tool for computing electronic band structures and was widely adopted, ultimately inspiring the development of density functional theory (DFT) in 1964 by Hohenberg, Kohn, and Sham [6,7]. Computing devices and the DFT framework provided thrust to the computational materials science field, which promised to identify new properties of matter and design better materials for widespread applications. Large scale, production codes for performing DFT simulations gained traction over the following years, such as Quantum ESPRESSO (QE) [8,9]. Software packages like QE are capable of capturing electronic behavior in

complex condensed matter systems and have become commonplace in materials science and quantum chemistry research. Chapter 2 introduces formalism of condensed matter theory, such as the mathematical construction of a crystal, Bloch's theorem for an independent electron in a periodic lattice, and solving many body problems with the density functional theory and Quantum ESPRESSO.

Much of this work is devoted to a software package called PAOFLOW, a tool designed to construct and analyze tight binding Hamiltonians from the solutions of DFT calculations. PAOFLOW leverages localized basis sets to greatly reduce computational costs of post-processing QE simulation results, enabling efficient determination of properties such as electronic density, band structures in the presence of electric or magnetic fields, magnetic or spin circular dichroism, spin-texture, Fermi surfaces, spin or anomalous Hall conductivity (SHC or AHC), electronic transport, and more. Chapters 3 and 4 describe the PAOFLOW software package and underlying theory for the code's broad functionality. Additionally, chapter 4 provides an explicit guide to using the software package and calling internal routines for constructing and operating on tight-binding Hamiltonians. Several applications for PAOFLOW are described in chapters 3 and 4, but chapters 5 and 6 provide independent studies where PAOFLOW's capabilities directly enabled research on promising candidates for ferroelectric and spintronic based technologies. Chapters 5 overviews one specific application of PAOFLOW, where a high-throughput study was performed on members of the group-IV monochalcogenide family. Chapter 6 provides a detailed study where a spintronic transistor is proposed with two dimensional SnTe as a component.

Today, a new paradigm of computing promises to outperform classical computers by an exponential factor for certain problems and physical simulations. Quantum computers are now at the forefront of computational information science. Materials scientists and quantum chemists can use quantum computers to simulate interacting systems of fermions, without having to perform the iterative methods of classical computing. Some crypto scientists study methods of factoring large numbers on quantum computers, which could provide an exponential speedup to modern decryption algorithms [10]. Biological scientists have proposed that determining how proteins fold may be easier on quantum computers [11], and pharmacologists may use quantum computers in the future to accelerate discovery of compounds used as medication [12]. Chapter 7 provides a brief introduction to quantum computing and quantum information science in the context of chemistry and materials science. Chapter 8 describes a recent study where the band structure for silicon is simulated for the first time on quantum hardware, and Chapter 9 broadens this study for simulating band structures to generic crystalline structures on quantum machines.

CHAPTER 2

CONDENSED MATTER THEORY

2.1 Theory of Crystalline Structures

Crystalline structures are naturally periodic, possessing long range uniformity. They are formed by atoms arranged in a spatially repeating configuration. The smallest volume which contains all features of a crystal's pattern is known as the primitive unit cell. Generally, two quantities are considered when creating a mathematical description of such crystalline structures, thae Bravais lattice and the atomic basis. The Bravais Lattice (BL) is all points defined by a set of vectors:

$$\vec{R} = a\vec{r}_1 + b\vec{r}_2 + c\vec{r}_3 \tag{2.1}$$

$\vec{R}$ is specified by a geometry of three primitive lattice vectors $\{\vec{r}_1, \vec{r}_2, \vec{r}_3\}$ and three integers $(a, b, c) \in \mathbb{Z}$. The resulting structure is infinite in extent and appears identical when viewed from any of its points. A primitive unit cell has the volume of a parallelepiped constructed with the primitive lattice vectors.

$$\Omega = |\vec{r}_1 \cdot (\vec{r}_2 \times \vec{r}_3)| \tag{2.2}$$

Every BL possesses a dual representation, namely the reciprocal lattice, which is constructed as the Fourier transform of the real space primitive lattice vectors. The reciprocal lattice is generally notated as

$$\vec{G} = k\vec{b}_1 + l\vec{b}_2 + m\vec{b}_3 \tag{2.3}$$

where $\{\vec{b}_1, \vec{b}_2, \vec{b}_3\}$ are called the reciprocal lattice vectors and $(k, l, m) \in \mathbb{Z}$. The lattice periodicity enforces that $\vec{r}_i \cdot \vec{\mathrm{b}}_\mathrm{j} = 2\pi\delta_{ij}$, and the reciprocal lattice vectors are related to the primitive vectors by the following relations.

$$\vec{b}_1 = \frac{2\pi}{\Omega}\vec{\mathrm{r}}_2 \times \vec{\mathrm{r}}_3 \tag{2.4a}$$

$$\vec{b}_2 = \frac{2\pi}{\Omega}\vec{r}_3 \times \vec{r}_1 \tag{2.4b}$$

$$\vec{b}_3 = \frac{2\pi}{\Omega}\vec{r}_1 \times \vec{r}_2 \tag{2.4c}$$

The unit cell constructed in reciprocal space is known as the Wigner-Seitz cell, and is often referred to as the First Brillouin Zone (FBZ or BZ) in solid state physics. Naturally, this dual corresponds to a momentum space of the conventional lattice and is used to study electronic behavior in crystals.

Another set of vectors, the atomic basis, specifies atomic positions within the primitive unit cell. This basis for a lattice containing N atoms is denoted $\tau = \{\vec{\tau}_1, \vec{\tau}_2, \dots, \vec{\tau}_N\}$. Together, the Bravais lattice and atomic basis define a periodic system of atoms in all of space, capable of approximating the macroscopic configuration of natural crystalline structures.

2.2 The Many Body Problem

The first order of business when modeling a system comprised of quantum mechanical particles is to solve an appropriate Schrödinger equation.

$$\widehat{H}|\psi\rangle = E|\psi\rangle \tag{2.5}$$

Here, $\widehat{H}$ is some Hermitian operator representing the systems total energy, E is a diagonalized matrix containing the energy eigenvalues obtainable by said system, and $|\psi\rangle$ is a vector of wavefunctions which solve the Schrödinger equation. To write a trivial Hamiltonian for a solid state system, we include kinetic energy terms for all particles (N_e electrons at positions $\vec{r}_i$ and N_n nucleons at positions $\vec{R}_j$, for $i = 1,2, \dots, N_e$ and $j = 1,2, \dots, N_n$) and Coulomb potential

terms for all charged particles (each electron with fundamental charge e and each nucleon with total charge $Z_i e$ for $i = 1,2, \dots, N_n$).

$$\widehat{H} = -\sum_{i}^{N_e} \frac{\hbar}{2m_e} \nabla_i^2 - \sum_{i}^{N_n} \frac{\hbar}{2m_n} \nabla_i^2 + \sum_{i}^{N_e} \sum_{j>i}^{N_e} \frac{e^2}{4\pi\epsilon_0} \frac{1}{\left|\vec{r}_i - \vec{r}_j\right|} + \frac{1}{2} \sum_{i}^{N_e} \sum_{j}^{N_n} \frac{e^2}{4\pi\epsilon_0} \frac{Z_j}{\left|\vec{r}_i - \vec{R}_j\right|} + \sum_{i}^{N_n} \sum_{j>i}^{N_n} \frac{e^2}{4\pi\epsilon_0} \frac{Z_i Z_j}{\left|\vec{R}_i - \vec{R}_j\right|} \tag{2.6}$$

Some simplifications are employed to reduce the number of elements in the above Hamiltonian. Perhaps the most obvious is that the nucleons are essentially stationary with respect to their surrounding electrons. The electrons move adiabatically with the nucleons, because of the proton to electron mass relation ${m_p}/{m_e} \approx 2000$. The kinetic energy contribution of nucleons in the Hamiltonian vanishes, i.e. $\sum_i^{N_n} \frac{\hbar}{2m_n} \nabla_i^2 = 0$. Subsequently, the electron-nucleon Coulomb term is treated as each electron independently interacting with a global potential formed by the nucleons and sharing periodicity with the unit cell.

$$\frac{1}{2} \sum_{i}^{N_e} \sum_{j}^{N_n} \frac{e^2}{4\pi\epsilon_0} \frac{Z_j}{\left|\vec{r}_i - \vec{R}_j\right|} + \sum_{i}^{N_n} \sum_{j>i}^{N_n} \frac{e^2}{4\pi\epsilon_0} \frac{Z_i Z_j}{\left|\vec{R}_i - \vec{R}_j\right|} \approx \sum_{i}^{N_e} \sum_{j}^{N_n} V\left(\left|\vec{r}_i - \vec{R}_j\right|\right) \tag{2.7}$$

This is known as the Born-Oppenheimer approximation, first proposed in the 1927 paper of Max Born and Robert Oppenheimer [13]. After reducing terms pertaining to atomic nuclei, notice that writing this Hamiltonian still leads to N_e^2 elements in the term describing the purely electronic Coulomb repulsion, which is clearly unfavorable scaling as the number of electrons increases. To achieve linear scaling, an effective potential $V_e(\vec{r}_i)$ for electron i at is

produced by the average potential contributions from all other electrons, and the Hamiltonian for each electron takes a familiar form dependent only on position $\vec{r}$ within the lattice:

$$\hat{H}(\vec{r}) = -\frac{\hbar}{2m_e}\nabla^2 + V(\vec{r}) \tag{2.8}$$

with

$$V(\vec{r}) = V_e(\vec{r}) + \sum_{j}^{N_n} V\left(\left|\vec{r} - \vec{R}_j\right|\right) \tag{2.9}$$

While the independent electron approximation alleviates the computational load of modeling an electron gas in the presence of ions and accurately describes many properties of solids, it fails to capture any phenomenon arising from correlations between electrons. Extensions to the theory, such as exchange functionals [14], can help to model a wider range of processes in chemicals and materials while still maintaining the linear scaling with number of electrons achieved by the independent electron approximation.

2.3 Bloch's Theorem

The independent electron approximation is employed to avoid the computational complexity of a many-body system of electrons. As such, a potential is introduced which accumulates interaction effects from all other electronic charges in the problem into a single potential term $V(\vec{r})$ for the electron at position $\vec{r}$. Under this single electron approximation, properties of the BL are exploited to simplify the form that electronic states can take. Namely, Bloch's theorem shows that electronic wavefunctions can be written as an easily computed phase factor $e^{i\vec{k}\cdot\vec{r}}$ times a function $u(\vec{k},\vec{r})$ sharing periodicity with the BL.

$$\psi(\vec{k},\vec{r}) = u(\vec{k},\vec{r})e^{i\vec{k}\cdot\vec{r}} \tag{2.10}$$

$$u(\vec{k}, \vec{r} + N_i\vec{r}_i) = u(\vec{k}, \vec{r}) \tag{2.11}$$

Adding to the unit cell position $\vec{r}$ any integer multiple N_i of lattice vectors $\vec{r}_i$ results in the same value of the periodic function $u(\vec{k}, \vec{r})$. Thus, the wavefunction for any spatial position in a lattice can be written in terms of the wavefunction within the primitive unit cell.

$$\psi(\vec{k}, \vec{r} + \vec{R}) = \psi(\vec{k}, \vec{r})e^{i\vec{k}\cdot\vec{R}} \tag{2.12}$$

Translational symmetry of the BL requires that, for integers N_i and lattice vectors $\vec{r}_i$ with $i = 1,2,3$,

$$\psi(\vec{k}, \vec{r} + N_i\vec{r}_i) = \psi(\vec{k}, \vec{r}) \tag{2.13}$$

This boundary condition restricts the values of $\vec{k}$ to

$$\vec{k} = \frac{k}{N_1}\vec{b}_1 + \frac{l}{N_2}\vec{b}_2 + \frac{m}{N_3}\vec{b}_3 \tag{2.13}$$

where $(k, l, m) \in \mathbb{Z}$. The wave vector $\vec{k}$ describes the electronic momentum in the crystal, usually called crystal momentum and given by $\vec{p}_{crystal} = \hbar\vec{k}$.

The following section describes the density functional theory, which relies on both the independent electron approximation from section 2.1 and the Bloch's theorem in section 2.2 for computational efficiency.

2.4 The Density Functional Theory

The pioneering logical assumptions of DFT were formulated in attempts to characterize the properties and behavior of an inhomogeneous electron gas. Hohenberg and Kohn introduce a functional to describe the total energy E in a system of interacting electrons, dependent upon the spatial charge density $n(\vec{r})$ [6,7]. They demonstrate that minimizing this functional provides the correct ground state electronic distribution. With the ground state charge distribution one

can write and solve a Schrödinger equation, similar to equation 2.8. The solutions to this equation are known to the condensed matter community as the Kohn-Sham (KS) wavefunctions. For more nuanced information on DFT methodology and application, the reader is encouraged to see reference [15].

To attack the DFT problem for some system of interacting electrons, one must choose a basis set suitable for capturing important features. Quantum ESPRESSO (QE) is a software suite for materials modeling which solves the DFT problem in the basis of plane waves (PW) [8]. The QE executable `pw.x` performs self-consistent calculations to converge on a ground state energy for a configuration of atoms, following the theory of Hohenberg, Kohn, and Sham. For the purpose of studying higher energy conduction bands, non self-consistent calculations can also be performed by `pw.x` to extend the number of bands in a calculation which has already been converged self-consistently. QE contains dozens of modules for various materials properties, but in this work the focus is on two routines: `pw.x` for self-consistent and non self-consistent DFT calculations, and `projwfc.x` for computing projections (described in section 3.1) if they are not computed directly with PAOFLOW.

2.5 Pseudopotentials and LCAO

A problem is encountered when attempting to implement DFT in the PW basis. The repulsive coulomb potential, scaling with $1/|\vec{r}|$, diverges close to core electron sites. This extreme localization of core electrons produces an rapidly oscillating feature in the KS wavefunctions near atomic sites. The number of PWs required to achieve accurate energies for valence states becomes intractably large. Valence electrons generally predict electronic/optical

properties in solids, and core electrons cause logistical problems for simulation grids. This motivates the removal of core states from the Hamiltonian. To achieve this, a Hamiltonian is built relative to a set of Orthogonal Plane Waves (OPW) for the valent electrons, orthogonal to those of the core electrons [16]. The new, orthogonal plane wave basis $|\Psi_{OPW}\rangle$ is given by

$$|\Psi_{OPW}(\vec{k})\rangle = |\Psi_{PW}(\vec{k})\rangle - \sum_{c} |\psi_c\rangle\langle\psi_c|\Psi_{PW}(\vec{k})\rangle \tag{2.14}$$

where $|\Psi_{PW}\rangle$ is the complete PW basis, and the summation index c runs across all core states. The lowest energy states in the new basis then receive no contribution from core states and are necessarily those of the most tightly bound valence electrons.

Pseudopotentials compatible with QE are stored in the Universal Pseudopotential Format (UPF), and generally contain several radial functions for various valence shell configurations. For example, a pseudopotential for silicon might contain radial functions for the 3S, 3P, and 3D valence shells, a common configuration for this element. An atomic orbital basis is constructed for these shells by weighting the radial functions with spherical harmonics for each allowed angular momenta. The 3D ($L = 2$) radial wave function would have five corresponding atomic orbital functions, one for each angular momentum $m_l = -2, -1, 0, 1, 2$.

Often in materials science and quantum chemistry, an incomplete basis of these atomic orbitals is used to capture the electronic properties of solids and molecules. Naturally, this method is known as the linear combination of atomic orbitals (LCAO). PAOFLOW, described in the following chapter, takes the plane wave solutions of QE and projects them onto a basis of atomic orbitals to drastically reduce the computational resources required for determination of further quantities.

CHAPTER 3

PAOFLOW*

PAOFLOW is a utility for the analysis and characterization of materials properties from the output of electronic structure calculation. By exploring an efficient procedure to project the full plane-wave solution on a reduced space of atomic orbitals, PAOFLOW facilitates the calculation of a plethora of quantities such as diffusive, anomalous and spin Hall conductivities, magnetic and spin circular dichroism, and Z_2 topological invariants and more. The computational cost associated with post-processing first principles calculations is negligible. This code, written entirely in Python under GPL 3.0 or later, opens the way to the high-throughput computational characterization of materials at an unprecedented scale. It is important to note here, that Chapter 3 focuses on version 1.0 of PAOFLOW. While the mathematical description of routines described in this chapter remain accurate in modern versions of PAOFLOW, the current software implementation here is outdated. The most recent software design and package usage is presented in Chapter 4, for PAOFLOW 2.0.

3.1 Introduction

PAOFLOW is a software tool to efficiently post-process standard first principles electronic structure plane-wave pseudopotential calculations in order to promptly compute from interpolated band structures and density of states several quantities that provide insight on transport, optical, magnetic, and topological properties such as anomalous and spin Hall

* This chapter is reproduced with permission from Elsevier, from the paper: *PAOFLOW: A utility to construct and operate on ab initio Hamiltonians from the projections of electronic wave functions on atomic orbital bases, including characterization of topological materials* by Marco Buongiorno Nardelli, Frank T. Cerasoli, Marcio Costa, Stefano Curtarolo, Riccardo De Gennaro, Marco Fornari, Laalitha Liyange, Andrew R. Supka, and Haihang Wang, published in Computational Materials Science 143, 462 (2018).

conductivity (AHC and SHC, respectively), magnetic circular dichroism, spin circular dichroism, and topological invariants. The methodology is based on the projection on pseudo-atomic orbitals (PAO) [17-19] and is the latest addition to the AFLOW software infrastructure [20,21]. Additional features of PAOFLOW include the calculation of selected integrated quantities using adaptive smearing, the ability to add spin orbit coupling using adaptive smearing, the ability to add spin orbit coupling using parametrized methods, and the calculation of surface projected band structures.

PAOFLOW is massively parallel by design (both CPU and GPU) and provides the user with the ability to determine measurable quantities with first principles accuracy and with the spend and robustness required by high-throughput materials characterization. The current implementation (using Quantum ESPRESSO, QE) [8,9] does not require any additional input with respect to standard electronic structure calculations and, seamlessly, provides a real space tight-binding (TB) representation of the Hamiltonian matrix in a self-contained XML format. The sparse PAO matrix can easily be Fourier transformed and interpolated to determine the full energy dispersion and to compute additional properties associated with derivatives of the energy bands, such as matrix elements of the momentum operator or electron velocities, with the desired level of resolution. PAOFLWO is publicly available under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version. PAOFLOW is integrated into the AFLOWπ high-throughput framework [22] and it is distributed at http://www.aflow.org/src/aflowpi and http://www.aflow.org/src/paoflow.

3.2 Software Design

PAOFLOW is written in Python 2.7 (using the Python standard libraries `NumPy` and `SciPy`). The systematic use of regular expression (`re` module) and XML parsing (`xml.etree.cElementTree` module) makes the software expandable to a variety of electronic structure engines with minimal effort. Parallelization on CPUs uses the OpenMPI protocol through the `mpi4py` module, while GPU parallelization is based on the CUDA kernel through the `pycuda` and `skcuda` [23] modules (https://mathema.tician.de/software/pycuda/).

Currently, PAOFLOW requires a few basic calculations with the QE package: a first one to generate converged electronic density and Kohn-Sham (KS) potential on an appropriate Monkhorst and Pack (MP) $\vec{k}$-point mesh (`pw.x`), a second non self consistent calculation (`pw.x`) to evaluate eigenvalues and eigenfunctions for a MP mesh centered at Γ ($\vec{k} = (0,0,0)$, `nosym` and `noinv=.true`) and a third post-processing run using `projwfc.x` to obtain the projections of the eigenfunctions on the pseudo atomic basis functions. No additional calculations with QE are required.

Starting with highly interpolated first principles electronic properties (figure 3.1), PAOFLOW computes band derivatives and Berry's curvature (figure 3.2). These ingredients are then used to determine efficiently the AHC (figure 3.3), magnetic circular dichroism spectra (figure 3.4), SHC (figure 3.5), and spin circular dichroism (figure 3.6).

The PAOFLOW package is distributed with several examples (in the main directory of the distribution, see section 3.6) describing the computable physical quantities and can be easily installed on any hardware.

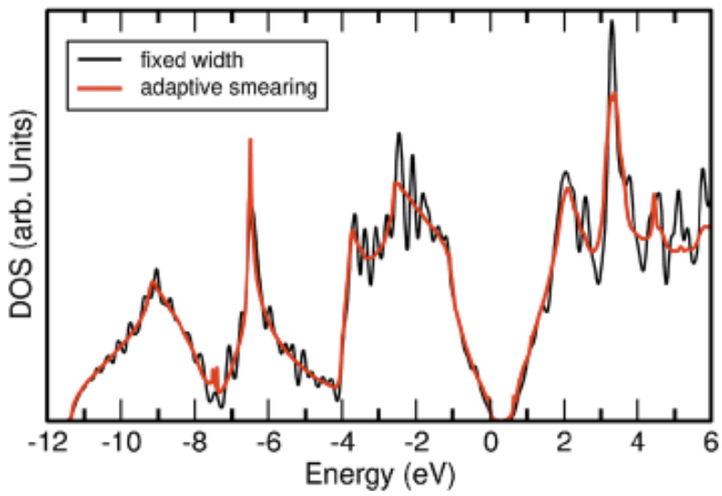


Figure 3.1: Comparison between fixed width and adaptive smearing for the density of state of FCC Si. The calculation is done on a $18 \times 18 \times 18$ MP grid and with $W = 0.1$ eV in the fixed width smearing algorithm. Data adapted from `example01`.

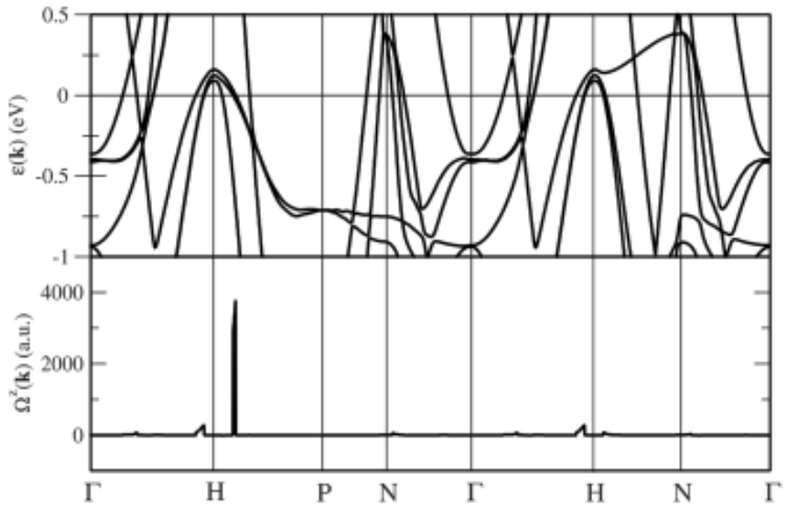


Figure 3.2: Top panel: Band structure of Fe including spin-orbit interaction. Bottom panel: Berry curvature. Data adapted from `example04` and evaluated along the AFLOW standard path for the BCC lattice [24].

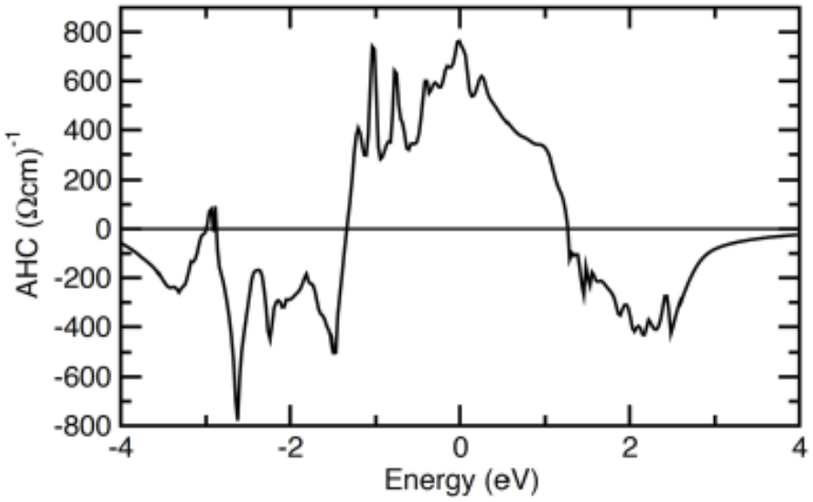


Figure 3.3: Energy resolved anomalous Hall conductivity in Fe. Data adapted from `example04` using a $42 \times 42 \times 42$ MP grid and adaptive broadening yielding a converged value for the AHC at E_{Fermi} of $751\ (\Omega cm)^{-1}$ in excellent agreement with the results from reference [25].

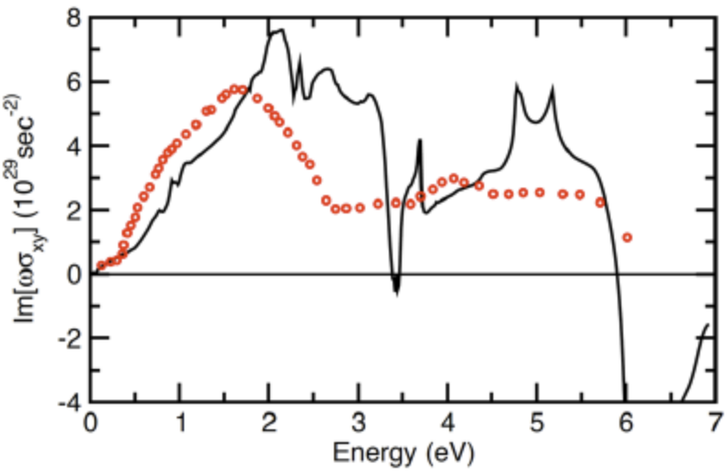


Figure 3.4: Magnetic circular dichroism spectrum of Fe. Red circles are experimental data from reference [26]. Data adapted from `example04` using a $42 \times 42 \times 42$ MP grid and adaptive broadening.

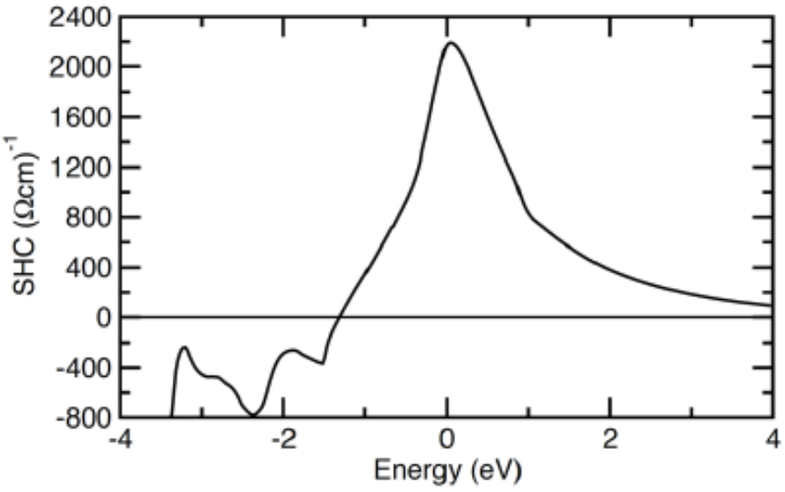


Figure 3.5: Energy resolved spin Hall conductivity in Pt. Data adapted from `example05` using a $42 \times 42 \times 42$ MP grid and adaptive broadening yielding a converged value for the SHC at E_{Fermi} of 2170 $(\Omega cm)^{-1}$ in excellent agreement with previous experimental [27] and theoretical [28] results.

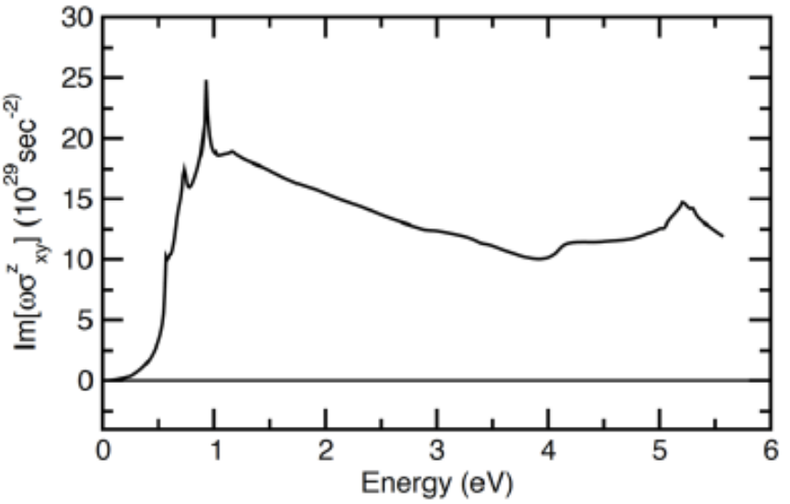


Figure 3.6: Spin circular dichroism spectrum of Pt. Data adapted from `example04` using a $42 \times 42 \times 42$ MP grid and adaptive broadening.

3.3 Description of the Code

3.3.1 Modules: `build_Pn.py, build_Hks.py`

Accurate PAO Hamiltonian matrices can be built from the direct projection of the KS Bloch states $|\psi_{n\vec{k}}\rangle$ onto a basis set of fixed localized functions, as it was discussed extensively in reference [17-19]. The Hamiltonian for a specific material, $\hat{H}(\vec{r}_\alpha)$, is computed in real space using atomic orbitals or pseudo atomic orbitals from the pseudopotential of any given element. The key in this procedure is the mapping of the *ab initio* electronic structure (solved on a well converged and large plane waves basis set) into a model that precisely reproduces a selected number of bands of interest. The crucial quantities that measure the accuracy of the basis set are the projectabilities $p_{n\vec{k}} = \langle\psi_{n\vec{k}}|\hat{P}|\psi_{n\vec{k}}\rangle \geq 0$ ($\hat{P}$ is the operator that projects onto the space of the PAO basis set, as defined in reference [18]) which indicate the representability of a Bloch state $|\psi_{n\vec{k}}\rangle$ on the chosen PAO set. Maximum projectability, $p_{n\vec{k}} \approx 1$, indicates that the particular Bloch state can be perfectly represented in the chosen PAO set; contrarily, $p_{n\vec{k}} \approx 0$ indicates that the PAO set is insufficient and should be augmented. Once the Bloch states with good projectabilities have been identified, the PAO Hamiltonian is constructed either as

$$\hat{H} = AEA^\dagger + \kappa(I - AA^\dagger) \tag{3.1}$$

following reference [17] or

$$\hat{H} = AEA^\dagger + \kappa(\mathrm{I} - \mathrm{A}(\mathrm{A}^\dagger A)^{-1}\mathrm{A}^\dagger) \tag{3.2}$$

as in reference [18], where the case can be chosen in the input of PAOFLOW (see listing 3.2 below). Here E is the diagonal matrix of KS eigenenergies and A is the matrix of coefficients obtained from projecting the Bloch wavefunctions onto the PAO set. Since the filtering procedure introduces a null space, the parameter κ is used to shift all the unphysical solutions

outside a given energy range of interest. The procedure in equation 3.2 is recommended for most cases. This procedure provides and accurate real space representation of the *ab initio* Hamiltonian $\widehat{H}(\vec{R})$ as a TB matrix of small dimension written in XML format, a crucial advantage for the accurate calculation of any physical properties that requires the precise integration in the reciprocal space.

3.3.2 Module: `add_ext_field.py`

Following the TB formalism (see, for instance, reference [29]), the user can add an arbitrary external electric field as a (time-dependent) scalar potential acting on the diagonal elements of the PAO Hamiltonian: $\epsilon_{\alpha,\vec{R}} = \epsilon^{0}_{\alpha,\vec{R}} - e\Phi(\vec{R},T)$. From here is straightforward to introduce an *ad hoc* Hubbard U correction in the Hamiltonian including defining a different U value for any orbital manifold on any element and obtain a fast evaluation of the effect of, for instance, self-consistently computed corrections on the band strcutre, *i.e.* the Agapito-Curtarolo-Buongiorno-Nardelli approach (ACBN0) [30,31]. This feature is documented in section 3.4.3.

3.3.3 Module: `do_spin_orbit.py`

The spin-orbit coupling (SOC) is essential to characterize the electronic structure and associated properties in topological phases (*e.g.* topological insulators, Weyl semimetals) and materials with magneto-crystalline anisotropy. In addition, the SOC modify the band strcutre and, in turn, affects transport and optical properties. From a fully relativistic QE calculation PAOFLOW can compute a TB Hamiltonian, $\widehat{H}(\vec{R})$, which includes all spin-orbit interactions. QE COS calculations are, however, computationally demanding and require relativistic

pseudopotentials. PAOFLOW provides an alternative approach to include the effects of the SOC *via* an effective approximation proposed by Abate and Asdente [32]. The SOC contribution to the Hamiltonians is usually written as:

$$H_{SOC} = \lambda \vec{L} \cdot \vec{S} \tag{3.3}$$

where λ is the SOC strength, which is orbital dependent, $\vec{L}$ and $\vec{S}$ are the orbital and spin angular momentum operators. The λ parameter can be added to non-SOC calculations semi-empirically and adjusted to reproduce the SOC splitting at any band of the element's stable phase. To ensure accuracy of the results, it is important to choose bands with reasonably high projectabilities.

3.3.4 Modules: `do_bands_calc.py`, `do_topology_calc.py`, `do_fermisurf.py`, `do_spin_texture.py`, `do_dos_calc.py`, `do_pdos_calc.py`

The calculation of topological invariants and related measurable quantities is important to analyze and validate predictions of topological effects in electronic structure theory. PAOFLOW facilitates the automatic computation of standard descriptors of the topology of band strcutre such as Fermi surfaces, spin texture, band velocities ($\partial\epsilon/\,\partial k_i, i = x, y, z$), Berry curvature, spin Berry curvature and the Z_2 invariant for topological insulators. In particular, we follow reference [33] and calculate the four Z_2 invariants v_0; (v_1, v_2, v_3) defined by

$$(-1)^{v_0} = \prod_{n_j=0,1} \delta_{n_1 n_2 n_3}$$

$$(-1)^{v_{i=1,2,3}} = \prod_{n_{j\neq i}=0,1; n_1=1} \delta_{n_1 n_2 n_3} \tag{3.4}$$

where

$$\delta_i = \frac{\sqrt{det[w(\Gamma_i)]}}{pf[w(\Gamma_i)]} = \pm 1 \tag{3.5}$$

Here $pf[w(\Gamma_i)]$ indicates the Pfaffian, $w(\vec{k}) = \langle \psi_i(-\vec{k})|\Theta|\psi_j(\vec{k})\rangle$ with Θ the time-reversal operator, and Γ_i the 8 distinct time reversal invariant momenta expressed in terms of primitive lattice vectors as $\Gamma_{i=(n_1 n_2 n_3)} = (n_1\vec{b}_1 + n_2\vec{b}_2 + n_3\vec{b}_3)/2$ [33]. Of course, all the properties defined in the first Brillouin zone (BZ) including the band structure, are calculated along the standard AFLOW path (see for instance figure 3.2) [24] using Fourier interpolation scheme.

3.3.5 Modules: `PAOFLOW.py, smearing.py, do_gradient.py, do_momenta.py`

Band structure interpolation on arbitrary MP $\vec{k}$-meshes as well as adaptive smearing for the integration in the BZ are at the very core of the ability of PAOFLOW to provide high-precision electronic structure data. Exploiting the analogy between the PAO and the Wannier functions representation of *ab initio* Hamiltonians, PAOFLOW implements a procedure developed in reference [34]. In practice, we estimate the broadening widths using the band derivatives that are readily available from the knowledge of the momentum operator and can be used to estimate the level spacing. Indeed, the TB Hamiltonian can be Fourier transformed from real space representation to the $\vec{k}$-space and interpolated with arbitrary precision using an efficient procedure based on a zero-padding algorithm that operates globally and fast Fourier transform (FFT) routines (see figure 3.7 for an illustration of the method).

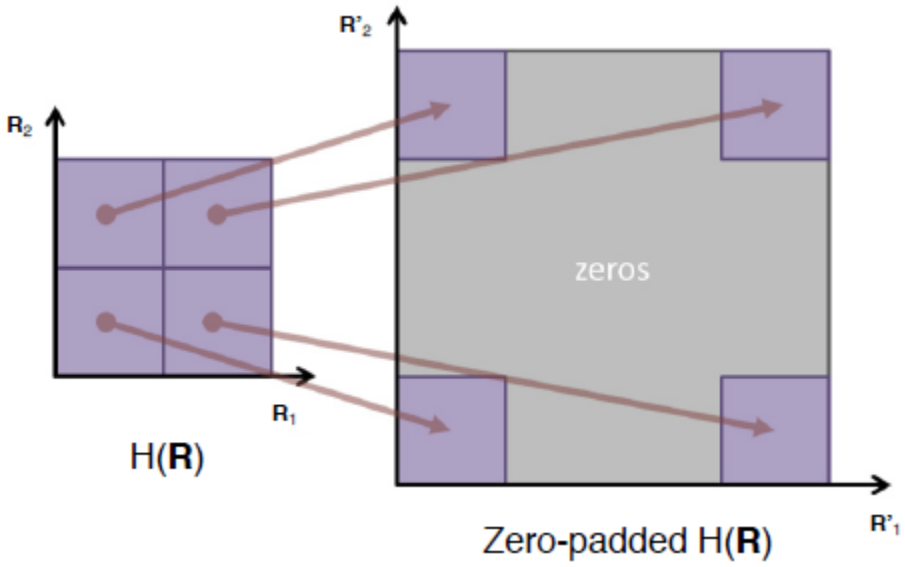


Figure 3.7: Schematic illustration of the zero padding algorithm for fast FFT interpolation. Picture adapted from https://tonyfast.com/nsf-goali/.

The same accuracy defined by the projectabilities is conserved in this process. The expectation values of the momentum operator, which is the main quantity in the definition of the adaptive smearing integration scheme, is given by

$$\vec{p}_{nm}(\vec{k}) = \langle \psi_n(\vec{k}) | \hat{p} | \psi_m(\vec{k}) \rangle = \langle u_n(\vec{k}) \left| \frac{m_0}{\hbar} \vec{\nabla}_{\vec{k}} \hat{H}(\vec{k}) \right| u_m(\vec{k}) \rangle \tag{3.6}$$

with

$$\vec{\nabla}_{\vec{k}} \hat{H}(\vec{k}) = \sum_{\alpha} i\vec{R}\, exp(i\vec{k} \cdot \vec{R}) \hat{H}(\vec{R}) \tag{3.7}$$

$\hat{H}(\vec{R})$ being the real space PAO matrix and $|\psi_n(\vec{k})\rangle = exp(-i\vec{k} \cdot \vec{R}) |u_n(\vec{k})\rangle$ the Bloch's functions [35].

Following reference [34] we define:

$$W_{n,\vec{k}} = a \left| \frac{\partial \epsilon_{n,\vec{k}}}{\partial k} \right| \tag{3.8}$$

for single band integrals and

$$W_{n,m,\vec{k}} = a \left| \frac{\partial \epsilon_{n,\vec{k}}}{\partial k} - \frac{\partial \epsilon_{m,\vec{k}}}{\partial k} \right| \tag{3.9}$$

for double band integrals. The factor a is of the order of one [22]. See figure 3.1 for a comparison between fixed *vs.* adaptive smearing in the calculation of the density of states of silicon.

3.3.6 Module: `do_Boltz_tensors.py`

The electronic transport coefficients with the constant relaxation time Boltzmann theory are computed as in reference [35]. The electrical conductivity tensor σ_{ij} can be expressed as an integral over the BZ:

$$\sigma_{ij} = \frac{e^2}{4\pi^3} \int_{BZ} \tau \sum_{n} v_n^i(\vec{k}) v_n^j(\vec{k}) \left(-\frac{\partial f_0}{\partial \epsilon} \right) d\vec{k} \tag{3.10}$$

where τ is the constant relaxation time, $v_n^i(\vec{k})$ is the i-th component of the electron velocity ($\vec{v}_n$) corresponding to the n-th band for each $\vec{k}$-point in the BZ, f_0 is the equilibrium distribution function, and ϵ is the electron energy.

Generalizing equation 3.10 it is also possible to define analogue expressions for the Seebeck coefficient, S, and the electronic contribution to thermal conductivity, κ_{el}. Following the notation from reference [36], we introduce the generating tensor $\mathcal{L}_\alpha$ ($\alpha = 0,1,2$):

$$\mathcal{L} = \frac{1}{4\pi^3} \int \tau \sum_{n} \vec{v}_n(\vec{k}) \vec{v}_n(\vec{k}) \left(-\frac{\partial f_0}{\partial \epsilon} \right) [\epsilon - \mu]^\alpha \tag{3.11}$$

where $\vec{v}_n(\vec{k}) \vec{v}_n(\vec{k})$ indicates the dyadic product and μ is the chemical potential. The coefficients σ, S, and κ_{el} can be expressed as follows:

$$\sigma = e^2 \mathcal{L}_0$$

$$S = -\frac{1}{Te}[\mathcal{L}_0]^{-1} \cdot \mathcal{L}_1 \tag{3.12}$$

$$\kappa_{el} = \frac{1}{T}(\mathcal{L}_2 - \mathcal{L}_1 \cdot [\mathcal{L}_0]^{-1} \cdot \mathcal{L}_1)$$

where T is the temperature.

3.3.7 Module: `do_epsilon.py`

In the limit of long wavelength (*i.e.* negligible momentum transfer) the optical properties of the material depends only on the frequency of the electro-magnetic field. The dielectric tensor can then be expressed in terms of the dielectric susceptibility $\chi_{ij}(\omega)$:

$$\epsilon_{ij}(\omega) = 1 + 4\pi\chi_{ij}(\omega) \tag{3.13}$$

The imaginary part of $\chi(\omega)$ in the single particle approximation can be written as: [29]

$$Im\chi_{ij}(\omega) = \frac{e^2\pi}{\omega^2\hbar m_0^2\Omega}\sum_{n,m,\vec{k}}\left[f_n(\vec{k}) - f_m(\vec{k})\right] \times p_{nm}^i(\vec{k})p_{nm}^j(\vec{k})\,\delta\left(\omega - \omega_{nm}(\vec{k})\right) \tag{3.14}$$

where m_0 is the bare electron mass, Ω is the unit cell volume, m and n are the band indices, $f_l(\vec{k})$ the Fermi-Dirac distribution evaluated on the band with index l at energy $E_l(\vec{k})$, $p^i(\vec{k})$ are the matrix elements of the momentum operator calculated over the states (both occupied and empty with indices m and n, and $\hbar\omega_{mn} = E_m(\vec{k}) - E_n(\vec{k})$ is the energy of the optical transition. The real part of the dielectric susceptibility can then be expressed using the Kramers-Kronig transformation of the imaginary part

$$Re\chi(\omega) = \frac{2}{\pi}\int_0^\infty z\frac{Im\chi(z)}{z^2 - \omega^2}\,dz \tag{3.15}$$

Alternatively, for semiconductors and insulators, one can evaluate directly the real part of the susceptibility as in equation 23 of reference [29]. See reference [35] for a comprehensive discussion of the evaluation of the calculation of the dielectric function in the PAO scheme.

3.3.8 Module: `do_Berry_curvature.py`, `do_spin_Berry_curvature.py`, `do_Berry_conductivity.py`, `do_spin_Hall_conductivity.py`

PAOFLOW automatizes the calculation of anomalous charge and spin transport quantities. We refer to the excellent review by Gradhand et al. [37] for an in depth discussion of these topics. The basic quantity computed by PAOFLOW is the Berry curvature (see figure 3.2):

$$\Omega_n^z(\vec{k}) = -\sum_{m\neq n} \frac{2Im\langle\psi_{n,\vec{k}}|v_x|\psi_{m,\vec{k}}\rangle\langle\psi_{m,\vec{k}}|v_y|\psi_{n,\vec{k}}\rangle}{(\omega_m-\omega_n)^2} \tag{3.16}$$

which provides, in turn, the anomalous Hall conductivity (AHC) (figure 3.3). Here v_x and v_y are obtained from the diagonal elements of the momentum operator defined in equation 3.6.

$$\sigma_{xy} = -\frac{e^2}{\hbar}\int_{BZ}\frac{d^3\vec{k}}{(2\pi)^3}\Omega^z(\vec{k}) \tag{3.17}$$

where

$$\Omega^z(\vec{k}) = \sum_n f_n\Omega_n^z(\vec{k}) \tag{3.18}$$

is the sum of the Berry curvature over the occupied bands (f_n is the Fermi-Dirac distribution).

In the case of variable fields, the Hall conductivity and magnetic circular dichroism (figure 3.4) can also be computed using

$$\sigma_{xy}(\vec{k}) = \frac{e^2}{\hbar}\int_{BZ}\frac{d^3\vec{k}}{(2\pi)^3}\sum_{m\neq n}\left[f_n(\vec{k}) - f_m(\vec{k})\right] \times \frac{2Im\langle\psi_{n,\vec{k}}|v_x|\psi_{m,\vec{k}}\rangle\langle\psi_{m,\vec{k}}|v_y|\psi_{n,\vec{k}}\rangle}{(\omega_m - \omega_n)^2 - \left(\omega + iW_{n,m}\right)^2} \tag{3.19}$$

where $W_{n,m}$ is the adaptive smearing parameter (equation 3.9).

Starting with relativistic (SOC) band structure PAOFLOW can also compute the spin Berry curvature:

$$\Omega_n^z(\vec{k}) = -\sum_{m\neq n}\frac{2Im\langle\psi_{n,\vec{k}}|j_x^z|\psi_{m,\vec{k}}\rangle\langle\psi_{m,\vec{k}}|v_y|\psi_{n,\vec{k}}\rangle}{(\omega_m - \omega_n)^2} \tag{3.20}$$

where $j_x^z = s_z, \vec{v}$ is the spin current operator with $s_z = \frac{\hbar}{2}(\beta, \Sigma:\ 4\times 4\ \text{Dirac matrices})$ [16] and the corresponding SHC (figure 3.5, equations 3.17 and 3.18). Similarly to equation 3.19, when fields are frequency dependent, PAOFLOW facilitates the calculation of the spin Hall conductivity (SHC) and magnetic circular dichroism (figure 3.6):

$$\sigma_{xy}(\vec{k}) = \frac{e^2}{\hbar}\int_{BZ}\frac{d^3\vec{k}}{(2\pi)^3}\sum_{m\neq n}\left[f_n(\vec{k}) - f_m(\vec{k})\right] \times \frac{2Im\langle\psi_{n,\vec{k}}|j_x^z|\psi_{m,\vec{k}}\rangle\langle\psi_{m,\vec{k}}|v_y|\psi_{n,\vec{k}}\rangle}{(\omega_m - \omega_n)^2 - \left(\omega + iW_{n,m}\right)^2} \tag{3.21}$$

where $W_{n,m}$ is the adaptive broadening parameter defined in equation 3.9.

3.4 Installation and Input File Structure

PAOFLOW does not need any specific setup provided that the required Python 2.7 modules are installed on the system. PAOFLOW is executed from a simple (but expandable) `main.py` module that calls `paoflow('input path', 'input file')` where the arguments specify the desired path and the XML input file name to be read at runtime (see the

file structure in the `examples` directory). PAOFLOW reads also two files from the execution of Quantum ESPRESSO: `data-file.xml` generated by the main run with `pw.x`, and `atomic-proj.xml` generated by the post-processing of `projwfc.x` (see also the discussion in Section 3.2).

3.4.1 Input File Format

In the following we discuss the individual input parameters in the `inputfile.xml` file. To preserve readability, in this description, the 'type' and 'size' XML tags have been neglected.

Listing 3.1: inputfile.xml – file format

```
<?xml version="1.0"?>
<root>
     <fpath>./dir.save/</fpath>
     …
     <out_vals>Hksp</out_vals>
     …
</root>
```

3.4.2 System Variables

Listing 3.2: inputfile.xml – system section

```
<fpath>./silicon.save/</fpath>
<restart>F</restart>
<verbose>F</verbose>
<non_ortho>F</non_ortho>
```

```
<write2file>F<write2file>

<shift_type>1</shift_type>

<shift>auto</shift>

<pthr>0.95</pthr>

<npool>1</npool>

<do_comparison>F</do_comparison>

<naw>

<a>0  0</a>

</naw>

<sh>

<a>1  2  0  1  2</a>

</sh>

<nl>

<a>1  1  1  1  1</a>

</nl>
```

- `fpath`
  - *Description.* Directory created by `projwfc.x` where the calculation (`data-file.xml`) and atomic projection (`atomic-proj.xml`) data are saved.
  - *Type.* String. Default '`dir.save`'.
  - *Example.* See listing 3.2
- `restart`
  - *Description.* Write data to disk at selected checkpoints to skip section of calculations in restart. Data are written in the uncompressed `.npz` format of `NumPy` to ensure maximum data transferability across architectures.
  - *Type.* Logical. Default = `False`

- *Example.* See listing 3.2

- `verbose`
  - *Description.* Writes run information on standard output.
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.2

- `non-ortho`
  - *Description.* Read overlaps to construct a non orthogonal PAO Hamiltonian. Necessary to perform ACBN0 calculations (see section 3.7). After the calculation of bands and band topology the basis is orthogonalized and the calculation proceeds in the new basis.
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.2

- `write2file`
  - *Description.* Write necessary data to perform ACBN0 calculations (see section 3.7).
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.2

- `write_binary`
  - *Description.* Write necessary data in binary format to perform ACBN0 calculations in AFLOWπ. [22]
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.2

- `writedata`
  - *Description.* Write 3-dim Berry curvature and spin Berry curvature for plotting (format suitable for the `Mayavi` 3D scientific data visualization and plotting package (http://docs.enthought.com/mayavi/mayavi/).
  - *Type.* Logical. Default = `False`

- *Example.* See listing 3.2

- `use_cuda`

- *Description.* Use nVIDIA CUDA libraries to perform fft on GPUs.

- *Type.* Logical. Default = `False`

- *Example.* See listing 3.2

- `shift_type`

  - *Description.* Selects the filtering and shifting scheme to construct the PAO Hamiltonians.

  - *Type.* Integer.

  - *Example.* See listing 3.2

  - *Request Syntax.* if 0 uses equation 3.1, if 1 uses equation 3.2 (default), if $2\ no\ correction$.

- `shift`

  - *Description.* Define the shifting parameter (κ in equations 3.1 and 3.2).

  - *Type.* String or Float

  - *Example.* See listing 3.2

  - *Request Syntax.* if 'auto', κ is chosen as the lowest eigenvalue of the highest band with projectability within the tolerance (default); if a number, amount of shift provided by the user.

  - *Units.* if Float, units are eV.

- `pthr`

  - *Description.* Tolerance on the projectability of bands: only bands with projectability > `pthr` will be considered.

  - *Type.* Float. Default = 0.95

  - *Example.* See listing 3.2

  - *Request Syntax.* For best Hamiltonian representability it must be a number between 0.9 and 1 (see reference [18] for an extensive discussion).

- `npool`
    - *Description.* Number of partitions of largest arrays to reduce the memory requirements and overhead of message passing in MPI. It can be used to optimize performance for large systems.
    - *Type.* Integer. Default = 1
    - *Example.* See listing 3.2
    - *Request Syntax.* Currently `npool` must be chosen so that the number of $\vec{k}$-points in the interpolated MP mesh divided by `npool` is an integer and this integer must be divisible by the number of cores.
- `do_comparison`
    - *Description.* Performs a comparison between the DFT eigenvalues from Quantum-ESPRESSO and the ones obtained from the PAO Hamiltonian. Useful when debugging for illustrative purposes. Writes a pdf file (`comparison`). Uses the `matplotlib` module.
    - *Type.* Logical. Default = `False`
    - *Example.* See listing 3.2
- `out_vals`
    - *Description.* Defines the data structures that the PAOFLOW module will return after all calculations have completed. Data structures are returned as a dictionary with the keys named as the variables are named in PAOFLOW.
    - *Type.* Dictionary. Default = {}
    - *Example.* See listing 3.2
- `naw`
    - *Description.* Dimensions of the atomic basis for each atom in the system. The order must be the same as the in the output of `projwfc.x`.
    - *Type.* `NumPy` array of Integers. Default = `np.array([0,0])`. Must be specified if external fields are required (see below in section 3.4.3).
    - *Example.* See listing 3.2
- `sh` and `nl`

- *Description.* Shell order `sh` and degeneracy `nl` of the atomic orbital basis in a spin-orbit calculation.
- *Type.* List of Integers. Default: `sh=[0,1,2,0,1,2], nl=[2,1,1,1,1,1]`. Must be specified if calculation has spin-orbit
- *Example.* See listing 3.2
- *Request Syntax.* `sh[:]` gives the order of shells with angular momentum l. `Nl[:]` gives the multiplicity of each l shell (depending on the pseudopotential one might have duplicate shells in the list of orbitals). The default values correspond to a system with two atoms, the first with one s, one p, and one d shell.

3.4.3 External Fields

Listing 3.3: inputfile.xml – external fields

```
<Efield>

<a>0 0 0</a>

</Efield>

<bval>0</bval>

<do_spin_orbit>F</do_spin_orbit>

<theta>0.0</theta>

<phi>0.0</phi>

<naw>9</naw>

<orb_pseudo>spd</orb_pseudo>

<lambda_p>0.0</lambda_p>

<lambda_d>0.0</lambda_d>
```

- `Efield`
  - *Description.* Magnitude of an electric field added to the diagonal elements of the PAO Hamiltonian.

- *Type.* Float. Default = 0.0
- *Example.* See listing 3.3
- *Units.* eV.

- `HubbardU`
  - *Description.* Add a Hubbard U to selected orbitals.
  - *Type.* `NumPy` array of Floats. Default = 0.0
  - *Example.* See listing 3.3
  - *Units.* eV.

- `bval`
  - *Description.* Index of the highest occupied band for alignment of the top state to 0.
  - *Type.* Integer. Default = 0
  - *Example.* See listing 3.3

- `do_spin_orbit`
  - *Description.* Add the spin-orbit interaction a posteriori in the PAO Hamiltonian.
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.3
  - *Request Syntax.* if `True` one must provide theta (θ), phi (ϕ), `naw`, `lambda_p` (λ_p), and `lambda_d` (λ_d).

- `theta` and `phi`
  - *Description.* Rotation angle around the z-axis and y-axis in cartesian coordinates, respectively
  - *Type.* Real
  - *Example.* See listing 3.3
  - *Units.* in degrees.

- `orb_pseudo`
  - *Description.* Type of pseudopotential used for each atom in the system.
  - *Type.* `NumPy` array of Strings. Default = `np.array(['spd'])`
  - *Example.* See listing 3.3
- `lambda_p` and `lambda_d`
  - *Description.* Spin-orbit coupling parameter for p and d states, respectively. The SOC will be included only in the outermost p and (or) d orbitals.
  - *Type.* Real
  - *Example.* See listing 3.3
  - *Units.* if Float, units are eV.

3.4.4 Band Structure and Band Topology

Listing 3.4: inputfile.xml – file format

```
<onedim>F</onedim>

<do_bands>F</do_bands>

<ibrav>0</ibrav>

<nk>2000</nk>

<band_topology>F</band_topology>

<spol>0</spol>

<ipol>0</ipol>

<jpol>0</jpol>
```

- `onedim`
  - *Description.* Interpolate a band structure calculated along a particular symmetry line in the BZ.
  - *Type.* Logical. Default = `False`

- *Example.* See listing 3.4
- *Request Syntax.* if `True` the non self consistent calculation of `pw.x` must be compatible in the choice of the reciprocal space path.

- `do_bands`
  - *Description.* Interpolate the band structure by Fourier transforming the PAO Hamiltonian in real space.
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.4
  - *Request Syntax.* if `True` the non self consistent calculation of `pw.x` must be compatible in the choice of the reciprocal space path.

- `ibrav`
  - *Description.* Bravais lattice index following the convention of Quantum-ESPRESSO. The $\vec{k}$-pints paths follow the AFLOW convention [24].
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.4

- `dkres`
  - *Description.* Resolution of the $\vec{k}$-grid along the high symmetry path.
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.4

- `band_topology`
  - *Description.* Calculates the Z_2 invariant and topological properties (band velocity, Berry and spin Berry curvature) along the band path in the BZ.
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.4

- `ipol, jpol, spol`

- *Description.* Polarization indices for the components of the Berry ($\Omega_{ipol,jpol}$) and spin Berry curvature ($\Omega^{spol}_{ipol,jpol}$). Components are calculated one at a time.
- *Type.* Integer. Default = 0
- *Example.* See listing 3.4

3.4.5 Hamiltonian Interpolation and Related Quantities

Listing 3.5: inputfile.xml – Hamiltonian interpolation

```
<double_grid></double_grid>

<nfft1>0</nfft1>

<nfft2>0</nfft2>

<nfft3>0</nfft3>

<smearing>gauss</smearing>

<do_dos>F</do_dos>

<do_pdos>F</do_pdos>

<emin>-1.0</emin>

<emax>2.0</emax>

<delta>0.01</delta>

<do_fermisurf>F</do_fermisurf>

<do_spintexture>F</do_spintexture>

<fermi_up>1.0</fermi_up>

<fermi_dw>-1.0</fermi_dw>
```

- `double_grid`
  - *Description.* Interpolation of the original PAO Hamiltonian on finer MP grids. Uses a “zero padding” algorithm.
  - *Type.* Logical. Default = `False`

- *Example.* See listing 3.5
- *Notes.* When `True` the code will automatically compute the gradient of the Hamiltonian and the momentum operators.

- `nfft1, nfft2, nfft3`
  - *Description.* Dimension of the dense FFT grid.
  - *Type.* Integer. Default = 0
  - *Example.* See listing 3.5
  - *Request Syntax.* `nfft1, nfft2, nfft3` must be even.

- `smearing`
  - *Description.* Choice of smearing protocol.
  - *Type.* String.
  - *Example.* See listing 3.5
  - *Request Syntax.* Possible choices are: '`None`' simple gaussian smearing with fixed `delta;` '`gauss`' adaptive gaussian smearing: '`m-p`' adaptive Methfessel-Paxton smearing (5th order Hermite polynomials).

- `delta`
  - *Description.* Width of the simple Gaussian smearing if `smearing=None` (see above).
  - *Type.* Float. Default = 0.1
  - *Example.* See listing 3.5
  - *Units. eV*.

- `do_dos, do_pdos`
  - *Description.* Calculate the density of states/projected density of states in the interval [emin,emax].
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.5

- `emin, emax`
    - *Description.* Energy interval for the calculation of the density of states/projected density of states.
    - *Type.* Float. Default = -10.0, 2.0
    - *Example.* See listing 3.5
    - *Units.* if Float, units are eV.
- `do_fermisurf`
    - *Description.* Evaluate the Fermi surface for energy values between `fermi_up` and `fermi_dw`.
    - *Type.* Logical. Default = `False`
    - *Example.* See listing 3.5
- `do_spintexture`
    - *Description.* Evaluate the spin texture isosurface for energy values between `fermi_up` and `fermi_dw`.
    - *Type.* Logical. Default = `False`
    - *Example.* See listing 3.5
- `fermi_up, fermi_dw`
    - *Description.* Energy interval for the calculation of the Fermi surface or spin texture.
    - *Type.* Float. Default = 0.1, -0.1
    - *Example.* See listing 3.5
    - *Units.* if Float, units are eV.

3.4.6 Transport and Optical Properties

Listing 3.6: inputfile.xml – Transport and optical tensors

```
<d_tensor>

     <a>0 0</a>
```

```
     <a>0 1</a>
     <a>0 2</a>
     <a>1 0</a>
     <a>1 1</a>
     <a>1 2</a>
     <a>2 0</a>
     <a>2 1</a>
     <a>2 2</a>
</d_tensor>
<temp>0.025852</temp><!--Room Temperature-->
<Boltzmann>F</Boltzmann>
<epsilon>F</epsilon>
<metal>F</metal>
<kramerskronig>F</kramerskronig>
<epsmin>0.0</epsmin>
<epsmax>10.0</epsmax>
<ne>500</ne>
<critical_points>F</critical_points>
<Berry>F</Berry>
<eminAH>-1.0</eminAH>
<emaxAH>1.0</emaxAH>
<ac_cond_Berry>F</ac_cond_Berry>
<spin_Hall>F</spin_Hall>
<eminSH>-1.0</eminSH>
```

```
<emaxSH>1.0</emaxSH>

<ac_cond_spin>F</ac_cond_spin>
```

- `d_tensor`
  - *Description.* Components of the dielectric tensor to be calculated.
  - *Type.* `NumPy` array of Integers. Default = All components are calculated
  - *Example.* See listing 3.6
- `t_tensor`
  - *Description.* Components of the Boltzmann transport tensors to be calculated.
  - *Type. Type.* `NumPy` array of Integers. Default = All components are calculated
  - *Example.* See listing 3.6
- `a_tensor`
  - *Description.* Components of the anomalous Hall magnetic circular dichroism tensors to be calculated.
  - *Type. Type.* `NumPy` array of Integers. Default = All components are calculated
  - *Example.* See listing 3.6
- `s_tensor`
  - *Description.* Components of the spin Hall and spin circular dichroism tensors to be calculated.
  - *Type. Type.* `NumPy` array of Integers. Default = All components are calculated
  - *Example.* See listing 3.6
- `epsilon`
  - *Description.* Compute the real and imaginary part of the dielectric function.
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.6

- `metal`
  - *Description.* Add intraband contribution to the real and imaginary part of the dielectric function.
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.6
- `kramerskronig`
  - *Description.* Compute the real part of the dielectric function using the Kramers-Kronig relation.
  - *Type.* Logical. Default = `True`
  - *Example.* See listing 3.6
  - *Request Syntax.* if `False` the program will use equation 23 from reference [29].
- `epsmin, epsmax`
  - *Description.* Energy interval for the calculation of the dielectric function.
  - *Type.* Float. Default = 0.0, 10.0
  - *Example.* See listing 3.6
  - *Units.* if Float, units are eV.
- `ne`
  - *Description.* Number of energy points in the [emin,emax] interval
  - *Type.* Integer. Default = 500
  - *Example.* See listing 3.6
  - *Notes.* For accurate integration of the Kramers-Kronig formula we recommend 1000 energy points/*eV.*
- `critical_points`
  - *Description.* Find $\vec{k}$-points where a band has zero derivative.
  - *Type.* Logical. Default = `False`

- *Example.* See listing 3.6

- `Berry`
  - *Description.* Evaluate Berry curvature and anomalous Hall conductivity (AHC).
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.6

- `eminAH, emaxAH`
  - *Description.* Energy interval for the calculation of the AHC.
  - *Type.* Float. Default = -1.0, 1.0
  - *Example.* See listing 3.6
  - *Units.* if Float, units are eV.

- `ac_cond_Berry`
  - *Description.* Evaluate magnetic circular dichroism spectrum in the interval [0.0, `shift`].
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.6

- `spin_Hall`
  - *Description.* Evaluate spin Berry curvature and spin Hall conductivity (SHC).
  - *Type.* Logical. Default = `False`
  - *Example.* See listing 3.6

- `eminSH, emaxSH`
  - *Description.* Energy interval for the calculation of SHC.
  - *Type.* Float. Default = -1.0, 1.0
  - *Example.* See listing 3.6
  - *Units.* if Float, units are eV.

- `ac_cond_spin`
    - *Description.* Evaluate spin circular dichroism spectrum in the interval [0.0, `shift`].
    - *Type.* Logical. Default = `False`
    - *Example.* See listing 3.6

3.5 Performance

Performance of PAOFLOW exploits massive parallelization over k-points throughout the code and over bands whenever possible (mainly in the Hamiltonian interpolation and the calculation of the gradients). Parallel performances have been analyzed on a Dell PowerEdge R730 server with two 2.4 GHz Intel Xeon E5-2680 v4 fourteen-core processors using `Example01`, run times of select FFTs on CPUs or GPUs are compared for MP grid sizes ranging from $12 \times 12 \times 12$ to $72 \times 72 \times 72$. These results are summarized in figure 3.9.

PAOFLOW demonstrates excellent scaling properties on manycore systems and possesses massively parallel capabilities.

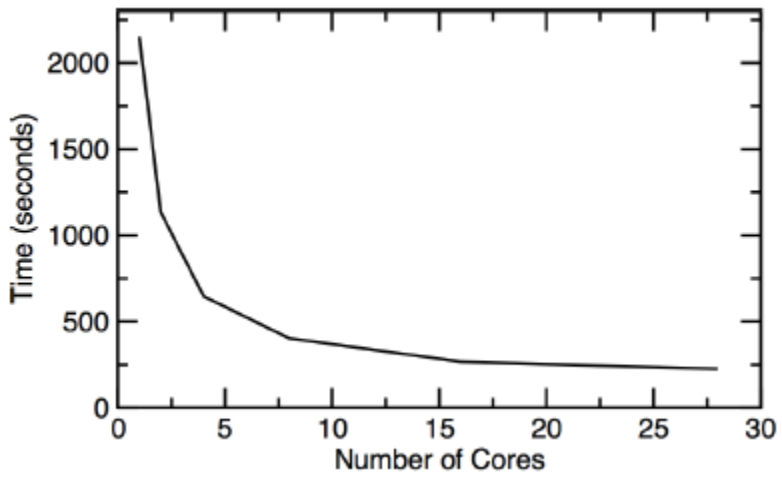


Figure 3.8: Parallelized routines provide run time scaling proportional to the number of cores used in a calculation, closely approaching the speed increase limit of Amdahl's Law.

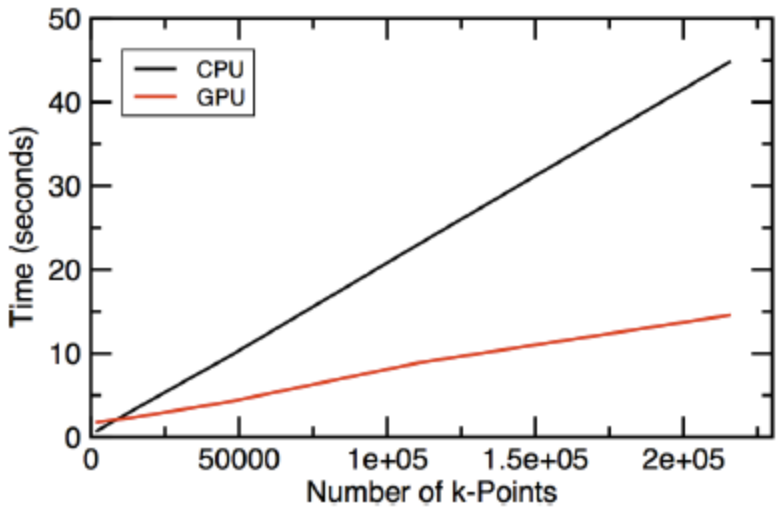


Figure 3.9: Executing Fast Fourier Transforms in parallel on GPUs yields a significant increase in run time performance on calculations with a large MP mesh.

3.6 Examples, Testing and Continuous Integration

PAOFLOW includes a full suite of examples in the `examples/` directory of the distribution. Examples can be run for testing and verifying the accuracy of the calculations upon different Python installations and computer architectures. Each example has a `Reference` directory where we have collected reference data for comparison. Examples can be run automatically using the module `run_examples.py`, which needs a small customization to define the directory where QE executables and PAOFLOW are installed. The module will run automatically all the DFT calculations and the PAOFLOW post processing steps. If run with no arguments it will run all examples, if an example name is given as an argument, only that example will be run. `run_examples.py` will also automatically verify the accuracy of the results against the data in `Reference` within a given tolerance to avoid false positive due to hardware-specific numerical precision. Examples included in the distribution are:

- `Example01`

- *Description.* Silicon with an spd pseudopotential: bands, density of states, projected density of states, Boltzmann transport and dielectric function.

- `Example02`

  - *Description.* Aluminum with an spd pseudopotential: density of states, Boltzmann transport and dielectric function `(metal=True)`.

- `Example03`

  - *Description.* Platinum in the local spin density approximation `(nspin=2)`: density of states, projected density of states, Boltzmann transport and dielectric function `(metal=True)`.

- `Example04`

  - *Description.* Iron with non-collinear magnetism and spin-orbit interaction: bands, band topology, density of states, spin Hall conductivity and spin circular dichroism.

- `Example05`

  - *Description.* Platinum with non-collinear magnetism and spin-orbit interaction: bands, band topology, density of states, spin Hall conductivity and spin circular dichroism.

- `Example06`

  - *Description.* AlP with *ad hoc* ACBN0 correction: bands, density of states, Boltzmann transport and dielectric function.

- `Example07`

  - *Description.* Bismuth with the effective spin-orbit interaction approximation: bands, density of states

Further tests can be added simply by incorporating more `exampleXX` directories with the same data structure as the existing ones. No modifications are needed in the `run_examples.py` module. This versatility is essential to ensure continuous integration and early detection of problems in new modules.

3.7 Externals

Modules and utilities that use data generated by PAOFLOW, or provide input information for a run, have been collected in the `src/external/` directory.

3.7.1 ACBN0 Calculations

The construction of the PAO Hamiltonian allows the direct computation of the direct and self-consistent evaluation of the on-site Coulomb U and exchange J parameters from the ACBN0 functional approach, recently introduced by some of us [30]. Thanks to the accurate PAO representation, the evaluation of the U and J for atoms in different chemical environments or close to topological defects (surfaces, interfaces, impurities, etc.) or for closed-shell atoms (like Zn) becomes trivial, thus overcoming the limitations of traditional linear response techniques with a computation cost comparable to a regular (LDA) PBE calculation. ACBN0 is integrated in the AFLOWπ framework and can be run seamlessly as part of a user-defined workflow. However, in order to provide maximum flexibility to users, we have included an external module to PAOFLOW that performs the ACBN0 calculation directly.

ACBN0 uses some modules and the `cints` library from `PyQuante` (http://pyquante.sourceforge.net/). In order to run the program properly, in `clib/` run the shell script `install.sh` (check the `$PATH` to the `include` files in the current python distribution). In order to run the main ACBN0 module `scfuj.py`, PAOFLOW must be run with the variable `write2file=True` (see listing 3.1).

3.7.2 Ballistic Transport

Calculations of the ballistic electrical conductance à *la* Landauer are naturally built on a

local representation of the electronic structure like the one provided by the PAO Hamiltonians. Our procedure reduces the problem of calculating electron transport [38,39] to a computationally inexpensive post-processing maintaining the predictive power and the accuracy of first principles methods. Briefly, using the Landauer approach the conductance is determined via the transmission function that can be written as: [38,39]

$$\mathcal{T}_{e\ell} = Tr(\gamma_L G_C^{\mathrm{r}} \Gamma_R G_C^{a}) \tag{3.22}$$

where $G_C^{(r,a)}$ are the retarded and advanced Green's functions of the conductor, respectively, and $\Gamma_{(L,R)}$ are functions that describe the coupling of the conductor to the leads. The Green's function for the whole system can be explicitly written as: [41]

$$G_C = (\epsilon - H_C - \Sigma_L - \Sigma_R)^{-1} \tag{3.23}$$

where Σ_L and Σ_R are the self-energy terms due to the semi-infinite leads.

Once the self-energy functions are known, the coupling functions $\Gamma_{(L,R)}$ can easily be obtained as [41]

$$\Gamma_{(L,R)} = i\left[\Sigma^r_{(L,R)} - \Sigma^a_{(L,R)}\right]. \tag{3.24}$$

The expression of the self-energies can be deduced along the lines of reference [38] using the formalism of principal layers in the framework of the surface Green's function matching theory. We obtain:

$$\Sigma_L = H_{LC}^{\dagger}(\epsilon - H_{00}^{L} - (H_{01}^{L})^{\dagger}\bar{T}_L)^{-1}H_{LC} \tag{3.25}$$

$$\Sigma_R = H_{CR}(\epsilon - H_{00}^{R} - H_{01}^{R}T_R)^{-1}H_{CR}^{\dagger},$$

where $H_{nm}^{L,R}$ are the matrix elements of the Hamiltonian between the layer orbitals of the left and right leads respectively, and $T_{L,R}$ and $\bar{T}_{L,R}$ are the appropriate transfer matrices. The latter are easily computed from the Hamiltonian matrix elements via an iterative procedure [38].

A simple generalization of the procedure above allows also for the calculation of surface protected bulk band structures, the essential and definitive tool to determine the existence or not of topologically protected surface states. In figure 3.10 we compare the surface projected bulk band structure of the Si (001) surface with an actual 32 atoms slab calculation.

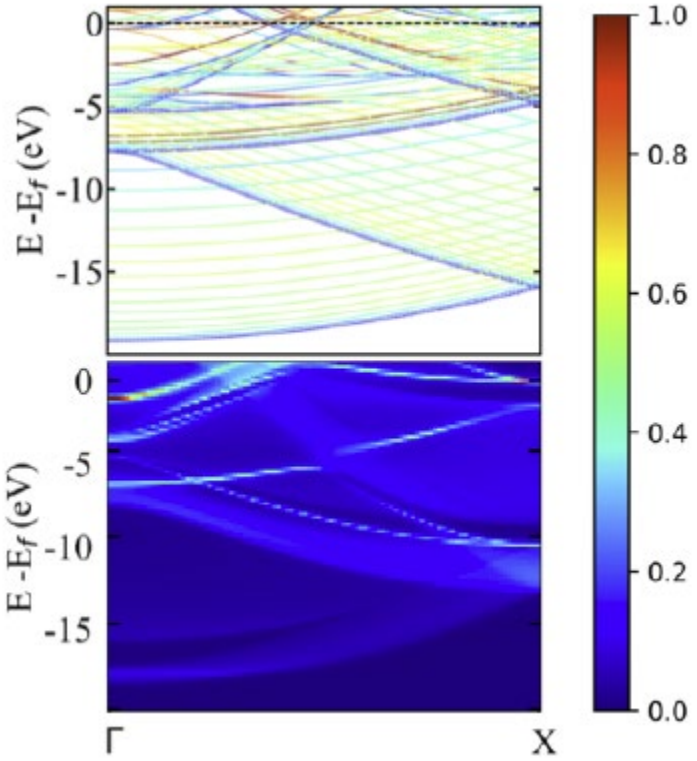


Figure 3.10: Si(001) surface projected band structure. Upper panel is the 32 atom slab projected onto the 2 outmost atomic planes. Lower panel is the surface projected bulk band strcutre using a semi-infinite bulk. Dark red (dark blue) is surface (bulk) bands.

The calculations are managed by the `transportPAO` utility, an adaptation of the WantT code [39] originally written for a Wannier functions representation of the DFT Hamiltonians.

`transportPAO` is written in Fortran90 and can be easily installed by running a standard `configure + make` procedure. Input for `transportPAO` is generated by `restart=True`.

3.8 Conclusions

With PAOFLOW we provide the electronic structure and materials community with a versatile and agile tool for the *ab initio* characterization of materials properties. The modularity of the code, its speed, and its accuracy, make it an ideal platform for the development of modern materials property databases.

3.9 Acknowledgments

We would like to thank Allan H MacDonald, Arrigo Calzolari, Priya Gopal, Marta dos Santos Guzmao, and Cormac Toher for useful discussions. We are grateful to the High Performance Computing Center and the University of North Texas and the Texas Advanced Computing Center at the University of Texas, Austin. The members of the AFLOW Consortium (http://www.aflow.org) acknowledge support by DOD-ONR (N00014-15-1-2266, N00014-13-1-0635, N00014-11-1-0136, and N00014-15-1-2863). The authors also acknowledge Duke University – Center for Materials Genomics. S.C. acknowledges the Alexander von Humbolt Foundation for financial support.

CHAPTER 4

ADVANCED MATERIALS MODELING WITH PAOFLOW 2.0*

Recent research in materials science open exciting perspectives to design novel quantum materials and devices, but it calls for quantitative predictions of properties which are not accessible in standard first principles packages. PAOFLOW is a software tool that constructs tight-binding Hamiltonians from self-consistent electronic wavefunctions by projecting onto a set of atomic orbitals. The electronic structure provides numerous materials properties that otherwise would have to be calculated via phenomenological models. In this paper, we describe recent re-design of the code as well as the new features and improvements in performance. In particular, we have implemented symmetry operations for unfolding k-points, which drastically reduces the runtime requirements of the first principles calculations, and we have provided internal routines of projections onto atomic orbitals enabling generation of real space atomic orbitals. Moreover, we have included models for non-constant relaxation time in electronic transport calculations, doubling the real space dimensions of the Hamiltonian as well as the construction of Hamiltonians directly from analytical models. Importantly, PAOFLOW has been now converted into a Python package, and is streamlined for use directly within other Python codes. The new object oriented design treats PAOFLOW's computation routines as calss methods, providing an API for explicit control of each calculation.

* This chapter is reproduced, in full, from the paper: *Advanced modeling of materials with PAOFLOW 2.0: New features and software design* by Frank T. Cerasoli, Andrew R. Supka, Anooja Jayaraj, Ilaria Siloi, Marcio Costa, Jagoda Sławińska, Stefano Curtarolo, Marco Fornari, Davide Ceresoli, and Marco Buongiorno Nardelli, submitted to Computational Materials Science in July 2021.

4.1 Introduction

Exploring phenomena and properties of novel materials requires accurate and efficient computational tools that can be easily customized and manipulated. In this context, ab initio tight-binding (TB) Hamiltonians constructed from self-consistent quantum-mechanical wavefunctions projected onto a set of atomic orbitals have been very successful, since they allow calculations for materials that cannot be properly addressed using only density functional theory (DFT) such as large moiré superstructures or properties of exotic quantum systems where spin and topology play an important role. PAOFLOW is a new software tool that employs an efficient procedure of projecting the full plane-wave solution on a reduced space of pseudoatomic orbitals [18,19], and provides an interpolated electronic structure to promptly compute a plethora of relevant quantities, including optical and magnetic properties, charge and spin transport as well as topological invariants. Importantly, in contrast with other common approaches the projection does not require any additional inputs and can be successively integrated in high-throughput calculations of arbitrary complex materials. The code has been employed in multiple areas of materials science since its initial release in 2016. In particular, several groups used it to compute the (spin) Berry curvature as well as spin and/or anomalous Hall conductivity (SHC and AHC) in a variety of mate- rials, ranging from β-W to magnetic antiperovskites [42-46]. Transport quantities, such as the electrical and thermal conductivity, were also computed in order to analyze carrier mobility in thermoelectrics [47,48].

The software package has recently undergone a major refactor, resulting in a large variety of properties that can be calculated as well as highly improved performance. Many improvements were made to simplify the user experience, to make the package more modular,

and to create an API for manipulating TB Hamiltonians. PAOFLOW, now installed as a Python package, features an object oriented design and contains an im- portable PAOFLOW class, allowing multiple Hamiltonians to be constructed and manipulated simultaneously. This framework enables high-throughput materials analysis within a single python file. This chapter outlines PAOFLOW's modified features, detail new functionalities, and provide a user manual for operating the various methods available within the package. Currently, PAOFLOW is publicly available under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version. It is also integrated in the AFLOWπ high-throughput framework [22] and distributed at http://www.aflow.org/src/aflowpi and http://www.aflow.org/src/paoflow [20,21].

4.2 Software Design and Installation

PAOFLOW is written in Python 3.8 (using the Python standard library, `NumPy`, and `SciPy`). Parallelization on CPUs uses the OpenMPI protocol through the `mpi4py` module. The PAOFLOW package can be easily installed on any hardware. Installation directly from the Python Package Index (PyPi) is possible with the single command: `pip install paoflow`.

Otherwise, one may clone the PAOFLOW repository from GitHub and install it from the root directory with the command: `python setup.py install`. The package requires no specific setup, provided that the prerequisite Python 3.8 modules are installed on the system. Example codes showcasing PAOFLOW's capabilities are housed on GitHub, in the `examples/` directory of the PAOFLOW repository.

Once installed, PAOFLOW can be imported into any Python code and used in conjunction with other software packages. It should be noted that, in PAOFLOW 1.0 two control

files were required: `main.py` to begin the execution, and `inputfile.xml` to provide details about the calculation. In PAOFLOW 2.1 the desired routines are called directly from the Python code, eliminating the need for `inputfile.xml`. However, an updated `main.py` file is provided in the repository's `examples/` directory, which allows use of the version 1.0 XML inputfile structure in PAOFLOW 2.1.

Generally, PAOFLOW requires a couple of basic calculations with the Quantum ESPRESSO [8,9] (QE) package. The first generates a converged electronic density and Kohn-Sham (KS) potential on an appropriate Monkhorst and Pack (MP) $\vec{k}$-point mesh (`pw.x`). A second, non self consistent calculation (`pw.x`) evaluates eigenvalues and eigenfunctions for a MP mesh on a larger band spectrum. Previous versions of PAOFLOW require that all the QE calculations include flags `nosym=.true.` and `noinv=.true.`, while version 2.1 can reconstruct equivalent $\vec{k}$-points from various symmetry operations and has no such requirement. Once the self consistent and non self consistent calculations are complete, the KS wavefunctions must be projected onto PAO basis functions. One of PAOFLOW's new capabilities is the generation of real space atomic orbitals, constructed from the product of radial components house in pseudopotential files and spherical harmonics specifying angular dependence. Either PAOFLOW can project the KS wavefunctions onto this internally constructed PAO basis set (section 4.3.2), or a third post-processing QE run with `projwfc.x` can perform the projection. If the projections are computed using QE, the eigenfunctions must be read by PAOFLOW explicitly before the construction of a PAO Hamiltonian (section 4.3.3). Alternatively, PAOFLOW can operate with no preprocessing requirements from QE, where built-in or user-defined models serve as the recipe from building Hamiltonians. Implementing these

models requires specific information about the atomic system *a priori*, and their usage is described in section 4.4.

Central to PAOFLOW is an internal object called the `DataController`, which has the sole responsibility of collecting and maintaining important information about the atomic system and its corresponding Hamiltonian. The `DataController` is initially populated with data from the QE run's `.save` directory, providing PAOFLOW's many routines with any information required for their calculation and simplifying function calls for the user. The `DataController` saves quantities which are used for later calculations, such as the Hamiltonian's gradient or the adaptive smearing parameters. Some calculations require other quantities as a prerequisite. For example, the `spin_Hall` routine requires the Hamiltonian's gradient, which means that `gradient_and_momenta` should be called first to populate the `DataController` with the gradient and momenta. The `DataController` stores system information in two dictionaries, one for strings and scalar attributes (`data_attributes`), and another for vector and tensor quantities (`data_arrays`). Such a structure allows computed quantities to be easily accessed from PAFOLWO and utilized in customized calculations defined by the user. Note that the dictionary keys are consistent with the naming conventions of PAOFLOW 1.0 to facilitate backwards compatibility with the XML inputfiles and minimize differences in the user experience when transitioning from the previous version.

4.3 Code Description and Package Usage

PAOFLOW's most fundamental procedure is the construction of accurate PAO Hamiltonians, and the code's object-oriented design allows users to manipulate multiple Hamiltonians easily. A PAOFLOW object is responsible for a single Hamiltonian, which is

constructed and operated on with PAOFLOW's class methods. If a tight binding model is used to construct the Hamiltonian, rather than from projections of KS wavefunctions on atomic orbitals, the new PAOFLOW object will create the Hamiltonian immediately. Otherwise, the KS wavefunctions are read from the output of a DFT engine, namely QE's `.save` directory. Next, the atomic orbitals are constructed and the KS wavefunctions are projected onto them with the `projections` routine, creating the PAO basis. If the projections are performed by QE's `projwfc.x` module, they must be read with `read_atomic_proj_QE`. The Hamiltonian is constructed with `build_pao_hamiltonian`, which allows PAOFLOW's other class methods to become functional. Listing 4.1 provides an example source code for building the PAOFLOW object, reading projections performed by QE, and constructing the PAO Hamiltonian. Listing 4.2 performs the same initialization procedure, but uses the internal atomic orbital projection scheme. An ellipsis appearing in any listing for this chapter indicates that other PAOFLOW routines may follow.

The following subsections outline PAOFLOW's individual routines and the arguments that they accept for control. These routines belong to the file `PAOFLOW.py`, located in the package's `src/` directory, and are meant for direct call by the user.

4.3.1 The Constructor: `PAOFLOW`

The PAOFLOW constructor acquires information about the python execution, the names of input/output/working directories, and about the atomic system. It build and populates the `DataController`, which will maintain the important quantities involved in calculations, handle communication in multi-core runs, and write files to disc when necessary.

For PAOFLOW to build a Hamiltonian, the constructor must be passed arguments with either the location of Quantum ESPRESSO's `.save` directory or required specifications for a TB model. The QE `.save` directory contains up to two files from the execution of QE: `data-file-schema.xml` (`data-file.xml` in previous versions of QE) generated by the main run with `pw.x`, and `atomic-proj.xml` generated by the post-processing too `projwfc.x` (if the projections are computed with QE). To implement a TB model from scratch, rather than starting from the KS wavefunction solutions of DFT, a dictionary containing the model's label and other required parameters should be passed into the **model** argument (see section 4.4).

Arguments for the constructor, `PAOFLOW`:

- **workpath** (string) – *Default:* `'./'` – Path to the working directory. Defaults to the current working directory.
- **outputdir** (string) – *Default:* `'output'` – Name of the directory to house output data files. The directory is crated automatically in the **workpath**, if it does not already exist.
- **inputfile** (string) – *Default:* `None` – This argument is primarily for backwards compatibility with PAOFLOW 1.0. It names the XML inputfile with control parameters described in the previous article. The XML inputfile also provides a consistent descriptor format for highly automated calculations, utilized by AFLOWπ.
- **savedir** (string) – *Default:* `None` – Name of the Quantum ESPRESSO `.save` directory, relative to the working directory.
- **model** (dict) – *Default:* `None` – Dictionary specifying parameters required to implement a TB model. See section 4.4.
- **npool** (integer) – *Default:* `1` – Number of batches to process when communicating between processors. This value will be automatically increased if the Hamiltonian size exceeds `mpi4py`'s limit for a single cross core message.
- **smearing** (string) – *Default:* `'gauss'` – Selects the broadening technique used to smooth computed quantities. Options include `'gauss'`, `'m-p'`(Methfessel-Paxton), and `None`.

- **acbn0** (bool) – *Default:* `False` – Read overlaps to construct a non-orthogonal PAO Hamiltonian. Necessary to perform ACBN0 calculations [30,31].
- **verbose** (bool) – *Default:* `False` – Flag for high verbosity. Set `True` to include additional information in the PAOFLOW output.
- **restart** (bool) – *Default:* `False` – Indicates the continuation of a previous run's saved state. Once the PAOFLOW object is instantiated, the `restart_load` routine should be called with the save file's prefix passed as an argument. An example is provided in listings 4.4 and 4.5.

4.3.2 `projections`

Perform projections of the KS eigenfunctions onto the atomic basis, constructed by PAOFLOW from information in the atomic pseudopotentials. This operation requires that PAOFLOW is instantiated by passing a QE `.save` directory with required XML file from the self consistent and non self consistent calculations. Listing 4.2 provides example usage of the `projections` routine, and a complete description of the projection methodology can be found in the Appendix of reference [18].

`projections` does not accept any arguments.

4.3.3 `read_atomic_proj_QE`

Read the projections of KS wavefunctions onto the atomic orbital basis of the pseudopotential, written to `atomic_proj.xml` by the QE routine `projwfc.x`. Any time that the Hamiltonian is built directory from the projections of QE, this routine should be called immediately after the constructor.

`read_atomic_proj_QE` does not accept any arguments.

4.3.4 `projectability`

The `projectability` routine determines which bands do not meet projectability requirements, flagging them for shift into the null space. The projectability $p_{\vec{k}}$ is a quantity measuring how well a KS Bloch state is represented by various orbitals of the PAO basis, as described in section 3.3. If $p_{\vec{k}} \approx 1$, the PAO basis set accurately represents that particular Bloch state for $\vec{k}$. $p_{\vec{k}} \ll 1$ indicates that the state is poorly represented and should be culled for removal. The projection threshold **pthr** selects the minimum allowed projectability for accepting a band. All bands which do not meet the criteria are projected to the null space during the Hamiltonian's construction, in one of two ways. Either as

$$\widehat{H} = AEA^{\dagger} + \kappa(I - AA^{\dagger}) \tag{4.1}$$

following reference [17] or

$$\widehat{H} = AEA^{\dagger} + \kappa(\mathrm{I} - \mathrm{A}(\mathrm{A}^{\dagger}A)^{-1}\mathrm{A}^{\dagger}) \tag{4.2}$$

as in reference [18]. Unless the **shift** argument is explicitly set to a floating point value, the shifting parameter κ is determined automatically by this routine. Which method is used to remove low-projectability bands during the Hamiltonian construction is selected by argument **shift_type**, in the `pao_hamiltonian` routine (section 4.3.5).

Arguments for `projectability`:

- **pthr** (float) – *Default:* `0.95` – The projectability threshold. All bands with a minimum projectability of the **pthr** value or higher are included in the Hamiltonian.
- **shift** (string or float) – *Default:* `'auto'` – Float to indicate the value (in *eV*) of the null space cutoff (κ in equations 4.1 and 4.2). Bands beneath the projectability threshold will be shifted to this value. Providing the default argument `'auto'` automatically sets **shift**'s value to the minimum energy of the first band to fail the projectability threshold.

4.3.5 pao_hamiltonian

This routine constructs the Hamiltonian in both real space and momentum space. After this routine is completed the data controller will contain arrays `HRs` and `Hks`, for the respective real space and k-space Hamiltonians.

Listing 4.1: main.py – Build Hamiltonian

```
from PAOFLOW import PAOFLOW


pao = PAOFLOW.PAOFLOW(savedir='system.save')

pao.read_atomic_proj_QE()

pao.projectability(pthr=0.95)

pao.pao_hamiltonian()

…
```

Arguments for `pao_hamiltonian`:

- **shift_type** (integer) – *Default:* `1` – Determines which method is used to (equation 4.2 by default) remove bands into the null space. `0` – equation 4.1, `1` – equation 4.2, or `2` – No shift.

- **insulator** (bool) – *Default:* `False` – Setting this flag to `True` asserts that the system is insulating, setting the top of the highest occupied band to $0eV$. The fermi energy is calculated for metallic systems, which corrects numerical discrepancies from the projection routine and irreducible wedge unfolding. This flag is set to `True` automatically if the QE output does not contain smearing parameters.

- **write_binary** (bool) – *Default:* `False` – Flag to write the files necessary for the ACBN0 routine. Overlaps from `projwfc.x` are required from prerequisite QE calculations, and this flag is required for further use with ACBN0.

- **expand_wedge** (bool) – *Default:* `True` – Applies provided symmetry operations to the k-mesh, unfolding the irreducible wedge into a Hamiltonian for every k-point in the brillouin zone. PAOFLOW routines act on the full grid of k-points (`True`), while ACBN0 only requires the irreducible wedge (`False`).

- **symmetrize** (bool) – *Default:* `False` – The Hamiltonian incurs numerical errors during the process of unfolding the wedge. Certain routines, such as `find_weyl_points`, are sensitive to the Hamiltonian's symmetric components. Setting this flag to `True` symmetrizes the Hamiltonian with an iterative procedure to reduce numerical errors in the Hamiltonian's symmetry.

- **thresh** (string) – *Default:* `1e-6` – The tolerance of symmetrization, if the procedure is performed.

- **max_iter** (string) – *Default:* `16` – The maximum number of iterations that the symmetrization procedure will perform.

4.3.6 `bands`

Compute the band structure along the AFLOW standard path for the specified Bravais lattice. A custom path can be created by defining the high symmetry points in a dictionary and the band path as a string (see listing 4.2). The path is Fourier interpolated to an arbitrary resolution, controlled by the argument **nk**.

Listing 4.2: main.py – Bands

```
from PAOFLOW import PAOFLOW


pao = PAOFLOW.PAOFLOW(savedir='system.save')

pao.projections()

pao.projectability()

pao.pao_hamiltonian()


path = 'G-X-S-Y-G'

sym_points = {'G':[0.0, 0.0, 0.0],

              'S':[0.5, 0.5, 0.0],

              'X':[0.5, 0.0, 0.0],
```

```
                'Y':[0.0, 0.5, 0.0]}

Pao.bands(ibrav=8, nk=1000,

          band_path=path, high_sym_points=sym_points)

…
```

Arguments for `bands`:

- **ibrav** (integer) – *Default:* `None` – The Bravais Lattice identifier, as specified by Quantum ESPRESSO.
- **band_path** (string) – *Default:* `None` – String of high symmetry point labels separated by '-' for a line connecting two points or '|' to place points directly adjacent on the path (see listing 4.2). If **band_path** is `None` the standard AFLOW path will be used [22].
- **high_sym_points** (dictionary) – *Default:* `None` – A dictionary mapping the string label of a high symmetry point to its three dimensional crystal coordinate (listing 4.2).
- **fname** (string) – *Default:* `'bands'` – File name prefix for the bands. One file is written for each spin component (see listing 4.2).
- **nk** (integer) – *Default:* `500` – Number of points to compute along the band path.

4.3.7 `interpolated_hamiltonian`

Fourier interpolation of the PAO Hamiltonian can increase the k-grid to an arbitrary density, as described and illustrated in the manuscript for PAOFLOW 1.0. The new desired dimensions should be specified for `nk1`, `nk2`, and `nk3`. The default behavior is to double the original **nk** dimension of any unspecified `nfft` argument. This routine populates the `DataController` with a new array `'Hksp'`, the interpolated Hamiltonian.

- **nfft1** (integer) – *Default:* `None` – The desired new dimension for the Hamiltonian's previous dimension `nk1`. The nfft dimension should be greater or equal to the previous **nk** dimension. If no argument is provided, the original k-grid dimension is doubled.

- **nfft2** (integer) – *Default:* `None` – New interpolated dimension for `nk2`, following the same scheme as **nfft1**.
- **nfft3** (integer) – *Default:* `None` – New interpolated dimension for `nk2`, following the same scheme as **nfft1** and **nfft2**.
- **reshift_Ef** (bool) – *Default:* `False` – Shift the Hamiltonian's diagonal elements such that zero lies at the Fermi energy.

4.3.8 `spin_operator`

The spin operator plays numerous roles in the PAOFLOW code. Generally, when the spin operator $\vec{S}_j$ is required PAOFLOW automatically constructs it. However, $\vec{S}_j$ can be explicitly computed by calling this routine. The shell levels and their occupations are automatically read from the pseudopotentials in the `.save` directory.

Arugments for `spin_operator`:

- **spin_orbit** (bool) – *Default:* `False` – Set this flag to `True` if spin orbit coupling is added at the PAO level (with `adhoc_spin_orbit` routine, section 4.3.10).

4.3.9 `add_external_fields`

PAOFLOW supports the addition of electric fields, on-site Zeeman fields, or Hubbard corrections directly to the PAO Hamiltonians [29,30]. Fields must be added after the Hamiltonian's construction. Listing 4.3 provides an example where an electric field and Hubbard correction are simultaneously added to a Hamiltonian.

Listing 4.3: main.py – External Fields

```
from PAOFLOW import PAOFLOW


pao = PAOFLOW.PAOFLOW(savedir='system.save')

pao.projections()
```

```
pao.projectability()

pao.pao_hamiltonian()



hubbardU = np.zeros(32, dtype=float)

hubbardU[1:4] = .1

hubbardU[17:20] = 2.31

pao.add_external_fields(Efield=[.1,0.,0.],

                        HubbardU=hubbardU)

…
```

Arguments for `add_external_fields`:

- **Efield** (ndarray or list) – *Default:* `[0.]` – An array of the form $[E_x, E_y, E_z]$, added to the diagonal elements of the Hamiltonian. Listing 4.3 provides an example of adding an electric field with one non-zero component.
- **Bfield** (ndarray or list) – *Default:* `[0.]` – An array of the form $[B_x, B_y, B_z]$, specifying the strength and direction of an on-site magnetic field.
- **HubbardU** (ndarray or list) – *Default:* `[0.]` – An array with one U entry for each orbital, e.g. $[U_1, U_2, \dots, U_n]$ where n is the number of orbitals. An example is provided in listing 4.3.

4.3.10 `adhoc_spin_orbit`

This routine allows the addition of spin orbit coupling at the PAO level. SOC is implemented for the following shell configurations, provided as the **orb_pseudo** argument: s, sp, spd, ps, ssp, $sspd$, $ssppd$.

Arguments for `adhoc_spin_orbit`:

- **naw** (ndarray or list) – *Default:* `[1]` – List containing the number of wavefunctions for each pseudopotential.

- **phi** (float) – *Default:* `0.` – Spin orbit azimuthal angle.
- **theta** (float) – *Default:* `0.` – Spin orbit polar angle.
- **lambda_p** (ndarray or list) – *Default:* `[0.]` – Array of p-orbital coupling strengths.
- **lambda_d** (ndarray or list) – *Default:* `[0.]` – Array of d-orbital coupling strengths.
- **orb_pseudo** (list) – *Default:* `['s']` – List of strings, containing the orbital configuration for each pseudopotential.

4.3.11 `doubling_Hamiltonian`

Double the real space dimensions of the Hamiltonian, creating a super cell in any desired direction. Naturally, the number of wavefunctions in the Hamiltonians increases by a factor $2^{n_x} \times 2^{n_y} \times 2^{n_z}$. Doubling is performed one time in each direction by default, and doubling can be suppressed by setting the argument for a dimension to 0.

Arguments for `doubling_Hamiltonian`:

- **nx** (int) – *Default:* `1` – Number of times to double the x dimension. If **nx** is set to 2 the resulting cell is 4 times larger in the x direction.
- **ny** (int) – *Default:* `1` – Number of times to double the y dimension.
- **nz** (int) – *Default:* `1` – Number of times to double the z dimension.

4.3.12 `topology`

The `topology` routine calculates various quantities along the AFLOW standard k-path. The user must call the `bands` routine before this one, to generate a k-path on which to compute the Z_2 invariance and topological properties.

Arguments for `topology`:

- **eff_mass** (bool) – *Default:* `False` – Setting this flag `True` computes the Hamiltonian's second derivative along the k-path. The effective mass is calculated

and saved to file with naming convention `effmass_IJ_S.dat`. `I` and `J` are the inplane polarization indices. `S` is the spin index.

- **Berry** (bool) – *Default:* `False` – Set `True` to calculate the Berry curvature along the k-path and writes the results to files `Omega_S_IJ.dat`. Here `I, J` are the inplane polarization indices. `S` is the spin polarization

- **spin_Hall** (bool) – *Default:* `False` – Setting `True` calculates the spin Hall conductivity along the k-path and writes the results to files `Omegaj_S_IJ.dat`. The indices `I, J`, and `S` are the same as in **Berry**. This routine automatically computes the Berry curvature, but no files for **Berry** are written unless its flag is explicitly set `True`.

- **spol** (integer) – *Default:* `None` – Spin polarization index of the spin Hall calculation. This selects which component of the spin operator is used to compute the spin current.

- **ipol** (integer) – *Default:* `None` – The inplane index for calculating the spin current.

- **jpol** (integer) – *Default:* `None` – The second inplane index for calculating the spin current.

4.3.13 `pao_eigh`

PAOFLOW's `pao_eigh` routine computes the eigenspectrum for the entire k-grid, saving the eigen -values and -vectors as new arrays in the `DataController` under keys `'E_k'` and `'v_k'` respectively. Some of the previously described functions compute the eigenvalues and eigenvectors along a path, such as `bands` and `topology`. This routine replaces values computed by such routine with a new set of eigenfunctions, running across the entire Brillouin Zone.

No arguments are accepted when calling `pao_eigh`.

4.3.14 `trim_non_projectable_bands`

Remove eigenvalues and momenta, from respective data arrays, which do not meet the projectability requirements set by `projectability`. This routine should be called after

`pao_eigh`, if such trimming is desired.

No arguments are accepted by `trim_non_projectable_bands`.

4.3.15 `fermi_surface`

Compute the bands with energies between **fermi_up** and **fermi_dw**. The results are saved in the NumPy .npz format with naming convention `Fermi_surf_band_N_M.dat`, where `N` is the band index and M is the spin index. The Fermi surface is saved with resolution of the existing k-grid.

Arguments for `fermi_surface`:

- **fermi_up** (float) – *Default:* `1.` – The upper energy bound for selecting bands. Bands within the range [**fermi_dw**, **fermi_up**] are included.
- **fermi_dw** (string) – *Default:* `-1.` – The lower energy bound for selecting bands.

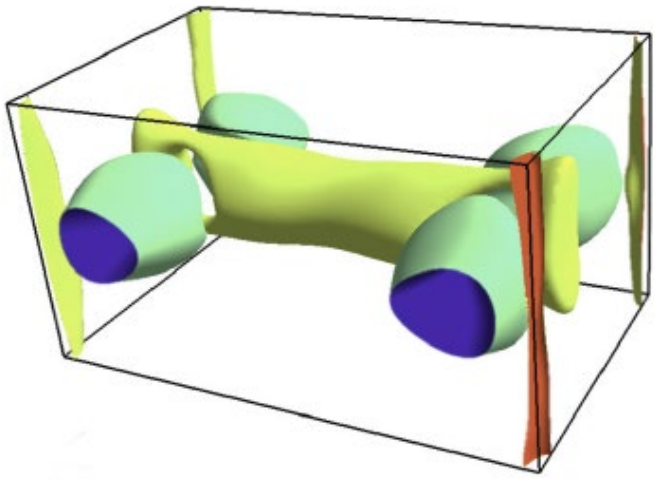


Figure 4.1: Fermi surface of FeP calculated on a ultra-dense k-grid in PAOFLOW and visualized in FermiSurfer [49]. For description of DFT calculations see reference [30].

4.3.16 `gradient_and_momenta`

The Hamiltonian's gradient is initially computed in real space, as it takes a simpler form. Afterward, it is Fourier transformed bank into reciprocal space. Thus, the Hamiltonian's

gradient takes the form

The momentum is given by

$$\vec{\nabla}_k \hat{H}(\vec{k}) = \sum_{\alpha} i\vec{R}\, exp(i\vec{k}\cdot\vec{R})\hat{H}(\vec{R}) \tag{4.3}$$

where $\hat{H}(\vec{R})$ is the real space PAO matrix and $|\psi_n(\vec{k})\rangle = exp(-i\vec{k}\cdot\vec{R})|u_n(\vec{k})\rangle$ are Bloch's functions [35]. The Hamiltonian's derivative is saved under a new array key `'dHksp'` in the `DataController`.

Next, the momenta are computed from the Hamiltonian's gradient as

$$\vec{p}_{nm}(\vec{k}) = \langle\psi_n(\vec{k})|\hat{p}|\psi_m(\vec{k})\rangle = \langle u_n(\vec{k})|\frac{m_0}{\hbar}\vec{\nabla}_k\hat{H}(\vec{k})|u_m(\vec{k})\rangle \tag{4.4}$$

Additionally, the Hamiltonian's second derivative can be computed by setting the **band_curvature** argument to `True`.

Arguments for `gradient_and_momenta`:

- **band_curvature** (bool) – *Default:* `False` – Compute the Hamiltonian's second derivative, stored as an array in the `DataController` under the key `'d2Ed2k'`.

4.3.17 `adaptive_smearing`

Generate the adaptive smearing parameters, stored in the `DataController` as `'deltakp'`, used to compute quantities on energy intervals, such as the density of states or spin Hall conductivity. Allowed adaptive smearing types are gaussian (`'gauss'`), Methfessel Paxton (`'m-p'`), or `None`.

Arguments for `adaptive_smearing`:

- **smearing** (string) – *Default:* `'gauss'` – Method of broadening used to smooth the discrete sampling of quantities computed on energy intervals.

4.3.18 `dos`

Compute the density of states (dos) and/or projected density of states (pdos) within a user defined energy range. If this routine is called after `adaptive_smearing`, the `'deltakp'` smearing parameter is used to smooth the dos calculations.

Arguments for `dos`:

- **do_dos** (bool) – *Default:* `True` – Flag to control whether the dos is computed.
- **do_pdos** (bool) – *Default:* `True` – Flag to control whether the pdos is computed
- **delta** (float) – *Default:* `0.01` – Width of the gaussian at each energy, used to smooth the dos curves. If it has been computed with the `adaptive_smearing` routine, `'deltakp'` replaces this quantity.
- **emin** (float) – *Default:* `-10.` – Lower limit for the energy range considered.
- **emax** (float) – *Default:* `2.` – Upper limit for the energy range considered.
- **ne** (integer) – *Default:* `1000` – The number of points to evaluate within the energy range [**emin, emax**].

4.3.19 `z2_pack`

Writes the real space Hamiltonian to a `bxsf` file, for use with Z2 Pack [50].

- **fname** (string) – *Default:* `'z2pack_hamiltonian.bxsf'` – Name for the `bxsf` file, written to PAOFLOW's output directory.

4.3.20 `spin_texture`

Compute the spin texture as the spin operator's expectation value for each band and for each k-point:

$$\vec{\Omega}_n(\vec{k}) = \langle \psi_n(\vec{k}) | \vec{S}_j | \psi_n(\vec{k}) \rangle \tag{4.5}$$

$\vec{S}_j$ is the spin operator, and $|\psi_n(\vec{k})\rangle$ are the momentum space PAO wavefunctions for

band index n. The spin texture is computed for bands which have values within the energy range specified by **fermi_up** and **fermi_dw**. Results are written to file in the NumPy `.npz` format, with naming convention `spin_text_band_N.npz`. Here, `N` is the band index, and each file contains spin texture computed on PAOFLOW's k-grid.

Arguments for `spin_texture`:

- **fermi_up** (float) – *Default:* `1.` – The spin texture is computed only for bands which contain energies beneath this upper bound.
- **fermi_dw** (float) – *Default:* `-1.` – The spin texture is computed only for bands which contain energies above this lower bound.

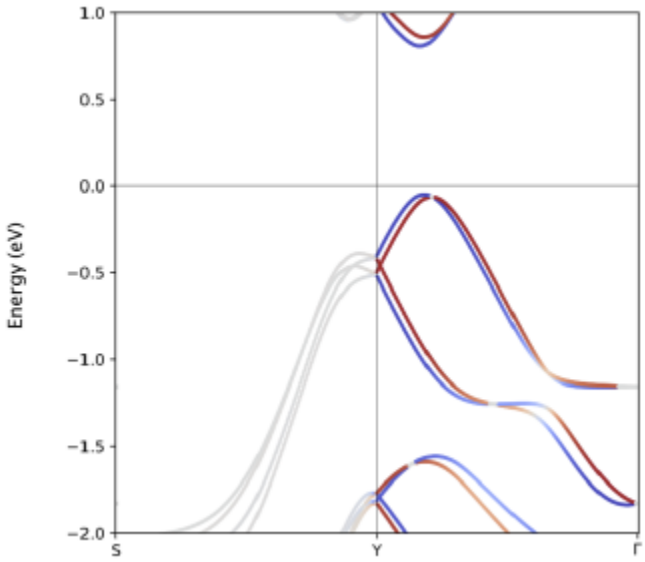


Figure 4.2: Spin texture (S_z) of two-dimensional ferroelectric SnTe along high-symmetry lines (`example08`) (see chapter 6).

4.3.21 `anomalous_Hall`

Calculating AHC relies on accurate evaluation of the Berry curvature and requires a preliminary run of `gradient_and_momenta`. PAOFLOW implements a standard Kubo formula for evaluating the k-resolved Berry curvature [25]. Further details are left to the previous manual and a comprehensive reference to Gradhand *et al* [37]. The AHC is computed

with adaptive smearing, provided that the broadening parameters are calculated beforehand by `adaptive_smearing`.

Arguments for `anomalous_Hall`:

- **do_ac** (bool) – *Default:* `False` – Compute the magnetic circular dichroism (MCD) on the same energy range [**emin**, **emax**].
- **emin** (float) – *Default:* `-1.` – The minimum energy in the range on which the AHC is computed.
- **emax** (float) – *Default:* `1.` – The maximum energy in the range. 500 points are evaluated within the interval [**emin**, **emax**].
- **fermi_up** (float) – *Default:* `1.` – Selects the upper energy bound for evaluating the Berry curvature.
- **fermi_dw** (float) – *Default:* `-1.` – Selects the lower energy bound for evaluating the Berry curvature.
- **a_tensor** (list) – *Default:* `None` – List of tensor elements to evaluate. For example, setting this argument to `[[0,0],[1,2]]` calculates the two components xx and yz. All 9 components are computed if the argument is left as `None`.

4.3.22 `spin_Hall`

The spin Hall conductivity (SHC, σ_{ij}^{k}) for spin polarization along **k** and charge (spin) current along i (j), is computed in a similar manner to AHC [28]. Here, evaluation of the *spin* Berry curvature is performed with the spin operator and Hamiltonian gradient as ingredients. Again, the `gradient_and_momenta` routine is a prerequisite, and running `adaptive_smearing` beforehand controls the inclusion of broadening parameters in the SHC calculation.

Arguments for `spin_Hall`:

- **twoD** (bool) – *Default:* `False` – Setting this flag `True` outputs the `spin_Hall` quantities in 2-dimensional units Ω^{-1}, removing any dependence on the sample

height. It is assumed that the dimensions of interest are oriented in the xy plane, and the sample height is in z.

- **do_ac** (bool) – *Default:* `False` – Compute the spin circular dichroism (SCD) on the same energy range, [**emin**, **emax**].

- **emin** (float) – *Default:* `-1.` – The minimum energy in the range on which the SHC is computed.

- **emax** (float) – *Default:* `1.` – The maximum energy in the range. Again, 500 points are evaluated in the interval [**emin**, **emax**].

- **fermi_up** (float) – *Default:* `1.` – Selects the upper bound for evaluating the spin Berry curvature.

- **fermi_dw** (float) – *Default:* `-1.` – Selects the lower bound for evaluating the spin Berry curvature.

- **s_tensor** (list) – *Default:* `None` – List of tensor elements to evaluate. To calculate x_xx and z_xy use `[[0,0,0],[2,0,1]]`. If the argument is left as `None` all 27 components are computed.

4.3.23 `doping`

Determine the chemical potential corresponding to a specified doping concentration and temperature range.

- **tmin** (float) – *Default:* `300` – Minimum temperature for which to evaluate the chemical potential.

- **tmax** (float) – *Default:* `300` – Maximum temperature for which to evaluate the chemical potential.

- **nt** (integer) – *Default:* `1` – The number of temperatures to evaluate in the range [**tmin**, **tmax**].

- **delta** (float) – *Default:* `0.01` – Gaussian broadening width, used to smooth the density of states along the energy range. Doping calculation involves an integration over density of states and therefore includes a call to the `dos` module.

- **emin** (float) – *Default:* `-1.` – Lowest value of energy of the occupied bands.

- **emax** (float) – *Default:* `1.` – At least the minimum of the conduction bands to obtain accurate results.
- **ne** (integer) – *Default:* `1000` – Number of points in the energy grid.
- **doping_conc** (float) – *Default:* `0.` – The doping concentration in carriers/cm^3 for which to compute the chemical potential. Specify negative value for n-type doping and positive value for p-type doping.
- **core_electrons** (integer) – *Default:* `0` – If the total number of electrons in the lower energy bands is known, this value can be input here. In this case, **emin** does not have to be the lowest energy value of occupied bands but can be instead set above energies of the core bands, to speed up integration.

4.3.24 `density`

Calculate the electronic density on a real space grid, performed for silicon in listing 4.4 and displayed in figure 4.1. Wavefunctions in k-space (produced by `pao_eigh`) are required as a prerequisite, and the PAO projections *must* be performed by PAOFLOW's `projections` method. This algorithm serves as a recipe for constructing the real space PAO wavefunctions. Although this is currently the only routine to utilize such construction, future versions of PAOFLOW will include other methods for computing spatially resolved quantities. The grid dimension defaults to $48 \times 48 \times 48$ but can be specified with the optional arguments **nr1**, **nr2**, and **nr3**.

Listing 4.4: main.py – Density

```
from PAOFLOW import PAOFLOW


pao = PAOFLOW.PAOFLOW(savedir='system.save')

pao.projections()

pao.projectability()

pao.pao_hamiltonian()
```

```
pao.pao_eigh()


pao.density(nr1=48, nr2=48, nr3=48)


pao.finish_execution()
```

Arguments for `density`:

- **nr1** (integer) – *Default:* `48` – Number of points in the first dimension of the real space grid, over which to compute the charge density.
- **nr2** (integer) – *Default:* `48` – Number of points in the seconds dimension of the real space grid.
- **nr3** (integer) – *Default:* `48` – Number of points in the third dimension of the real space grid.

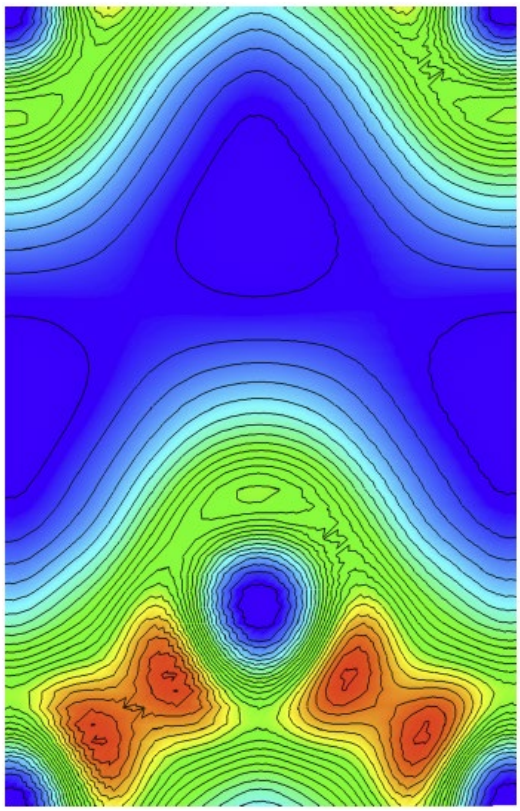


Figure 4.3: Electronic density for diamond structure of silicon on the $\langle 1, 0, -1 \rangle$ plane cut, calculated from the real space PAO wavefunctions (`example01`).

4.3.25 `transport`

Calculate the transport properties, such as electrical conductivity, Seebeck coefficients, and thermal conductivity. The transport properties are computed in the constant relaxation time approximation, unless built-in or user defined τ models are provided. See `Example09` for detailed information on specifying models for the relaxation time τ.

Arguments for `transport`:

- **tmin** (string) – *Default:* `300` – Minimum temperature for which to evaluate transport properties.
- **tmax** (string) – *Default:* `300` – Maximum temperature for which to evaluate transport properties.
- **nt** (string) – *Default:* `1` – The number of temperatures to evaluate in the range [**tmin**, **tmax**].
- **emin** (string) – *Default:* `1` – Minimum value in the energy grid [**emin**, **emax**].
- **emax** (string) – *Default:* `10` – Maximum value in the energy grid.
- **ne** (string) – *Default:* `1000` – Number of points in the energy range [**emin**, **emax**].
- **scattering_channels** (list) – *Default:* `None` – List of strings and/or `TauModel` objects containing the scattering models to be included in the calculation of τ.
- **scattering_weights** (list) – *Default:* `None` – Initial guess for the parameters a_{imp}, a_{ac}, a_{op} etc to be used for the fitting procedure if fit is set to `True`. The default behavior with this argument set to `None` is to use unity as every scattering weight.
- **tau_dict** (dict) – *Default:* `{}` – Dictionary of parameters required for the calculation of scattering models.
- **write_to_file** (bool) – *Default:* `True` – Path to the working directory. Defaults to the current working directory.
- **save_tensors** (bool) – *Default:* `False` – Path to the working directory. Defaults to the current working directory.

4.3.26 `find_weyl_points`

Perform a search for Weyl points within the first Brillouin Zone. The search identifies Weyl point candidates by utilizing scipy's `minimize` function with the 'L-BFGS-B' algorithm.

Arguments for `find_weyl_points`:

- **symmetrize** (bool) – *Default:* `False` – Use QE symmetry operations to unfold equivalent k-points. If equivalent k-points are Weyl points, all such points are reported.
- **search_grid** (list) – *Default:* `[8,8,8]` – Dimensions of the grid on which the minimization routine is performed. Bands are Fourier interpolated on this grid to improve resolution.

4.3.27 `restart_dump`

PAOFLOW's computational state can be saved at any time with the `restart_dump` routine. Data is stored in the `json` format, and the naming convention for such files can be chosen with the **fname** prefix argument. Each processor saves a file in the **workpath** directory with name `fname_prefix_N.json`, where `N` is the core's rank. For this reason, restarted calculations must be executed with the same number of cores. See listing 4.5 for an example where the gradient and momenta are computed and dumped for reuse in another calculation.

Listing 4.5: main.py – Restart (dump)

```
from PAOFLOW import PAOFLOW


pao = PAOFLOW.PAOFLOW(savedir='system.save')

pao.projections()

pao.projectability()

pao.pao_hamiltonian()

pao.pao_eigh()
```

```
pao.gradient_and_momenta()

pao.restart_dump(fname_prefix='pao')
```

Arguments for `restart_dump`:

- **fname_prefix** (string) – *Default:* `'output'` – Path to the working directory. Defaults to the current working directory.

4.3.28 `restart_load`

Recover PAOFLOW's calculation state from a previous run, saved to the `json` format with `restart_dump`. A restarted run must be executed with the same number of cores as the run which produced the dump files. Listing 4.6 provides example usage for `restart_load`.

Listing 4.6: main.py – Restart (load)

```
from PAOFLOW import PAOFLOW



pao = PAOFLOW.PAOFLOW(savedir='system.save')

pao.restart_load(fname_prefix='pao')

pao.adaptive_smearing()

…
```

Arguments for `restart_load`:

- **fname_prefix** (string) – *Default:* `'output'` – Path to the working directory. Defaults to the current working directory.

4.3.29 `finish_execution`

Conclude the PAOFLOW run and remove references to memory intensive quantities.

Details about the execution are provided, such as run duration and total memory requirements. This routine should be called once all desired calculations are performed for a given PAOFLOW object, especially if the code continues to create other PAOFLOW Hamiltonians.

`finish_execution` accepts no arguments.

4.4 Tight Binding Models

PAOFLOW is capable of generating a Hamiltonian from analytical tight binding models, such as the Kane-Mele or Slater-Koster models [51,52]. Each type of model requires specification of simple parameters to provide details about which model to use and about the system's properties, such as hopping parameters, lattice constant, etc. These parameters, including the label selecting the model to implement, should be initially stored in a dictionary which is subsequently passed into PAOFLOW's constructor as the **model** argument. Once the Hamiltonian is constructed, PAOFLOW's class methods can be applied in the standard fashion to compute desired quantities about the modeled system. Two examples are provided in listings 4.7 and 4.8, and others are housed within the package's `examples/` directory.

4.4.1 Cubium

Create a Hamiltonian for a single atom in the simple cubic geometry, containing a single band with one orbital per site. The hopping parameter is defined by including an entry in the parameters dictionary with key `'t'`, and should have units of *eV*.

Required dictionary entries:

- *Key*: `'label'` – Keyword identifier for the model: `'cubium'`, in this case. The labels are not case sensitive

- *Key*: `'t'` – The hopping parameter for nearest neighbor interactions, in units of *eV*.

4.4.2 Cubium 2

Creates a Hamiltonian for a single atom in the simple cubic geometry, implementing the double band model with two orbitals per site. The hopping parameter and band gap energy are given by `'t'` and `'Eg'` respectively.

Required dictionary entries:

- *Key*: `'label'` – Keyword identifier for the model: `'cubium2'`
- *Key*: `'t'` – The hopping parameter for nearest neighbor interactions.
- *Key*: `'Eg'` – Band gap energy, in *eV*.

4.4.3 Graphene

A simple TB model for graphene, considering only nearest neighbor interactions. The hopping parameter is specified with parameter `'t'`. The lattice constant is taken as $a = 2.46$ Å, and lattice vectors are standard for graphene. $\vec{a}_1 = a\langle 1,0,0\rangle$, $\vec{a}_2 = a\langle \frac{1}{2}, \frac{\sqrt{3}}{2}, 0\rangle$, $\vec{a}_3 = a\langle 0,0,10\rangle$.

Required dictionary entries:

- *Key*: `'label'` – Keyword identifier for the model: `'graphene'`
- *Key*: `'t'` – The hopping parameter for nearest neighbor interactions.

4.4.4 Kane Mele

Construct a Kane-Mele Hamiltonian for graphene. The first nearest neighbors are handled in the standard manner, with hopping parameter `'t'`. Second nearest neighbors are

treated with spin depended amplitude, characterized by the parameter `'soc_par'`. See listing 4.7 for an example.

Listing 4.7: main.py – Kane-Mele model

```
from PAOFLOW import PAOFLOW


model = {'label':'Kane_Mele', 't':1.0,
         'soc_par':0.1, 'alat':1.0}


pao = PAOFLOW.PAOFLOW(model=model, outputdir='./kane_mele')
…
```

Required dictionary entries:

- *Key*: `'label'` – Keyword identifier for the model: `'kane_mele'`
- *Key:* `'alat'` – The lattice parameter, a. The lattice vectors are the same as in section 4.4.3.
- *Key*: `'t'` – The hopping parameter for nearest neighbor interactions.
- *Key*: `'soc_par'` – The spin-orbit coupling parameter for second nearest neighbor interactions.

4.4.5 Slater Koster

A generalized Slater Koster TB model in the two-center approximation, considering only s and p orbitals of first nearest neighbors. The user must specify the lattice vectors, the atomic positions, the included orbitals for each atom, and the hopping parameters. See listing 4.8 for further details.

Required dictionary entries:

- *Key*: `'label'` – Keyword identifier for the model: `'slater_koster'`

- *Key:* `'a_vectors'` – A `NumPy` array containing the three primitive lattice vectors.

- *Key*: `'atoms'` – A dictionary with entries specifying atomic information for each atom. Dictionary keys label the atoms numerically with strings (e.g. the first atom has key `'0'`), and the corresponding values are dictionaries with information about the atom. The species, position (in crystal coordinates), and string identifier for each represented orbital should be saved in the atomic dictionary with respective keys: `'name'`, `'tau'`, and `'orbitals'`. The name is simply a string, the atomic position is a 3-vector, and orbitals is a list of strings denoting the orbitals belonging to each atom. See listing 4.8 for an example.

- *Key*: `'hoppings'` – A dictionary defining the various hopping strengths, in *eV*. The Slater-Koster hopping parameters should be labeled sss, sps, pps, and ppp.

Listing 4.8: main.py – Slater-Koster model

```
from PAOFLOW import PAOFLOW

import numpy as np


model = {'label':'slater_koster}


avecs = np.array([[.5,.5,0],

                  [.5,0,.5],

                  [0,.5,.5]])


atoms = {'0':

         {'name':'Si',

          'tau':[0,0,0],

          'orbitals':['s','px','py','pz']},

         '1':
```

```
          {'name':'Si',
           'tau':[.25,.25,.25],
           'orbitals':['s','px','py','pz']}}


hops = {'sss':-2.36233, 'sps':1.86401,
         'pps': 2.85882, 'ppp':-0.94687}


model['a_vectors'] = avecs
model['atoms'] = atoms
model['hoppings'] = hops


pao = PAOFLOW.PAOFLOW(model=model,
outputdir='./slater_koster')


…
```

4.5 Scattering Models

PAOFLOW supports a diverse set of scattering effects by allowing users to implement temperature and energy dependent models for the relaxation time parameter τ. Functional models are defined with the `TauModel` class. There are many built-in models, which only require the specification of empirical constants, and users can define new models to pass into PAOFLOW directly. A Python dictionary containing required parameters for any selected built-in models must be passed to the transport routine as the **tau_dict** argument (see listing 4.9).

Table 4.1 details the various constant parameters and their key strings for dictionary entries. Tau models are necessarily dependent on two quantities, the temperature and the Hamiltonian's energy eigenvalues. Other varying parameters can be supplied to `TauModels` through the params dictionary. As such, a python function accepting three arguments (the temperature, the energy, and the parameters dictionary) is the required format when constructing a custom `TauModel` object. The τ for each included model are computed, by evaluating the `TauModel` functions. Then, the τs are harmonically summed to obtain the effective τ for all scattering channels. The functional form for a `TauModel` is presented in listing 4.8 and a usage case in `example10`.

Listing 4.9: main.py – TauModel

```
From PAOFLOW.defs.TauModel import TauModel
from PAOFLOW import PAOFLOW


pao = PAOFLOW.PAOFLOW(savedir='system.save')
pao.projections()
pao.projectability()
pao.pao_hamiltonian()
pao.pao_eigh()
pao.gradient_and_momenta()
pao.adaptive_smearing()


# Define the functional model for acoustic scattering
rho = 5.3e3;  v = 5.2e3;  D = 7 * 1.6e-19
```

```
m = .7 * 9.11e-13;  h_bar = 6.58e-16
ac_const = 2 * np.pi * h_bar**4 * rho * v**2
ac_const /= ((2*m)**(3/2) * D**2)
def acoustic_scat ( temp, ene, params ):
  return ac_const / (temp*np.sqrt(ene))

# Define TauModel object
ac_model = TauModel(function=acoustic_scat)
channels = [ac_model, 'optical']

# Define parameters for built-in models
tau_params = {'ms':0.7, 'hwlo':[0.03536],
              'eps_inf':11.6, 'eps_0':13.5}

pao.transport(scattering_channels=channels,
              tau_dict=tau_params)

pao.finish_execution()
```

4.5.1 Charged Impurity Scattering

In order to include the effect of electron scattering from impurities, include `'impurity'` in the list **scattering_channels** [53,54]. This calculates the relaxation time as

$$\tau_{im}(E,T) = \frac{E^{\frac{3}{2}}\sqrt{2m^*}4\pi\epsilon^2}{\left(log\left(1+\frac{1}{x}\right)-\frac{1}{1+x}\right)\pi n_I Z_I^2 e^4} \tag{4.6}$$

$$x = \frac{E}{k_B T} \tag{4.7}$$

Required parameters are m^*, ϵ_0, ϵ_{inf}, n_I, Z_I.

4.5.2 Acoustic Scattering

In order to include the effect of electron scattering from acoustic phonons, include `'acoustic'` in the list **scattering_channels**. This calculates the relaxation time according to [53,54]

$$\tau_{ac}(E,T) = \frac{2\pi\hbar^4\rho v^2}{(2m^*)^{\frac{3}{2}}k_B T D_{ac}^2\sqrt{E}} \tag{4.8}$$

Required parameters are m^*, ρ, v, D_{ac}.

4.5.3 Optical Scattering

In order to include the effect of electron scattering from optical phonons, include `'optical'` in the list **scattering_channels**. This calculates the relaxation time according to [53,54]

$$\tau_{op}(E,T) = \frac{\sqrt{2k_B T}\pi x_0 \hbar^2}{m^{*\frac{3}{2}}D_{op}^2 N_{op}\sqrt{x+x_0}+\left(N_{op}+1\right)\sqrt{x-x_0}} \tag{4.9}$$

$$N_{op} = \frac{1}{exp\left(\frac{\hbar\omega_l}{k_B T}\right)-1},\ x = \frac{E}{k_B T},\ x = \frac{\hbar\omega_l}{k_B T} \tag{4.10}$$

Required parameters are m^*, ρ, ω_l, D_{op}, N_{op}.

4.5.4 Polar Acoustic Scattering

In order to include the effect of electron scattering from acoustic phonons in polar materials, include `'polar_acoustic'` in the list **scattering_channels**. This calculates the relaxation time according to [53,54]

$$\tau_{pac}(E,T) = \frac{\sqrt{2E}2\pi\epsilon^2\hbar^2\rho v^2}{p^2e^2\sqrt{m^*}k_BT}\left[1 - \frac{\epsilon_0}{2E}log\left(1+4\frac{E}{\epsilon_0}\right) + \frac{1}{1+4\frac{E}{\epsilon_0}}\right] \tag{4.11}$$

Required parameters are m^*, ϵ_0, ϵ_{inf}, ρ, v, p.

4.5.5 Polar Optical Scattering

In order to include the effect of electron scattering from optical phonons in polar materials, include `'polar_optical'` in the list **scattering_channels** [53-55].

$$\tau_{op}(E,T) = \sum_i \frac{Z\left(E,T,\omega_i^l\right)E^{\frac{3}{2}}}{C\left(E,T,\omega_i^l\right) - A\left(E,T,\omega_i^l\right) - B\left(E,T,\omega_i^l\right)} \tag{4.12}$$

$$A(E,T,\omega_l) = n(\omega_l+1)\frac{f_0(E+\hbar\omega_l)}{f_0(E)} \times \left[(2E+\hbar\omega_l)sinh^{-1}\left(\frac{E}{\hbar\omega_l}\right)^{\frac{1}{2}} - [E(E+\hbar\omega_l)]^{\frac{1}{2}}\right] \tag{4.13}$$

$$B(E,T,\omega_l) = \theta(E-\hbar\omega_l)n(\omega_l)\frac{f_0(E-\hbar\omega_l)}{f_0(E)} \times \left[(2E-\hbar\omega_l)cosh^{-1}\left(\frac{E}{\hbar\omega_l}\right)^{\frac{1}{2}} - [E(E-\hbar\omega_l)]^{\frac{1}{2}}\right] \tag{4.14}$$

$$C(E,T,\omega_l) = 2E\left[n(\omega_l+1)\frac{f_0(E+\hbar\omega_l)}{f_0(E)}sinh^{-1}\right. \tag{4.15}$$

$$\left. + \theta(E-\hbar\omega_l)n(\omega_l)\frac{f_0(E-\hbar\omega_l)}{f_0(E)}cosh^{-1}\left(\frac{E}{\hbar\omega_l}\right)^{\frac{1}{2}}\right]$$

$$Z(\omega_l) = \frac{2}{W_0(\hbar\omega_l)^{\frac{1}{2}}},\ W_0(\omega_l) = \frac{e^2\sqrt{2m^*\omega_l}\epsilon^{-1}}{4\pi\hbar^{\frac{3}{2}}} \tag{4.16}$$

Required parameters are m^*, ϵ_0, ϵ_{inf}, ω_{LO}

Table 4.1: Symbol, units, and corresponding key in tau_dict for the parameters required in various scattering models.

Parameter	Symbol	Units	Key
Mass density	ρ	kg/m^3	rho
Low frequency dielectric constant	ϵ_0	-	eps_0
High frequency dielectric constant	ϵ_∞	–	eps_inf
Acoustic velocity	v	m/s	v
Effective mass ratio	m^*	-	ms
Acoustic deformation potential	D_{ac}	eV	D_ac
Optical deformation potential	D_{op}	eV	D_op
Optical phonon energy	$\hbar\omega_l$	eV	hwlo
Number of impurities	n_I	cm^{-3}	nI
Charge on impurity	Z_I	-	Zi
Piezoelectric constant	p	C/m^2	piezo

4.5.6 Effective Scattering Time

$$\frac{1}{\tau_{total}(E,T)} = \frac{1}{\tau_{im}(E,T)} + \frac{1}{\tau_{ac}(E,T)} + \cdots \tag{4.17}$$

The total scattering time is calculated as a harmonic sum of the specified scattering mechanisms, evaluated for each energy and temperature, equation 4.17. The effective

scattering time τ_{total} is computed for each energy and temperature during the execution of the `transport` routine.

4.6 Performance

Benchmark performance tests reveal excellent scaling for massively parallelized calculations. PAOFLOW exploits parallelization over bands whenever possible, primarily in the calculation of gradients. However, most routines are parallelized across the k-point mesh or path. PAOFLOW also possesses excellent scaling of memory requirements, in parallel runs. Increasing the number of processors used can reduce the memory load on each processor, as many of the large arrays are distributed evenly among the cores. Performance is analyzed on a Dell PowerEdge R730 server with two 2.4GHz Intel Xeon E5-2680 v4 fourteen-core processors, and results for several examples are presented in figures 4.4 and 4.5. PAOFLOW demonstrates excellent scaling properties on manycore systems and possesses massively parallel capabilities.

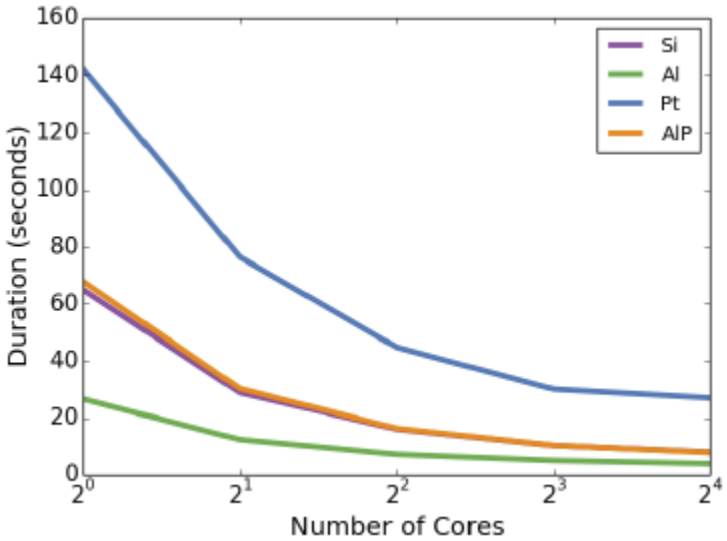


Figure 4.4: Select examples (see `examples/` on GitHub) performed on an increasing number of processors. Parallelized routines provide run time scaling nearly proportional to the number of cores used in a calculation, closely approaching the speed increase limit of Amdahl's Law.

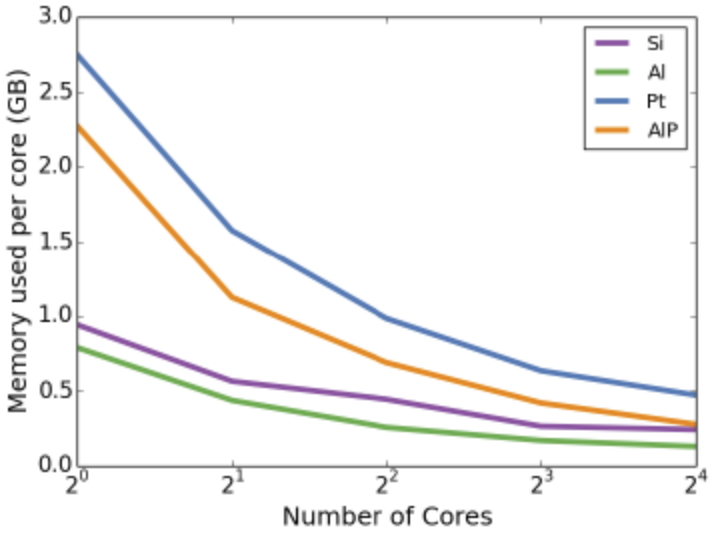


Figure 4.5: Memory scaling per core (in GB), for select examples. An increasing core count reduces the memory requirements per processor.

4.7 Conclusion

PAOFLOW provides a lightweight, robust tool for efficient materials and Hamiltonian analysis. Continuous development of the package has streamlined its functionality and enabled many new tools for effectively characterizing the electronic properties of solids. The updated framework offers an ideal tool for high throughput condensed matter simulation and generation for materials genomics [56].

4.8 Acknowledgments

We are grateful for computational resources provided by the High Performance Computing Center at the University of North Texas and the Texas Advanced Computing Center at the University of Texas, Austin. The members of the AFLOW Consortium (https://www.aflow.org) acknowledge support by DOD-ONR (N00014-13-1-0635, N00014-11-1-0136, N00014-15-1-2863). The authors also acknowledge Duke University Center for Materials Genomics.

CHAPTER 5

GIANT SPIN HALL EFFECT IN TWO-DIMENSIONAL MONOCHALCOGENIDES*

One of the most exciting properties of two dimensional materials is their sensitivity to external tuning of the electronic properties, for example via electric field or strain. Recently discovered analogues of phosphorene, group-IV monochalcogenides (MX with M = Ge, Sn and X = S, Se, Te), display several interesting phenomena intimately related to the in-plane strain, such as giant piezoelectricity and multiferroicity, which combine ferroelastic and ferroelectric properties. Here, using calculations from first principles, we reveal for the first time giant intrinsic spin Hall conductivities (SHC) in these materials. In particular, we show that the SHC resonances can be easily tuned by combination of strain and doping, and, in some cases, strain can be used to induce semiconductor to semimetal transitions that makes a giant spin Hall effect possible even in absence of doping. Our results indicate a new route for the design of highly tunable spintronics devices based on two-dimensional materials.

5.1 Introduction

The spin Hall effect (SHE) is a phenomenon emerging from spin-orbit coupling (SOC) in which an electric current or external electric field can induce a transverse spin current resulting in spin accumulation at opposite sample boundaries [57-60]. The charge/spin conversion without the need for applied magnetic fields makes the SHE an essential tool for spin manipulation in any spintronics device [61,62] and the subject of intensive theoretical and

* This chapter is reproduced, with permission from IOP Science, from the paper: *Giant spin Hall effect in two-dimensional monochalcogenides* by Jagoda Sławińska, Frank T. Cerasoli, Haihang Wang, Sara Postorino, Andrew Supka, Stefano Curtarolo, Marco Fornari, and Marco Buongiorno Nardelli, published in 2D Materials 6, 025012 (2019).

experimental research. The intrinsic SHE in crystals was predicted and observed experimentally in a variety of materials, ranging from doped semiconductors (GaAs) [58] to elemental metals with strong SOC, such as platinum, tantalum, palladium, and tungsten [28,64-68]. It has been also investigated in metallic and semimetallic thin films [69] where SHE can be enhanced with respect to the corresponding bulk phase. Studies related to SHE in two dimensional (2D) materials are limited to only few works focused on transition metal dichalcogenides [70] and simple material models [71].

In this paper, a giant intrinsic SHE tunable by combination of strain and doping is predicted for the first time in monolayer group-IV monochalcogenides MX with M = Ge, Sn and X = S, Se, Te, often referred to as analogues of phosphorene due to their structural similarity [72-74]. The bulk parent compound has the orthorhombic crystal structure of black phosphorous (*Pnma*) and consists of weakly bonded van der Waals layers, making an exfoliation process a viable routine to produce atomically thin films or single layer crystals; indeed, some materials from this family have been already synthesized experimentally [75-79]. As do most 2D materials, group-IV monochalcogenides exhibit several extraordinary mechanical, electronic, and optical properties. These include high flexibility, large thermal conductivity, giant piezoelectricity [80], multiferroicity [81,82], superior optical absorbance [83-85], and even valley Hall effects [86], making them promising candidates to use in multifunctional devices. Finally, the strong SOC suggests their high potential for spintronics.

The group-IV monochalcogenides possess wide band gaps, which precludes existence of non-negligible spin Hall conductivity (SHC) at intrinsic chemical potential, similar to conventional bulk semiconductors where significant electron or hole doping is needed to

achieve a measurable spin Hall effect. Although our calculations show that a giant SHC could be reached with *p*- or *n*- type doping of $n_{h/e} = 1 \times 10^{14} \frac{e}{cm^2}$, which is an order of magnitude lower than in case of transition metal dichalcogenides [70], such values may still be difficult to reach experimentally. Here, we propose an alternative route to realize SHE in these materials. We demonstrate that compressive or tensile strain along any axis not only can tune the position of the SHC resonances, but can also induce semiconductor to metal transitions that make a giant spin Hall effect possible even in absence of doping. As such, different phases of SHE can be switched externally via strain allowing direct engineering of spintronics functionalities in these materials.

5.2 Methods

Our noncollinear DFT calculations were performed using the QUANTUM ESPRESSO code [8,9] interfaced with the AFLOWπ and PAOFLOW infrastructures [22]. We used the generalized gradient approximation (GGA) in the parameterization of Perdew, Burke, and Ernzerhof (PBE) [87] and, to further improve the description of the electronic properties, a novel pseudo-hybrid Hubbard self-consistent approach ACBN0 [30]. The ion-electron interaction was treated with the projector augmented wave fully-relativistic pseudopotentials [88] from pslibrary database [89] while the wavefunctions were expanded in a plane-wave basis of 50 Ry (500 Ry for the charge density). The Brillouin zone sampling at DFT level was performed following the Monkhorst-Pack scheme using a $24 \times 24 \times 1$ k-points grid, further increased to $140 \times 140 \times 1$ with PAOFLOW's Fourier interpolation method to accurately integrate spin Berry curvatures.

The intrinsic spin Hall conductivities were calculated using the PAOFLOW code following the linear response Kubo-like formula [25,37,90]:

$$\sigma_{ij}^{s} = \frac{e^2}{\hbar}\sum_{\vec{k}}\sum_{n} f_n(\vec{k})\Omega_{n,ij}^{s}(\vec{k}) \tag{5.1}$$

$$\Omega_{n,ij}^{s}(\vec{k}) = \sum_{m\neq n}\frac{2\mathrm{Im}\left\langle\psi_{n,\vec{k}}\middle|\vec{j_i^s}\middle|\psi_{m,\vec{k}}\right\rangle\left\langle\psi_{m,\vec{k}}\middle|v_j\middle|\psi_{n,\vec{k}}\right\rangle}{(E_n - E_m)^2} \tag{5.2}$$

where $\vec{j} = \{s, \vec{v}\}$ is the spin current operator with $s = \frac{\hbar}{2}\beta\Sigma(\beta, \Sigma : 4\times 4$ Dirac matrices) and $f_n(\vec{k})$ is the Fermi distribution function for the band n at $\vec{k}$. We note that, in contrast to most reported calculations of SHC based on the above formula [25,63,70], we do not add any infinitesimal term δ in the denominator to avoid singularities if the bands are degenerate. We have evidence (see figure S1 in supplementary material (SM) (stacks.iop.org/TDM/6/025012/mmedia)) that using a finite δ in Kubo's formula leads to unphysical behavior of non-zero values of SHC within the semiconductor's gap whos origin was unclear so far. Using perturbation theory for degenerate states to avoid numerical singularities ensure that σ_{ij}^{s} always vanishes at the Fermi level.

Figure 5.1 shows the 2D non-centrosymmetric unit cell of the phosphorene-like phase used in the calculations for all six compounds. It contains for atoms arranged in two buckled layers resembling a monolayer of black phosphorous with a mirror symmetry axis along x, representing one of the four ground states of the system [81,82,91]. The lattice constants and ionic positions were fully relaxed without including SOC whose influence on forces is known to be negligible. The electronic structure was then recalculated with SOC self-consistently. The vacuum region of 20 Å was set to prevent any interaction between spurious replicas of the slab. The configurations with the relaxed lattice constants were used as a starting point for the simulations of strained structures; we considered strains varying between -10% and 10%

along x and y axes simultaneously, in each case relaxing the positions of the atoms. We analyzed, in total, 726 different structures. Further details of the calculations as well as additional results, including lattice constants, values of band gaps, and convergence tests for SHC are reported in the SM (table S1, figure S1).

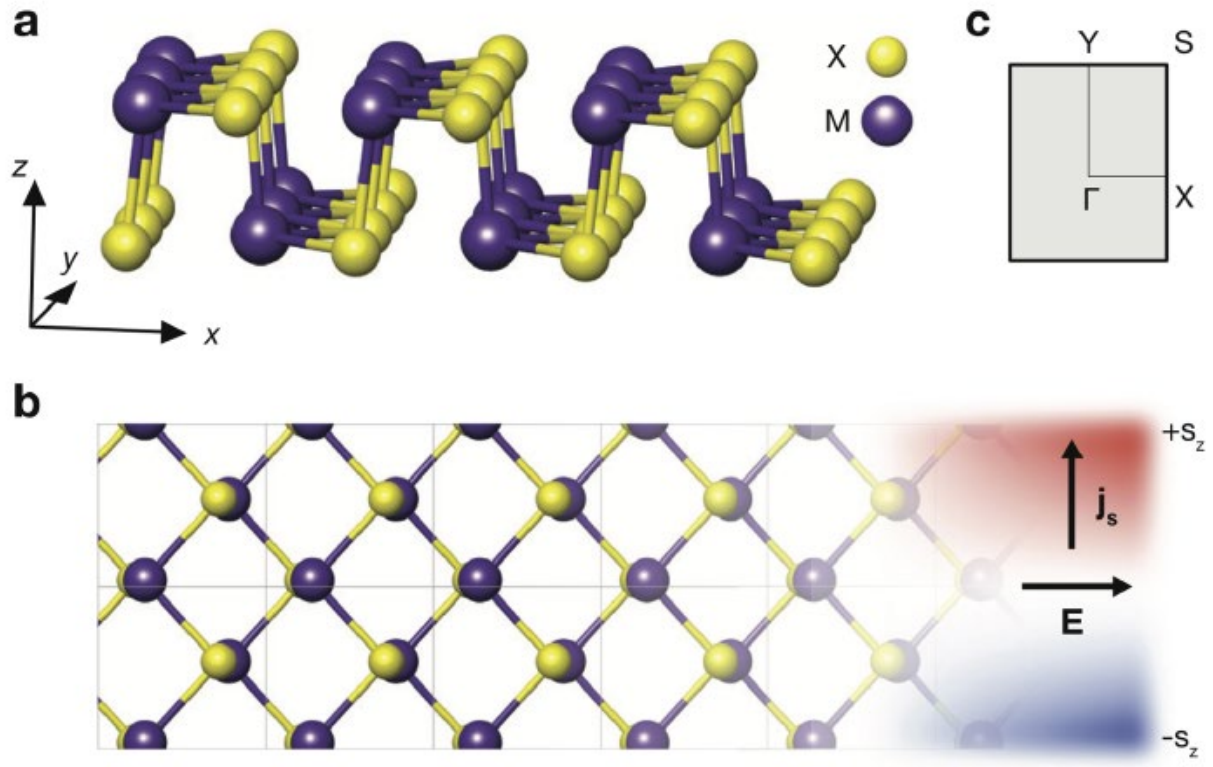


Figure 5.1: Structure of 2D group IV monochalcogenides. Panels (a) and (b) show side and top views, respectively, while the scheme of Brillouin Zone is displayed in (c). Right hand side of panel (b) additionally illustrates an example of geometry setup for spin Hall effect possible to realize in doped monolayers. We note that only σ_{xy}^{z} and σ_{yz}^{z} components of SHC tensory are different from zero.

5.3 Spin Hall Effect in Unstrained Monolayers

Let us first consider unstrained structures. Figures 5.2(a) and (b) summarize the relativistic band structures of sulfides, selenides, and tellurides (green, blue, and red lines, respectively), while panels (c) and (d) display their corresponding spin Hall conductivity as a function of chemical potential. Due to the reduced symmetry of two-dimensional structures,

the only non-vanishing independent component is $\sigma_{xy}^{z} = -\sigma_{yx}^{z}$. Our band structures and the values of the band gaps calculated with the ACBN0 functional are in good agreement with existing simulations that use hybrid functionals and with available experimental data (see SM, table S1). Comparison of the scalar relativistic band structures (plotted as black lines in panels (a) and (b)) with the fully relativistic ones, clearly indicates a strong impact of SOC, which indicates several anti-crossings and splittings of the bands. These effects are moderate in selenides and most pronounced in tellurides given the heaviness of Te atoms. Although it is quite difficult to attribute particular features of the relativistic band structures to the specific peaks (resonances) in $\sigma_{xy}^{z}(E)$, one can easily observe that severe SOC-induced modifications in the electronic structure of GeTe and SnTe result in giant values of spin Hall conductivity ~300 and 500 $\frac{\hbar}{e}(\Omega \times cm)^{-1}$, respectively, for higher binding energies. We note that even the resonances closest to the Fermi level (E_F) still achieve values as large as 200 $\frac{\hbar}{e}(\Omega \times cm)^{-1}$. The selenides exhibit slightly lower (~100) magnitudes of SHC, and the sulfides do not seem to display any spin Hall effect at all.

As we have mentioned above, similarly to other semiconductors, either *p*-type or *n*-type doping is needed to reach the SHC resonances. In table 5.1, we list the values of the SHC peaks and the corresponding doping levels expressed as a Fermi level shift and number of electron per surface unit, for six compounds reported in figure 5.2. In general, the doping concentrations are of the order of $n_{h/e} = 1 \times 10^{14} \frac{e}{cm^2}$, an order of magnitude lower than in the case of recently studied transition metal dichalcogenides [70] but still beyond the typical values achieved in experiments ($\sim 10^{12} - 10^{13} \frac{e}{cm^2}$). However, for the compounds with highest SHC

peaks, the spin Hall effect could be very large even at lower doping: for instance, in SnTe the SHC reaches 100 $\frac{\hbar}{e}(\Omega \times cm)^{-1}$ for doping of $\sim 10^{13}\,\frac{e}{cm^2}$. We also note that the estimated values of doping listed in table 5.1 are simply derived from the density of states following similar analysis in previous theoretical works dealing with SHE in semiconductors [70] and the carrier concentration in real samples might be different. Therefore, we believe that the intrinsic spin Hall effect could be achieved experimentally even in the unstrained structures.

5.4 Spin Hall Effect in Strained Monochalcogenides

As a next step, we have explored the possibility of tuning the electronic properties and the spin Hall conductivity via external strain. The band gap (E_g) manipulation in group-IV sulfides and selenides was previously reported in [74], where compressive strains along either x or y axes were found to strongly reduce the band gap, and for larger strains could induce a semiconductor to semimetal transition. In the present study, we consider all possible strain configurations varying between -10% and $+10\%$ with the step of 2%, thus 121 different configurations for each compound. Below, we will discuss only the results for SnTe which displays the highest potential for spintronics; the complete set of results for all considered group-IV monochalcogenides are reported in the SM (figures S3-S7).

5.4.1 SHE in the Metallic Phase

Figure 5.2 summarizes the electronic properties and spin Hall conductivity in SnTe for each considered strain configuration. The band gap landscape displayed in panel (a) shows several interesting features: (i) compressive (tensile) strain always leads to decrease (increase) in E_g, (ii) the lowest values of E_g are achieved when a compressive strain along *only one* axis is

applied, (iii) the combinations of tensile and compressive strain can also lead to a decrease in E_g, larger than in the case of compressive strains applied along both axes. We have observed similar behavior in all compounds; in selenides the bad gaps are in general wider, thus larger strains are required to enable the semiconductor to semimetal transition. In sulfides, in contrast, a metallic phase cannot be achieved, in agreement with conclusions of [74].

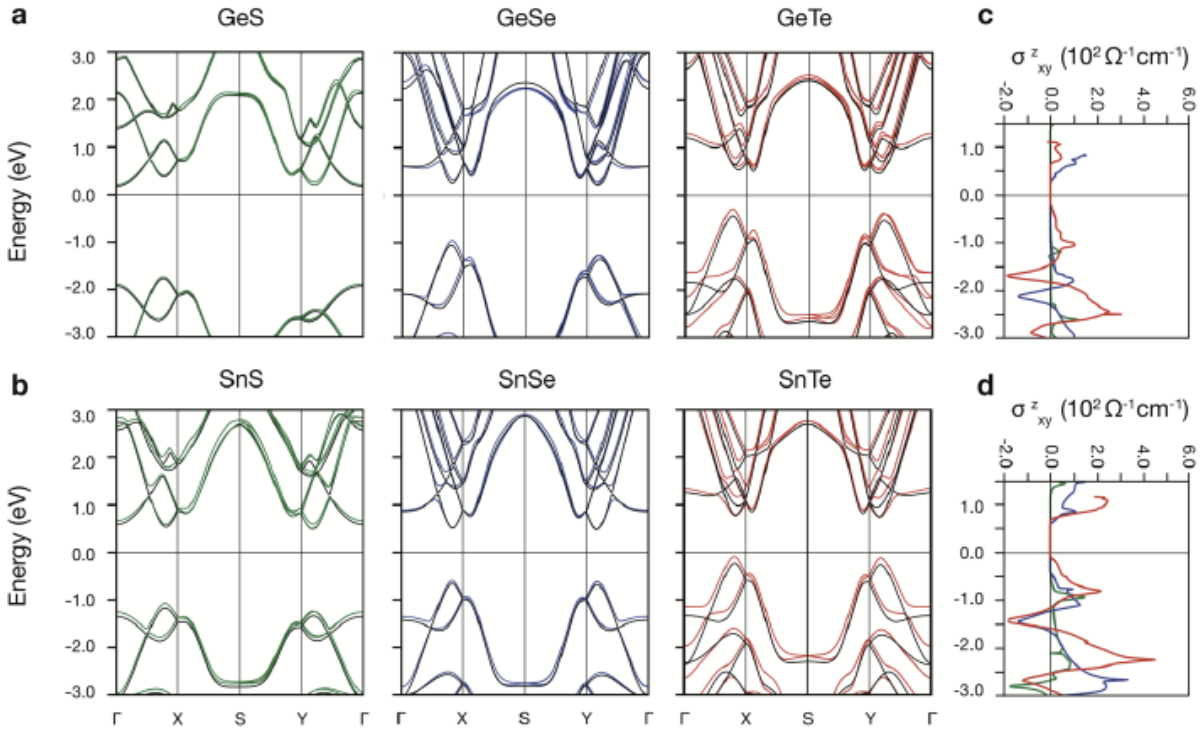


Figure 5.2: Relativistic electronic structures of group IV monochalcogenides GeX (a) and SnX (b), X = S, Se, Te represented as green, blue, and red lines, respectively. The corresponding scalar-relativistic band structures are superimposed (black lines). ((c) and (d)) Spin Hall conductivities σ_{xy}^{z} calculated as a function of chemical potential for compounds in panels ((a) and (b)) employing the same color scheme.

The corresponding spin Hall conductivities calculated at intrinsic chemical potential plotted in panel (b) clearly reflect the profile in (a), that is, as long as the material is semiconducting, it cannot exhibit any spin Hall effect. Within the regions of $E_g = 0$, the values of σ_{xy}^{z} vary because each of these strain configurations induces different modifications in the

electronic structure. However, we are still able to draw general conclusions regarding the impact of the strain based on the analysis of few selected configurations A, B, and C displayed in figure 5.2(c) which embody rather huge modifications of both electronic and spin properties. First of all, it is clear that the same stain applied along axis x and y (configurations A and B) affect the dispersion of the bands in an asymmetric way. The strain along x (panel A), in general, brings upwards the occupied bands along the S—Y—Γ paths and downwards the unoccupied spectrum along Γ—X—S resulting in *p*-type pockets near Y and *n*-type pockets near X, while y-strain (panel B) is found to cause opposite shifts. The band structure of configuration C confirms that such tendency is more general; this lower strain configuration is structurally similar to B and indeed its electronic properties are very similar to the latter.

The spin Hall conductivities of structures A, B, and C shown in figure 5.2(d) significantly differ from those of the unstrained structure in figure 5.2(d), which is not surprising, given the substantial modification of electronic structure at the Fermi level. In order to facilitate a systematic analysis, we have introduced the labels E_1 and E_2 corresponding to the two SHC resonances below and above E_F, and we followed their behavior due to the electronic structure changes induced by the strain (the positions of E_1 and E_2 without strains are reported in table 5.1).

It is evident that the σ_{xy}^{z} can be giant without any doping (structures A and B), which we attribute to the presence of resonance E_2 much closer to the Fermi level than in the unstrained monolayer (E_F is located on the slope in both A and B configurations). Statistical analysis of these two parameters for all calculated structures confirms that indeed the change in the band gap is mainly correlated with the resonance E_2 (calculated Pearson's correlation between E_g

and E_2 is around 80%), while the position of E_1 hardly depends on the band gap. This means that strain is more likely to shift/modify unoccupied bands which will also determine the SHC. Finally, the σ_{xy}^{z} in the moderately strained configuration C also exhibits a finite value at the Fermi level, but since the peak E_2 is not so close to E_F, the spin Hall effect is weaker.

5.4.2 Tuning of SHC via Strain and Doping

The experimental realization of SHC, its tuning and switching on/off, is likely to require a combination of doping and strain. In order to quantitatively estimate the effect of both, we have calculated the values of σ_{xy}^{z} averaged over the range of chemical potential that correspond to a given electron/hole concentration for every configuration of strain. The results for SnTe are shown in figure 5.3. In accordance with table 5.1, *n*-type doping of $\sim 10^{14}\,\frac{e}{cm^2}$ (a) guarantees giant values of SHC for unstrained and weakly strained structures. Surprisingly, a comprehensive strain (a) leads to the reduction of SHC rather than to its increase, which is related to the oscillating character of the intrinsic spin Hall conductivity as a function of chemical potential. In this case, the combination of doping and compressive strain shifts the Fermi level excessively, well beyond the resonance peak of SHC. Moreover, it is clear that a biaxial strain could be used to switch on/off the SHC. We can observe similar behavior also for *n*-type doping of $\sim 10^{13}\,\frac{e}{cm^2}$ (b), in such case the values of SHC are lower, but still can be considered large even for unstrained/weakly strained structures. For $n_e \sim 10^{12}\,\frac{e}{cm^2}$ (c) the finite but small values of SHC can be increased by uniaxial compressive strain; this low doping configuration resembles the properties of the undoped structures (d) and weakly *p*-type doped configuration (e). Further increase in *p*-type doping to $n_h \sim 10^{13}\,\frac{e}{cm^2}$ (f) offers a possibilitiy of

even broader modulation of SHC by strain; while biaxial compressive strain can result in switching off the SHC, uniaxial strain leads to its increase. Overall, however, the values of SHC are lower here than in the case of electron doping. Thus, as anticipated in the previous section, the manipulation of SHE seems to be more feasible in *n*-type doped systems. Finally, among the other monochalcogenides, GeTe reveals interesting properties for spintronics, while the sulfides and selenides either do not possess large SHC at all or do not exhibit sufficiently high tunability (see SM, figures S4-S7).

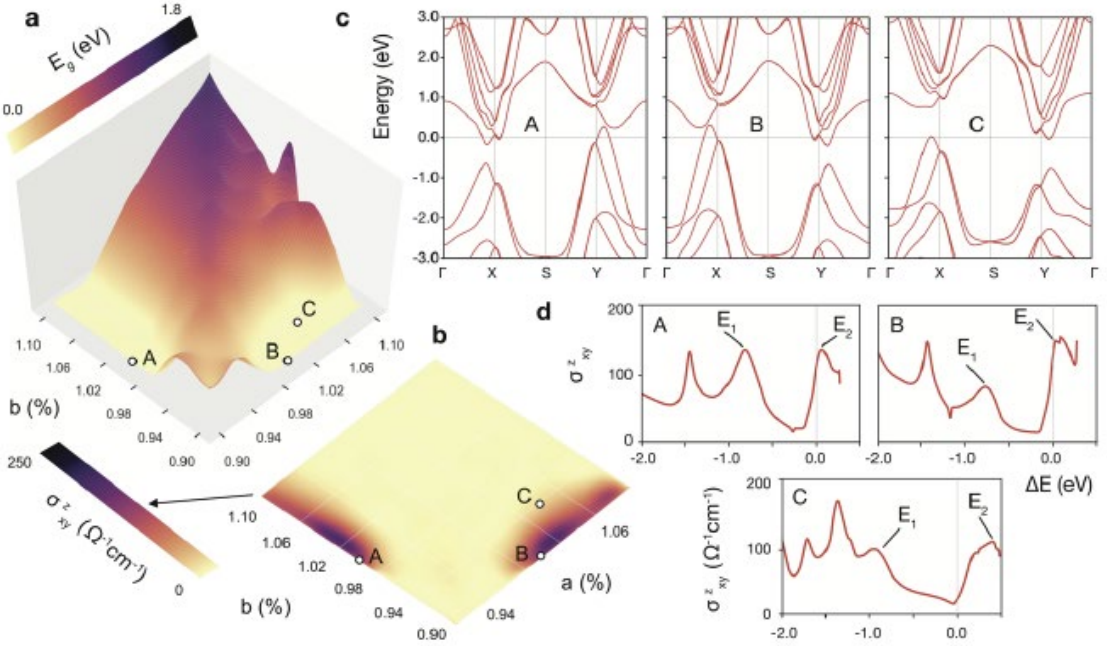


Figure 5.3: Electronic properties and spin Hall conductivities of SnTe as a function of strain. (a) Band gap E_g versus strain a (along x axis) and b (along y axis) displayed as a 3D surface in lattice constants space. The corresponding legend is shown in the upper left corner. (b) Heat-map of spin-Hall conductivities calculated at zero chemical potential for each strain configuration. The legend is displayed in the bottom left corner. The reversed contrast of maps (a) and (b) clearly reflects their physical meaning and the fact that σ^z_{xy} can have a finite value only for $E_g = 0$. Since uniaxial strain is often considered in experimental realizations, we have also introduced complementary plots of summarizing the influence of uniaxial strains along x and y on the band gap and the SHC (see SM, figure S2). (c) Band structures of SnTe calculated for selected points in strain space marked in ((a) and (b)), A: $a = 0.9a_0$, $b = 1.0b_0$, B: $a = 1.0a_0$, $b = 0.94b_0$, C: $a = 1.04a_0$, $b = 0.94b_0$, where a_0 and b_0 denote original (unstrained) lattice constants. (d) Corresponding σ^z_{xy} calculated as a function of chemical potential. Labels E_1 and E_2 at each curve denote resonances of SHC closest to the Fermi level.

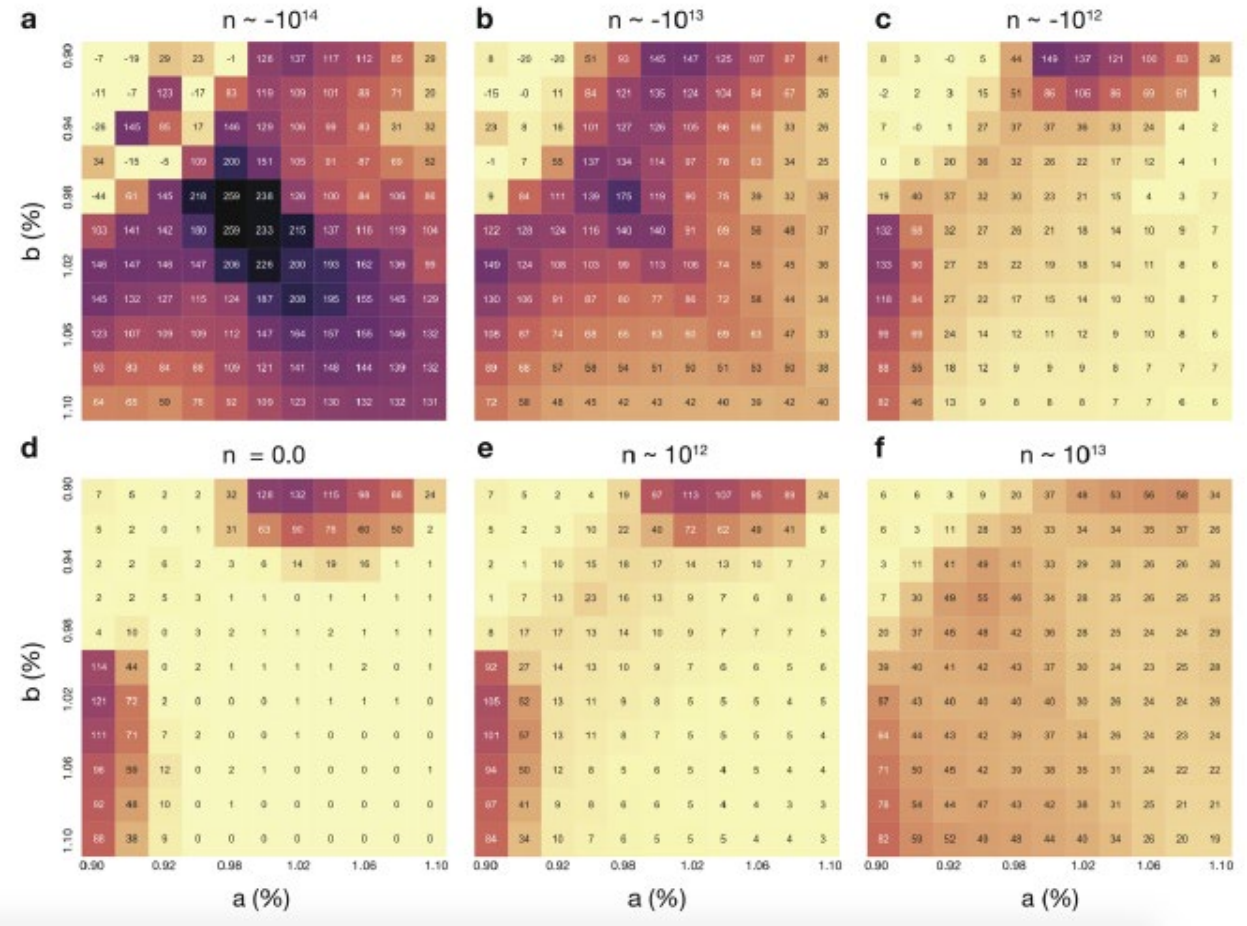


Figure 5.4: Spin Hall conductivity for SnTe as a function of strain and doping. (a) σ_{xy}^{z} averaged over the values of chemical potential which correspond to doping of order $\sim 10^{14}$e cm^{-2} displayed as a function of strains along x and y. ((b) and (c)) Same as (a) for $n_e = 10^{13}$ e cm^{-2} and $n_e = 10^{12}$ e cm^{-2}, respectively. (d) σ_{xy}^{z} at intrinsic chemical potential, numerically identical to the map in figure 5.3(b). Non-zero values for the regions of $E_g \neq 0$ are related to the numerical accuracy of calculations. (e) and (f) Same as ((b) and (c)) for p-type doping. Color scheme same as in figure 5.3(b) in all the maps.

5.5 Summary and Conclusions

In summary, we have reported for the first time the emergence of a giant spin Hall effect in group-IV monochalcogenides, which can be switched on/off and modulated either by doping or uniaxial compressive strain. The most interesting candidate for spintronics is SnTe. We have predicted that the SHE in this compound can be very strong. Moreover, the monolayer and multilayer samples have been recently synthesized and they reveal high potential for

technology [78,79,92]. While our work is limited to monolayers structurally similar to black phosphorous, it is worthwhile to mention that the multilayer stacks can exhibit more intriguing spin-orbit related properties and with even broader possibilities of tuning. Finally, SHE has been achieved in β-SnTe which suggests that its successful realization in 2D phase is very probable [93]. Despite the values of doping/strain required to reach/tune the giant SHC seeming large, the actual 2D character of these compounds can greatly help to overcome these difficulties. For example, different stacking order or thickness of the multilayer might reduce required doping. Also, the Fermi level could be additionally shifted by any charge originating from a substrate [94], or any additional strain could result from lattice constant matching at the interface.

Finally, we emphasize that 2D spintronics are not just a hypothesis; the recent discovery of 2D ferromagnetism [95,96], brings it much closer to realization and new candidate materials will be needed. The giant SHC of monochalcogenides combined with low charge conductivity, even in the metallic phase, suggest they might be useful in versatile spintronics applications, such as spin detectors [62], and even more likely in novel multifunctional devices. We believe that the properties unveiled in this paper clearly show the high potential of 2D phosphorene analogues for spintronics and will trigger interest in the experimental realization of the spin Hall effect in these materials.

5.6 Acknowledgments

We would like to thank Ilaria Siloi and Priya Gopal, for useful discussions. The members of the AFLOW Consortium (www.aflow.org) acknowledge support by DOD-ONR (N00014-13-1-0635, N00014-11-1-0136, N00014-15-1-2863). MBN and FTC acknowledge partial support from Clarkson Aerospace Corporation. The authors also acknowledge Duke University—Center for

Materials Genomics—and the CRAY corporation for computational support. Finally, we are grateful to the High Performance Computing Center at the University of North Texas and the Texas Advanced Computing Center at the University of Texas, Austin.

CHAPTER 6

ULTRATHIN SnTe FILMS AS A ROUTE TOWARD ALL-IN-ONE SPINTRONICS DEVICES*

Spin transistors based on a semiconducting channel attached to ferromagnetic electrodes suffer from fast spin decay and extremely low spin injection/detection efficiencies. Here, we propose an alternative all-in-one spin device whose operation principle relies on electric manipulation of the spin lifetime in two-dimensional (2D) SnTe, in which the sizable spin Hall effect eliminates the need for using ferromagnets. In particular, we explore the persistent spin texture (PST) intrinsically present in the ferroelectric phase which protects the spin from decoherence and supports extraordinarily long spin lifetime. Our first-principles calculations followed by symmetry arguments revealed that such a spin wave mode can be externally detuned by perpendicular electric fields, leading to spin randomization and decrease in spin lifetime. We further extend our analysis to ultrathin SnTe films and confirms the emergence of PST as well as moderate enhancement of intrinsic spin Hall conductivity. The recent room-temperature observation of the ferroelectric phase in 2D-SnTe suggests that novel all-electric spintronics devices are within reach.

6.1 Introduction

The idea of using electron spins in transistors for information transfer and processing lies at the heart of research in the area of spintronics [97]. However, after two decades of efforts the pioneering concept of spin transistors, proposed by Datta and Das, still suffers from

* This chapter is reproduced, with permission from IOP Science, from the paper: Ultrathin SnTe films as a route toward all-in-one spintronic devices, by Jagoda Sławińska, Frank T. Cerasoli, Priya Gopal, Marcio Costa, Stefano Curtarolo, and Marco Buongiorno Nardelli, published in 2D Materials 7, 025026 (2020).

two major performance issues impeding its use in applications [98,99]. First is the low efficiency of spin injection and detection through ferromagnets caused by the conductivity mismatch at the interface. Second is the two-faceted nature of spin-orbit interaction; it enables spin manipulation along the channel, but it is adverse for spin lifetime and leads to spin randomization in the diffusive transport regime. Several approaches were proposed to overcome these obstacles, including all-electric spin Hall transistors without the ferromagnets [100,101] or devices protected from spin decoherence by uniform spin configuration known as a persistent spin helix (PSH) [102,103]. However, these successful realizations rely on precisely controlled structures, such as semiconductor quantum wells [104-106], which usually limits the operating temperatures to a few kelvin, preventing any practical use.

In parallel with the progress in spin transistors, several multifunctional materials have been recently designed or rediscovered; some of them reveal intriguing quantum phenomena intimately related to dimensionality, topology, and symmetries. Group IV-VI monochalcogenides (MX, M=Ge, Sn; X=S, Se, Te) are narrow gap semiconductors widely used in conventional devices where they serve as thermoelectrics, ferroelectrics, optical filters and detectors, photocatalysts and sensors. Remarkably, their intriguing spin-dependent electronic properties remained unexplored over decades; GeTe and SnTe have been just recently recognized as excellent candidates to use in spintronics. In particular, they represent a class of so-called ferroelectric Rashba semiconductors (FERSC) whereby the spin degree of freedom coupled to the ferroelectricity manifests in the electrically tunable Rashba spin texture of bulk electronic states [107-112]. Moreover, sizable spin Hall effect (SHE) has been proposed in both materials [93,113], which opens a perspective to integrate different functionalities and

construct ferromagnet-free spin devices.

In this paper, we put forward the idea of all-in-one spin transistor based on two-dimensional (2D) SnTe. In such a device the spin injection and detection can be accomplished via direct and inverse spin Hall effects, while the on/off state is manipulated through the electric control of spin lifetime along the channel. Specifically, an atomically-thick SnTe in the structure form of phosphorene was suggested to host a persistent spin wave mode enforced by the crystal space group symmetry [114]. Uniform spin polarization along the out-of-plane direction is linked to the ferroelectricity, or more precisely to the in-plane polar displacement. As an intrinsic property of the material, it does not require any fine-tuning between spin-orbit parameters in order to support an exceptionally long spin lifetime [115]. Here, we propose that such a spin configuration can be externally detuned by perpendicular electric field breaking the crystal symmetries. The spins are then dephased by electron scattering, which enables the realization of the switch-off mechanism. Importantly, the injected spins are by construction parallel to the spin-orbit field, thus they will be transported without precession, making the device robust against switching between different momenta and sub-bands with opposite spin textures in the presence of doping.

Finally, the ferroelectric ground state has been recently observed in mono- and multi-layer SnTe structures and, surprisingly, it seems to persist in the latter at room temperature [78]. Our density functional theory (DFT) calculations confirmed that the SHE/PSH combination can be realized in specific ultrathin SnTe films. In light of the above, the all-in-one spin transistors are an attractive hypothesis, further discussed in terms of phase robustness and alternative realizations.

6.2 SHC and PSH in 2D SnTe

Let us start with a brief overview of geometry, electronic properties, and intrinsic spin Hall conductivity calculated for the SnTe monolayer (1ML-SnTe; in our notation, we refer to van der Waals monolayer equivalent to two atomic layers – 2AL [79]). Figure 6.1 (a)-(b) shows the orthorhombic lattice of the crystal. The ions are arranged in two buckled layers with distortions inducing the spontaneous polarization along x, while the tiny anisotropy between the lattice constants a and b emerges as a natural signature of the ferroelectric phase. Two-dimensional SnTe belongs to the space group number 31 (*Pmn2$_1$*) invariant with respect to the following symmetry operations: (1) the identity operation E, (2) mirror reflection M_y with respect to the $y = 0$ plane, (3) glide reflection $\overline{M_z}$ consisting of mirror reflection M_z about the $z = 0$ plane followed by a fractional translation by a vector $\vec{v} = \left(\frac{1}{2a}, \frac{1}{2b}, 0\right)$, and (4) two-fold screw rotation $\overline{C_{2x}}$ combing two-fold rotation around the x-axis with a translation of $\vec{v}$. Clearly, these symmetries suggest the intrinsic PSH in terms of the criteria formulated buy Tao and Tsymbal [115]. Following their general arguments based on group theory, we can rationalize the full (scalar) relativistic electronic structure of SnTe represented by red (black) lines in figure 6.1(b). In particular, the energy levels without the spin-orbit coupling (SOC) are fourfold degenerate, while the SOC splits them into doublets with eigenvalues of $\overline{M_z} = \pm 1$ over the entire BZ except for the Γ—X line. Furthermore, these doublets split into singlets with eigenvalues of spin operator S_z, indicating the persistent spin texture (PST) (anti-)aligned in the out-of-plane direction.

Next, we will analyze the electronic states over the full Brillouin zone. We note that the unidirectional spin texture of SnTe was previously considered in terms of the effective $\vec{k} \cdot \vec{p}$

models which aptly describe the physics around the valence band maximum (VBM) [114,116]. However, in the presence of doping the regions beyond the VBM may also contribute to the spin transport. In figure 6.1 (c)-(e), we have plotted the topography of the topmost valence band along with the corresponding momentum-resolved map of the energy splitting and the spin texture. The three-dimensional view helps to localize the regions near the Fermi level which could potentially be reached at sufficient level of doping. Indeed, the maxima residing close to the X/Y points are connected by saddle-like regions forming a shape similar to a four-point star. Importantly, from the map shown in figure 6.1(d) we can conclude that the regions of the highest splitting reflect the topography of the band; in the neighborhood of the VBM the values of splitting are as high as $80\ meV$ whereas around the saddle-like regions they achieve $50\ meV$. Such values are certainly sufficient to ensure proper functioning of devices at room temperature. Finally, the spin texture (figure 6.1(e)) reveals only the S_z component is consistent with the prediction based on symmetries (the negligible components S_x/S_y are omitted in the figure). The sign reversals present in (e), e.g. along the Γ—Y line, seem to be associated to the swapping of sub-bands forming the Kramer's pairs.

Finally, we will turn our attention to the spin Hall conductivity reported in figure 6.1(f). Due to 2D nature of the structure (both charge and spin currents have to be in-plane), the only non-vanishing independent components of the SHC tensor are $\sigma_{xy}^{z} = -\sigma_{yx}^{z}$. As one could expect from the huge impact of the SOC on electronic states in the considered energy window (figure 6.1(b)), the estimated magnitudes are relatively large, reaching almost 400 $\frac{\hbar}{e}(\Omega \times cm)^{-1}$ at the first resonance peak. As derived in the previous chapter, the *p*-type doping required to achieve such a value would have to be as large as $10^{14}\ \frac{e}{cm^2}$ which is difficult to

realize. In fact, experimental studies indicated a slight intrinsic *p*-type doping ($10^{11}\frac{e}{cm^2}$) in 2D-SnTe samples [78,79], suggesting that SHC would be rather small. We note, however, that the spin Hall angles can still be sufficiently large because of the low charge conductivity; this may enable an efficient spin injection/detection in such a case. Importantly, we have not found any significant difference in SHC between the polar and centro-symmetric SnTe phases which ensures the realization of SHE even in the presence of doping.

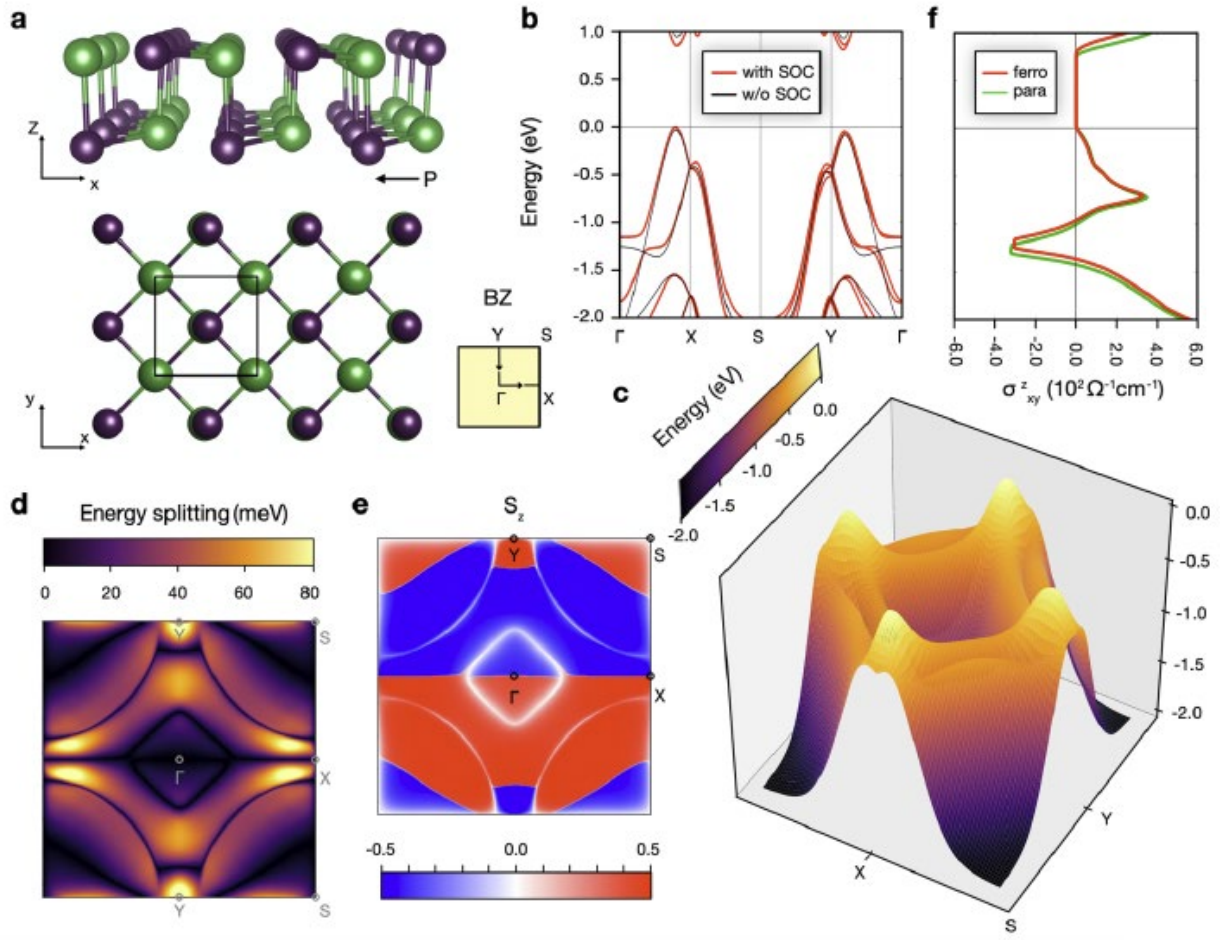


Figure 6.1: Structural and electronic properties of SnTe monolayer. (a) Top and side view of the optical structure. The Sn/Te atoms are displayed as green/purple spheres. Black rectangle denotes the unit cell and the inset shows the Brillouin zone. (b) Electronic structure calculated along high-symmetry lines marked in (a). Red/black lines represent the bands with/without including SOC. (c) Three-dimensional view of the topmost valence band over the entire BZ revealing its quasi-fourfold shape. The VBM is located close to X along Γ-X direction, but similar local maxima are present around Y. The energy splitting and the spin polarization of this band are presented in panels (d) and (e), respectively.

The components S_x and S_y are negligible in this case, thus only S_x component is shown. The labels marked in the maps denote high-symmetry points in the BZ, as defined in the inset in (a). (f) Spin Hall conductivity σ_{xy}^{z} as a function of chemical potential calculated for the ferroelectric structure (red) shown in (a-b) and the corresponding paraelectric phase (green).

6.3 All-in-One Spin Transistor

The all-in-one spin transistor is designed based on the geometry illustrated in figure 6.2(a). The spin injection is realized via the direct spin Hall effect in the left part of the device. The charge current along the y direction is converted into a spin current along x. The spins with out-of-plane polarization are then efficiently injected into a gate-controller region. The logic functionally is achieved by a purely electric manipulation of the anisotropy in spin lifetime determined by the presence or absence of the PSH mode, similar to approaches employing semiconductor quantum wells [106]. In the absence of electric field ($\vec{E} = \vec{0}$) the robust PSH maintains the spin polarization along the $+z$ axis; we note that no spin precession occurs because the PST is always aligned or anti-aligned to the polarization of the spin current, whereby the sign depends on the electron's momentum and/or specific sub-band in a doped system. The spin current is then injected into the right part of the transistor and detected through the inverse spin Hall effect which induces the electric current along the y direction. On the contrary, setting $\vec{E} \neq \vec{0}$ perturbs the PSH mode leading to spin dephasing upon electron scattering. In this case the Hall voltage generated in the detecting region should be significantly reduced.

Let us analyze in more detail the physical mechanisms that cause such a modulation of the PSH. Figure 6.2(b) compares the band structures of SnTe calculated at different values of E. It is clear that the external electric field lifts the valence band degeneracy along the Γ—X—S line and further separates the energy of both pairs of doublets associated with the eigenvalues

of the $\overline{M_z}$ symmetry operator. While the electric potential along z induces spatial anisotropy between the layers, the glide reflections will not be a valid transformation anymore, thus the observed change in the electronic structure along the high-symmetry lines is not surprising. Based on the general considerations performed by Tao and Tsymbal [115], we note that for $\vec{E} = \vec{0}$ the $\overline{M_z}$ in spin space anticommutes with σ_x and σ_y which results in zero expectation values of both in-plane components of spin operator. Since such a condition cannot hold for $\vec{E} \neq \vec{0}$, we expect a severe change in the spin texture. Indeed, our DFT calculations confirmed the emergence of the in-plane spin components over the entire BZ, as shown in figure 6.2(c). Although around the VBM the S_x/S_y are rather small, they can significantly detune the PST provided that a sufficiently strong electric field is applied.

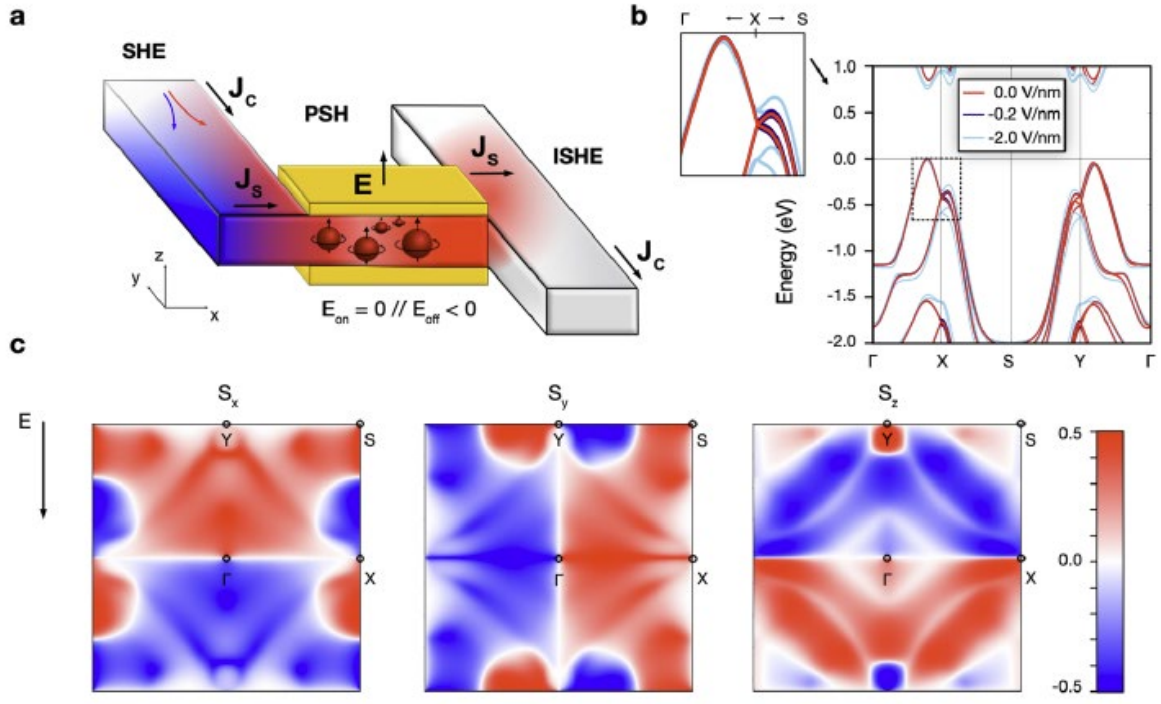


Figure 6.2: Properties and applications of biased 1ML-SnTe. (a) Operation principle of all-in-one spin transistor based on 2D-SnTe. The spin injection is realized via SHE which induces the accumulation of spins polarized along +z. In the on state (E = 0) the direction of the spin-orbit field is parallel to spin orientation, thus no spin procession occurs during the transport along the channel. The spin orientation is further detected based on the ISHE and measured as an induced voltage. In the off state

(E < 0) the PSH state is detuned by the electric field leading to spin decoherence. The ISHE is largely limited and the measured Hall voltage is negligible. The colors in the scheme denote the direction of spins, but do not reflect any numerical values. (b) Band structure of SnTe monolayer calculated for different values of electric field E. The inset shows the details around the VBM. (c) Spin polarization of the topmost valence band at $E = -2.0$ V/nm calculated over the entire BZ. We note that even close to the band maxima the components S_x and S_y are different from zero. The labels in the maps refer to high-symmetry points of the BZ.

We emphasize that the presence of PSH is strictly related to the space group (number 31) of 2D-SnTe. This raises a question about the thickness limit for the proposed spin-transistor or, more specifically, for how many layers the *Pmn2₁* space group remains a valid description of the crystal. From the experimental side, ultrathin films form in a layered structural (γ) phase until around 3ML, while thicker samples may already consist of either mixed α and γ phases at low temperatures, or β and γ phases at room temperature [79]. This indicates an important limit for the proposed all-in-one spin device. In the next paragraphs, we will report a more detailed analysis of 2ML and 3ML structures in different configurations. In particular, we will demonstrate that PSH/SHE combinations can be still realized in such systems, which softens a strict criterion of a truly monolayer material.

Ultrathin SnTe films have been constructed by stacking single layers along vertical direction either in parallel (AA) or anti-parallel (AB) [117], corresponding to polar and antipolar configurations, respectively. We have further optimized the internal coordinates and lattice parameters imposing strict convergence criteria. Our calculations revealed a clear tendency to stabilize in the AA configuration for each considered film thickness (see table 6.1, δE denotes the energy difference between two stacking orders), but we will briefly compare the structural and electronic properties of polar and anti-polar SnTe illustrated in figure 6.3(a)-(c). At first glance, the geometry does not reveal any striking dissimilarity except for the direction of the

polar displacement in the upper layer. However, the electronic structures represented by the orange and dark green lines corresponding to the AA and AB respectively, are dramatically different. While in the former the bands seem to consist of two slightly shifted/modified replicas of the 1ML-SnTe valence states visible close to the Fermi level, in the latter the severe modification of the electronic structure suggests a more fundamental property change.

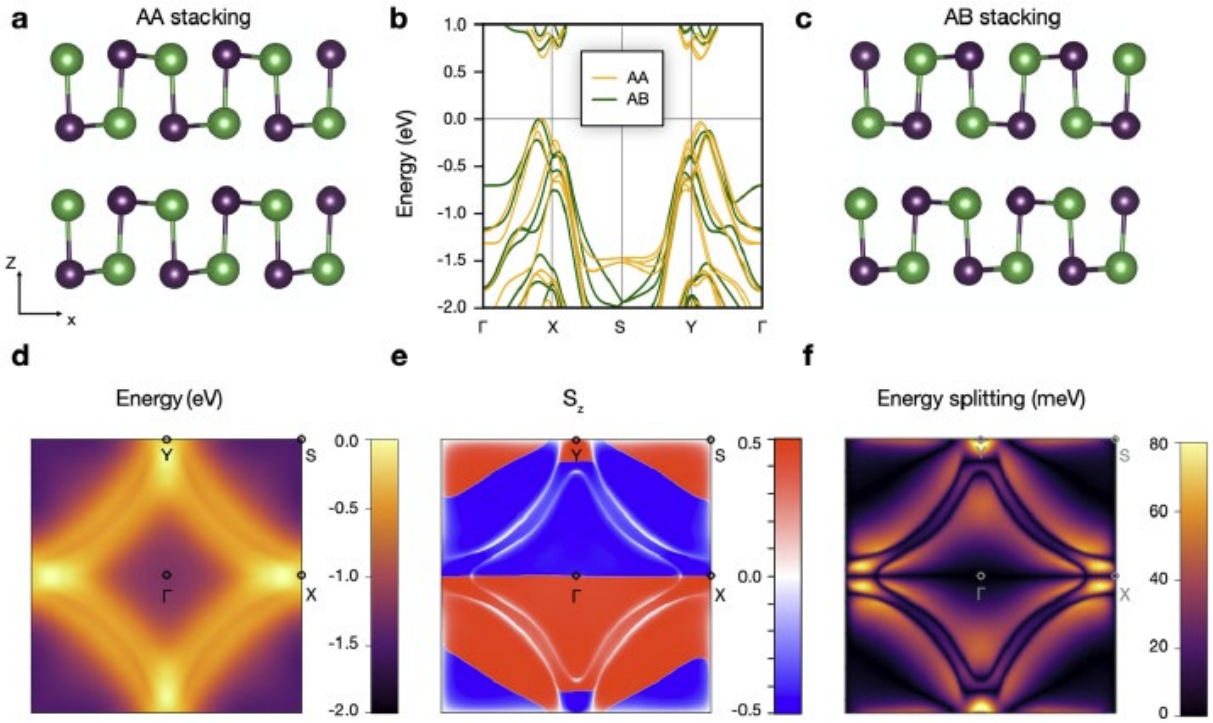


Figure 6.3: Geometry and electronic structure of the 2ML-SnTe. (a) Optimized structure of the AA-stacked bilayer. (b) Band structure calculated for the AA and AB stacking configurations, represented by orange and dark green lines, respectively. (c) Relaxed geometry of the AB-stacked bilayer. (d) Momentum-resolved band topography, (e) the associated spin texture (S_z) and (f) the energy splitting of the topmost valence state calculated over the entire BZ. The negligible components S_x and S_y are omitted in (e).

Indeed, a closer analysis of the crystal lattice reveals that the relaxed AA structure maintains the *Pmn2₁* symmetry, while the AB configuration belongs to space group number 62 (*Pnma*), which does not support the PST [115]. This observation is further confirmed by the complementary plots evaluated over the entire BZ; in the AA stacking, the topography of the

topmost valence band (figure 6.3(d)) is quite similar to its monolayer counterpart from figure 6.1(c), and the spin texture contains only the S_z component which confirms the emergence of the PSH mode. In addition, the momentum resolved map of the energy splitting calculated for this band suggests that, in terms of the spin-dependent electronic properties, the relaxed AA configuration could be equally useful as the 1ML-SnTe. Finally, we note that the AB structure manifests an intricate spin texture (not shown), as expected from the difference in the space group symmetry.

Figure 6.4(a)-(c) reports the properties of 3ML-SnTe. The relaxed geometry reveals the slight differences in ferroelectric displacement between the subsequent atomic layers; they display an oscillating dependence with respect to the layer number, being the largest close to the surface (compare d_{out} and d_{in} parameters in table 6.1). This conclusion is consistent with the previous studies of ferroelectricity in SnTe films [92,118], whereby the distinct surface and bulk like properties are explained in terms of different coordination of the inner and outer ions. Although the relaxation has slightly changed the ionic displacements and lattice constants, it has not altered the symmetry of the cell, which still belongs to the *Pmn2$_1$* space group. Also, the electric polarization (P_{eff}) is rather robust with increasing number of layers (see table 6.1). However, the analysis of the momentum-resolved map of the spin-splitting (figure 6.4(b)) indicates that the PSH may become less stable for larger film thickness, especially at higher temperatures, as the spin-splitting does not exceed $50\ meV$ and rapidly decreases away from the band maxima. We also emphasize a larger number of bands close to the Fermi level which, in principle, could contribute to spin transport even at moderate doping. As evident from figure 6.4(c), the lower bands have a trend to lose the spin texture, which could also be detrimental

for the PSH. Last, we note that the spin Hall conductivities (figure 6.4(d)) hardly change with the number of layers which confirms that the PSH/SHE combination can be realized in SnTe films.

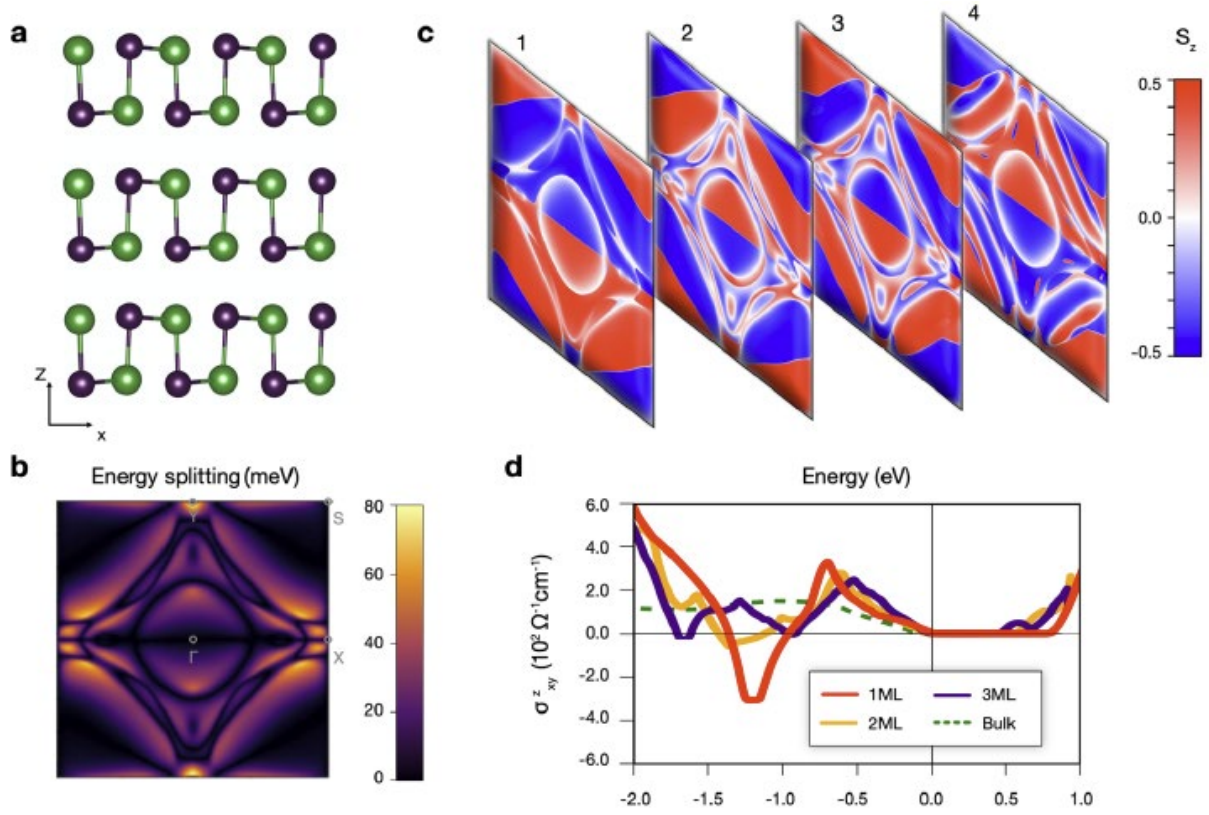


Figure 6.4: Properties of multilayer SnTe. (a) Relaxed geometry of the 3ML-SnTe in the AA stacking configurations. (b) Momentum-resolved map of the spin-splitting in the topmost valence band calculated over the entire BZ for the structure shown in (a). (c) Spin textures of four topmost valence bands enumerated 1-4. The negligible in-plane components are omitted. (d) Spin Hall conductivity σ^z_{xy} as a function of chemical potential calculated for different multilayers and bulk α-SnTe in orthorhombic setting. Spin Hall conductivities of the multilayers are normalized by their effective volumes listed in Table 6.1.

Finally, let us remark on the accuracy of our density functional theory approach. First, SnTe thin films are van der Waals (vdW) lone-pair ferroelectrics, whereby the hybridization interactions compete with the Pauli repulsion and strongly depend on the number of layers [92,118]. Although such systems are challenging to simulate, a reliable estimation of the geometry is a key of ingredient of the analysis, as the emergence of spin texture is a direct consequence of the ferroelectric distortion. Our extensive tests of computational strategies

confirmed the importance of vdW interactions; we have used the semiempirical Tkatchenko-Scheffler approach which ensured a robust distortion as well as a stable convergence of the lattice in the region between the monolayer and the bulk. Notably, a similar method has been successfully used in a recent study of elemental Te, a peculiar vdW ferroelectric with in-plane polarization due to the interlayer interaction between the lone pairs [119]. Second, one has to determine the favorable stacking of the multilayer. We have found the strong preference for polar (AA) structure (see table 6.1), which is in reasonable agreement with the previous theoretical calculations employing an atomic orbital basis [120]. Surprisingly, experimental results from the same study indicate that the anti-polar configurations are more stable which may be assigned to an interplay of several factors, such as details of the growth (bottom-up versus cutting from the bulk), coexistence of different phases (α, β, γ) [79,121], or the influence of the substrate [122]. Last, we emphasize that the calculated intrinsic spin Hall conductivities are extremely sensitive to details of the electronic structure. In order to obtain valid predictions, we have used pseudo-hybrid Hubbard self-consistent approach ACBN0 which was previously demonstrated to provide an excellent accuracy in the case of the bulk phase [113].

6.4 Conclusions

In summary, we have demonstrated that the all-in-one spin transistors employing the combined effect of PSH and SHE can be constructed based on the SnTe multilayers. Such devices could benefit from the long spin lifetime ensured by the PSH as well as the efficient spin/charge interconversion without ferromagnetic electrodes. However, one needs to have in mind several important conditions that have to be satisfied. First, a natural thickness limit for such a device would be of 3ML-SnTe (~ 20 Å), as concluded from our first-principles calculations

and earlier experimental results. Second, the emergence of PSH is strictly determined by the crystal space group; it is possible only for multilayers with a polar stacking order, which preserve $Pmn2_1$ symmetry. Third, the realistic predictions of spin transistor performance should take into account the role of the substrate. In particular, the interface could change the favorable stacking of vdW layers yielding an anti-polar instead of a polar order. Moreover, the presence of the substrate may itself perturb the spin wave mode. In such a case, the spins would not be fully protected from relaxation and the disorder could cause a decrease in spin lifetime, as observed in similar two-dimensional devices [123,124]. In order to maximally preserve the spatial symmetry, the system should preferably have a sandwich structure.

We believe that the presented results will stimulate a further search of low-dimensional structures with similar properties. We also note that, in general, the simultaneous use of PSH and SHE does not need to limited to 2D ferroelectrics. There are several bulk materials with persistent spin texture (e.g. BiInO3, LiTeO3, $CsBiNb_2O_7$, or Bi_2WO_6) [113,123,124], they may also reveal a strong spin Hall effect and serve in all-in-one spin devices with a different operation principle.

6.5 Methods

Our calculations based on density functional theory (DFT) were performed using the Quantum ESPRESSO package [8,9]. We treated the ion-electron interaction with norm-conserving pseudopotentials from the pslibrary database [89] and expanded the electron wave functions in a plane wave basis set with a cutoff of 150 Ry. The exchange and correlation interaction was taken into account within the generalized gradient approximation (GGA) parameterized by the Perdew, Burke, and Ernzerhof (PBE) functional [87]. We modeled the

SnTe multilayers within the slab approach minimizing the errors induced by the periodic boundary conditions with a large vacuum region of at least 20 Å and with dipole corrections added to the local potential. We fully relaxed the structures setting the convergence criteria for energy and forces to 10^{-7} Ry and $10^{-4}\frac{Ry}{bohr}$, respectively. The Tkatchenko-Sheffler van der Waals corrections were included in order to ensure the stability and reliable lattice parameters of weakly interacting layers [30]. The electronic structures were further corrected by using a novel pseudo-hybrid Hubbard self-consistent approach ACBN0, with the calculated U parameters equal to 0.17 and $2.15\ eV$ for Sn and Te, respectively. Although the SOC was included self-consistently in DFT calculations, the spin-orbit related quantities were evaluated as a post-processing step employing the tight-binding Hamiltonians; the latter were constructed from the projections of eigenfunctions on pseudoatomic orbitals following implementation in the PAOFLOW code. After interpolating the Hamiltonians to an ultra-dense k-points mesh of $140 \times 140 \times 1$ we calculated the spin polarization of each eigenstate $\psi(\vec{k})$ represented as $S(\vec{k}) = [S_x(\vec{k}), S_y(\vec{k}), S_z(\vec{k})]$, where $S_n(\vec{k}) = \langle\psi(\vec{k})|\sigma_n|\psi(\vec{k})\rangle$ and σ_n denote the pauli matrices. Spin Hall conductivities were computed from the Kubo's formula following the details given in chapter 5. Finally, we modeled the influence of electric fields perpendicular to the layers by modifying the tight-binding Hamiltonians.

6.6 Acknowledgments

The members of the AFLOW Consortium (www.aflow.org) acknowledge the grant ONR-MURI N000141310635. The authors also acknowledge Duke University—Center for Materials Genomics—and the CRAY corporation for computational support. Finally, we are grateful to the

High Performance Computing Center at the University of North Texas and the Texas Advanced Computing Center at the University of Texas, Austin.

CHAPTER 7

QUANTUM INFORMATION IN MATERIALS SCIENCE

Using quantum mechanical phenomena as a computational tool was first proposed in a seminal paper by Richard Feynman, in which he concluded that the laws of physics do not prevent effective computation at quantum mechanical scales [127]. Since then, it has been shown that the quantum mechanical principle of superposition can be leveraged to gain computational advantage over classical computers, in algorithms like Peter Shor's quantum procedure for factoring numbers [128]. That is, certain problems which take exponential time to solve on classical computers, with respect to the input length, can be solved in polynomial time with quantum resources. This section outlines the most basic interpretation of quantum computers and introduces the early applications of quantum algorithms to materials science and quantum chemistry.

7.1 Quantum Computing

Quantum computers operate by manipulating qubits, which are the fundamental objects for storing information on quantum hardware, analogous to bits in classical computers. While a bit on classical computers stores a binary value (generally referred to as 0 and 1), qubits take on superpositions of these two states. A system with N two state components can be completely described with N bits classically, while a quantum description of the same system requires $2^N - 1$ complex numbers. To introduce the basic principle consider an object that, when measured, is found in either state $|0\rangle$ or $|1\rangle$. An arbitrary state $|\psi\rangle$ for such a (single) qubit is written as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \tag{7.1}$$

where $\alpha, \beta \in \mathbb{C}$. The state is normalized such that $\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1$, and the probability of measuring $|\psi\rangle$ in state $|0\rangle$ or $|1\rangle$ is given by $|\alpha|^2$ or $|\beta|^2$ respectively.

Quantum computers can, of course, perform any operation computable on a classical computer. A simple demonstration of this principle, and an introduction to operators acting on quantum states, is given by the NOT gate in a classical computer. The NOT gate takes one input and outputs the opposite binary value of the input. E.g. A gate where an input of 0 yields an output of 1, and an input of 1 yields an output of 0. To see how the classical operator could function in a quantum computer, take the σ_x Pauli matrix in natural units, $\hat{X} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$. Now, apply the $\hat{X}$ gate to a pure $|0\rangle$ state (equation 7.1 with $\alpha = 1$ and $\beta = 0$).

$$\hat{X}|0\rangle = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} = \begin{pmatrix}0\\ 1\end{pmatrix} = |1\rangle \tag{7.2}$$

Applying the σ_x gate to a qubit to a pure $|1\rangle$ state ($\alpha = 0$ and $\beta = 1$) is similar.

$$\hat{X}|1\rangle = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix} = \begin{pmatrix}1\\ 0\end{pmatrix} = |0\rangle \tag{7.3}$$

From equations 7.2 and 7.3, it is clear that a qubit is capable of functioning the same way as a classical bit under the NOT operation, and it can be shown that this principle holds for other types of classical gates. What quantum computers do that a classical computer cannot is to represent mixed states, where $\alpha \neq 0$ and $\beta \neq 0$ at the same time.

Unitary operators have their complex conjugate transpose equivalent to their inverse. i.e. an operator $\hat{U}$ is unitary if and only if $\hat{U}^\dagger = \hat{U}^{-1}$. Quantum operators are necessarily unitary. Summing the probabilities of obtaining each state must yield 1 both before and after an operator is applied. The evolution of a quantum state via unitary operation is governed

appropriately by the laws of quantum mechanics; the resulting state after applying a unitary operation implies the input state, motivating the principle of reversible computing. While classical computers can escape reversibility by incurring heat or other forms of entropy, quantum computers must maintain fidelity of their quantum states throughout the lifetime of their computation. Given no noise or environmental turbulence, quantum computations are inherently reversible processes. For further discussion of gate model quantum computing and reversibility in quantum computers see reference [128].

To understand how multiple qubits can interact, consider the product state $|\Psi\rangle$ of just two qubits in states $|\psi_1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$ and $|\psi_2\rangle = \begin{pmatrix}\delta\\ \gamma\end{pmatrix}$, given by:

$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \otimes \begin{pmatrix}\delta\\ \gamma\end{pmatrix} = \begin{pmatrix}\alpha\delta\\ \alpha\gamma\\ \beta\delta\\ \beta\gamma\end{pmatrix} \tag{7.4}$$

The state $|\Psi\rangle$ is said to be factorizable, because the complete state can be written as a product of two composite states ($|\psi_1\rangle$ and $|\psi_2\rangle$ in this case). Certain states, such as Bell states, can be achieved such that no factorization exists [129]. That is, for some state

$$|\Psi\rangle = \begin{pmatrix}A\\ B\\ C\\ D\end{pmatrix} \tag{7.5}$$

$$|\Psi_{\text{Bell}}\rangle = \begin{pmatrix}A\\ B\\ C\\ D\end{pmatrix}$$

there is no such $|\psi_1\rangle$ and $|\psi_2\rangle$ which could satisfy $|\psi_1\rangle \otimes |\psi_2\rangle = |\Psi\rangle$. A system of two or more qubits in a configuration which cannot be represented as a product state are said to be

entangled. If the entangled state $|\Psi\rangle$ is known *a priori*, measurement of $|\psi_1\rangle$ implies the value of $|\psi_2\rangle$ and vice versa.

Quantum computers possess other unintuitive qualities which are not discussed here, such as teleportation, and the inability to clone states. A discussion about quantum teleportation is available in reference [130], and the original proposal of the *no cloning theorem* can be found in reference [131]. The next section discusses the modern state of quantum computing in materials science and introduces some common algorithms that prove useful for simulation molecules and solids.

7.2 Quantum Computing in Materials Science

Quantum supremacy, a term used quite loosely in modern reports about the current state of quantum computing, is achieved by using a quantum processing unit (QPU) to solve a problem in less time than a classical CPU. A recent publication from Google engineers claimed quantum supremacy by calculating with quantum hardware the diffraction pattern of interacting laser pulses, a process that becomes exponentially harder to simulate on a classical computer as the physical size of the sample space grows [132]. Another study, by researchers at IBM, computes the bond lengths of H_2, LiH, and BeH_2 molecules by simulating them on a quantum processor [133]. Though you cannot buy a quantum computer in your local electronics store, they are quickly becoming a more ubiquitous technology, and some are even available for public use through the IBMQ web interface*.

Quantum computers promise the ability to simulate large correlated systems of

* https://www.research.ibm.com/quantum-computing/

fermions, enabling emulation of large-scale molecules and materials that are intractable on classical computers. Because noisy intermediate-scale quantum (NISQ) processors are limited to a few qubits with limited connectivity and low fidelity, researchers are developing hybrid algorithms, which perform some computation on classical computers and utilize quantum computers only when a QPU would achieve exponential advantage over a CPU. The most popular hybrid classical algorithm is the variational quantum eigensolver (VQE), which aims to find minimal or maximal eigenvalues for a Hamiltonian by defining it in terms of variational parameters [134]. The Hamiltonian is simulated and measured on a QPU, and the parameters are updated with a classical optimization algorithm to achieve a global minimum or maximum. Explicit details of the VQE algorithm and an implementation are described in the following chapter. Several others noteworthy strategies for simulating atomic systems have become popular, such as unitary coupled cluster (UCC), quantum phase estimation (QPE), trotterization, and adiabatic state preparation, and are briefly described in chapters 8 and 9. See reference [135] for a more rigorous account of these procedures. The following chapters discuss the simulation of periodic crystalline silicon on quantum computers and propose various techniques for obtaining electronic structure for materials using a QPU.

CHAPTER 8

QUANTUM COMPUTATION OF SILICON ELECTRONIC BAND STRUCTURE*

Development of quantum architectures during the last decade has inspired hybrid classical-quantum algorithms in physics and quantum chemistry that promise simulations of fermionic systems beyond the capability of modern classical computer, even before the era of quantum computing fully arrives. Strong research efforts have been recently made to obtain minimal depth quantum circuits which could accurately represent chemical systems. Here, we show that unprecedented methods used in quantum chemistry, designed to simulate molecules on quantum processors, can be extended to calculate properties of periodic solids. In particular, we present minimal depth circuits implementing the variational quantum eigensolver algorithm and successfully use it to compute the band structure of silicon on a quantum machine for the first time. We are convinced that the presented quantum experiments performed on cloud-based platforms will stimulate more intense studies towards scalable electronic structure computation of advanced quantum materials.

8.1 Introduction

Quantum computing aims to leverage superposition, entanglement, and interference of quantum bits in order to tackle computational tasks that scale exponentially on classical computers [136,137]. While renowned quantum algorithms, such as unsorted database search or integer factorization require resources that remain out of reach [128,138], quantum

* This chapter is reproduced, with permission from The Royal Society of Chemistry, from the paper: *Quantum computation of silicon electronic band structure* by Frank T. Cerasoli, Kyle Sherbert, Jagoda Sławińska, and Marco Buongiorno Nardelli, published in Physical Chemistry Chemical Physics 22, 21816-21822 (2020).

chemistry calculations are gaining steam as a key application performed on available quantum architectures [139,140]. The idea of so-called quantum simulations, originally proposed by Feynman [141-143], relies on a mapping between the fermionic system and the set of qubits, so that the dynamics of the former is directly followed by the latter. Therefore, wave functions of complex many-body systems could be effectively reproduced in quantum measurements performed on qubits, providing a tool to compute desired quantities with an unprecedented accuracy. Even though available quantum computers contain merely few tens of qubits [132], they have been employed to solve quantum chemistry problems, such as the estimation of nuclear binding energies or molecular ground states [133,144-146]. Remarkably, these successful quantum experiments relied on variational approaches that greatly reduced the required hardware resources, inspiring more active research in order to solve elusive condensed matter systems beyond quantum chemistry [147-151].

Here, we put forward an approach to calculate the electronic structure of the periodic crystal on a quantum computer. While developments of quantum computation for molecules were primarily focused on the ground state energies, to evaluate a band structure one needs to determine the excited states. We have shown that a standard hybrid quantum/classical algorithm, variational quantum eigensolver (VQE) can easily be adapted to provide an accurate estimation of the electronic bands in solids. In particular, by casting a Si tight-binding (TB) Hamiltonian in terms of fermionic operators, we have designed a low-depth quantum circuit, robust enough to capture the electronic properties of a crystal in the reciprocal space. The quantum measurements have been performed on sets of qubits available remotely via cloud-based platforms provided by IBM and Rigetti Computing. Importantly, we have tested different

classical optimization routines that minimize expectation values, corrected beforehand against the readout errors. Comparison between bands computed on the quantum processors, the quantum virtual machine, and by classical diagonalization revealed a satisfactory agreement, confirming the validity of the algorithm which could be generalized to explore materials more complex than crystalline silicon.

8.2 Hamiltonian Representation

Let us consider a silicon lattice in the diamond cubic structure. The Hamiltonian describing the electronic system can be approximated, in atomic units, as

$$\widehat{\mathrm{H}}_{\mathrm{el}} = -\sum_{\mathrm{i}} \frac{\nabla_{\mathrm{i}}^2}{2} - \sum_{\mathrm{i,j}} \frac{\mathrm{Z_j}}{|\mathrm{R_j} - \mathrm{r_i}|} + \sum_{\mathrm{i<j}} \frac{1}{|\mathrm{r_i} - \mathrm{r_j}|} \tag{8.1}$$

where r_i (R_i) are the positions of electrons (nuclei) and Z_j denotes the nuclear charge, respectively. We have assumed the Born-Oppenheimer approximation and considered the nuclei as stationary charges, thus neglecting their kinetic energy and treating the ion repulsion as a constant. The last term of equation 8.1 represents the electron-electron interaction, whose correct simulation is one of the long-term goals of quantum simulation. However, we are now primarily focused on the proof-of-principle band structure calculations and have disregarded the electronic correlations for the purpose of the present study.

In order to convert the Hamiltonian into a computational problem, a suitable basis set needs to be selected. While different representations were proposed for quantum computation [147], we introduce here a simple basis of atomic orbitals at each lattice site arising from the tight-binding (TB) approximation. The unit cell of silicon contains two tetrahedrally coordinated ions and is well described in terms of s, p_x, p_y, and p_z orbitals centered at each atom. Because

magnetic order is absent, the spin degrees of freedom can be omitted in the analysis. Using the second quantization formalism, we can express the TB Hamiltonian via creation and annihilation operators ($a_{in}^{\dagger}$ and $a_{in}^{\dagger}$) acting at the orbital n and the site R_i:

(8.2)

$$\hat{\mathrm{H}} = \sum_{\mathrm{i,n}} \mathrm{E_n a_{in}^{\dagger} a_{in}} - \sum_{\langle \mathrm{i,j} \rangle, \mathrm{n,m}} \mathrm{t_{in,jm} a_{in}^{\dagger} a_{jm}}$$

In this expression, E_n correspond to the atomic energies and $t_{in,jm}$ denote the hopping integrals whose numerical values have been reported elsewhere [152]. Only the tunneling between pairs of nearest neighbors, denoted by the $\langle i, j \rangle$ summation, have been considered. The Hamiltonian can be then easily converted to the momentum space via standard Fourier transform applied to the raising and lowering operators. Last, such a representation ($\widehat{H}_k$) needs to be mapped onto the system of qubits.

In practice, qubits are manipulated on a quantum processor by operating on a set of Pauli matrices $\hat{X}$, $\hat{Y}$, $\hat{Z}$, and $\hat{I}$, the latter denoting 2×2 identity matrix. Any Hermitian matrix can be decomposed using a complete Pauli basis for matrices of dimension $N = 2^n$ with $n = log_2 N$ terms, that can be generated by taking a tensor product:

$$\{\widehat{\sigma}\}_n = \{\hat{I}, \hat{X}, \hat{Y}, \hat{Z}\}^{\otimes n} \tag{8.3}$$

Thus, the TB Hamiltonian can be decomposed as follows:

$$\widehat{H}_k = \sum_{i=1}^{4^n} c_{ik} \widehat{\sigma}_i \tag{8.4}$$

where the set $\{\widehat{\sigma}\}_n$ is the set of 4^n possible basis matrices, and $\{c_k\}_n$ is a set of complex coefficients. $\{c_k\}_n$ is known as the spectral decomposition and can be determined easily. In

particular, we can exploit the orthogonality of Pauli matrices and the trace inner product between two of them:

$$Tr\left(\hat{\sigma}_j^{\dagger}\hat{\sigma}_j\right) = 2^n\delta_{ij} \tag{8.5}$$

By taking the inner product $\widehat{H}_k^{\dagger}\widehat{\sigma}_i$, we can eliminate all terms but one from the sum, yielding:

$$c_i = \frac{Tr\left(\widehat{H}_k^{\dagger}\widehat{\sigma}_i\right)}{2^n} \tag{8.6}$$

Therefore, the Hamiltonian is represented by a list of coefficients corresponding to each of the 4^n Pauli basis matrices suitable for simulation on a QPU [153].

8.3 Variational Quantum Eigensolver

We have computed the energy spectrum using the variational quantum eigensolver in conjunction with overlap-based techniques. VQE is a standard hybrid quantum-classical algorithm capable of determining the lowest or highest eigenvalue of an operator using minimal quantum resources, implemented by combining measurements on a quantum computer with classical optimization routines [154-156]. The ground state wave function and energy can be found based on Rayleigh-Ritz variational principle, whereby the energy expectation value can be minimized by a specific set of parameters. In practice, the state preparation and the expectation value measurements are implemented on a quantum machine, while the optimization of the parameters is performed classically. The whole algorithm used for the ground state calculation can be summarized in three following steps:

(1) We create a quantum circuit $\hat{V}\left(\vec{\theta}\right)$ depending on a set of parameters $\vec{\theta}$, known as the variational form. Then, we prepare a trial wave function (or ansatz) $\left|\psi\left(\vec{\theta}\right)\right\rangle = \hat{V}\left(\vec{\theta}\right)$, where $|0\rangle$ denotes an initial state ensuring the measurement of each qubit.

(2) We measure the expectation value of $\widehat{H}_k$, which depends on the parameters $\vec{\theta}$, $E(\vec{\theta}) = \langle\psi(\vec{\theta})|\widehat{H}_k|\psi(\vec{\theta})\rangle$. The Hamiltonian is represented by a series of operators. The wave function $|\psi\rangle$ is measured in the Pauli basis, yielding each $\langle\hat{\sigma}_i\rangle$. We can then reconstruct $\langle\widehat{H}_k\rangle$ with the spectrum c_k:

$$\langle\widehat{H}_k\rangle = \sum_{i=1}^{4^n} c_{ik}\langle\hat{\sigma}_i\rangle \tag{8.7}$$

The measurement should be treated as a probabilistic element of the algorithm and needs to be performed several times. An arbitrary precision can be achieved with a sufficient number of repetitions.

(3) We apply a classical optimization routine to explore the parameter space and minimize $E(\vec{\theta})$. We define $\epsilon_0 = \langle\psi(\vec{\theta}_{min})|\widehat{H}_k|\psi(\vec{\theta}_{\min})\rangle$ as the ground state energy, where $\vec{\theta}_{min}$ denotes the set of parameters minimizing the expectation value of $\widehat{H}_k$.

8.4 Energies Beyond the Ground State

After having determined the ground state, we can calculate excited states using a procedure similar to the quantum deflation algorithm that exploits orthogonality of the Hamiltonian eigenvectors [157-159]. In particular, we define an effective Hamiltonian ($\widehat{H}'_k$) whose lowest eigenstate is the excited state of the original one ($\widehat{H}_k$). By subtracting from the latter a corresponding ground state projector weighted by the ground state energy, we obtain:

$$\widehat{H}'_k = \widehat{H}_k - \epsilon_0|\psi_0\rangle\langle\psi_0| = \sum_{i=1}^{4^n}\left(c_i - \epsilon_0\frac{\langle\hat{\sigma}_i\rangle}{2^n}\right)\hat{\sigma}_i \tag{8.8}$$

We observe that the last equality provides the following spectral decomposition of the excited Hamiltonian:

$$c'_i = c_i - \epsilon_0\frac{\langle\hat{\sigma}_i\rangle}{2^n} \tag{8.9}$$

The procedure is used iteratively to determine as many eigenvalues as desired. Updating the spectral decomposition $c_i \to c_i - \epsilon_0\frac{\langle\hat{\sigma}_i\rangle}{2^n}$ effectively removes all ground state contributions from

the Hamiltonian.

We note that the effect of subtracting the ground state density matrix weighted by its corresponding eigenvalue is to project that eigenstate onto the zero value. Because an arbitrary Hermitian matrix can have both positive and negative eigenvalues, special care must be taken to ensure that zero is not erroneously computed as a ground state after all negative eigenvalues are determined. One reconciliation is to subtract a value greater than the maximum eigenvalue from the diagonal elements of the Hamiltonian, ensure that all eigenvalues are lower than zero. Therefore, projecting an eigenstate to zero would not affect the remaining eigenvalues that need to be determined. Such a shift requires the modification of only one coefficient of the spectral decomposition, which stands before the identity matrix.

8.5 Data Acquisition

Before discussing the results of quantum experiments, let us remark on the various techniques that we have employed to compute the band structure of silicon. A careful distinction must be made between the use of quantum processor, quantum virtual machine, and quantum state simulation. In particular, simulated qubits helped us analyze the performance of variational forms and the effect of measurement uncertainty on a noiseless machine. Three independent techniques will be further referenced:

(1) Quantum processor unit (QPU) is prepared for measurements under subsequent set of parameters. The measurements are performed in real time. The available APIs compile quantum programs and directly manipulate qubits, providing measured expectation values in the form of bitstrings.

(2) Quantum virtual machine (QVM) chooses one of the possible outcomes to be "measured", weighted by its respective probability computed with the quantum state simulator (see below). The quantum processor is mimicked, providing a noiseless (unless noise is simulated) simulation of the measurement process. This

method helps to analyze the effects in the band structure determined by discrete measurements of the energy expectation values.

(3) Quantum state simulator (QSS) carries out linear algebra to obtain an exact wave function which would represent the simulated state of a qubit on a quantum processor after the application of specified gates. It can serve as an analytical guideline for quantum measurements. Optimization can be easily performed with the quantum state simulator, providing a convenient framework to test the performance of variational forms.

8.6 Quantum Experiments

Quantum computations of the band structure have been performed following two different techniques, both yielding a correct spectrum while compared with the classical diagonalization of the TB Hamiltonian. The first approach relies on a true quantum measurement, employing one qubit that we access on remote quantum machines Rigetti Aspen and IMBQ Armonk. Although these cloud platforms permit the use of larger resources, the practical realization of the VQE algorithm for diagonalization of the 8×8 Hamiltonian of Si required a substantial amount of time. Therefore, we have started with a reduced Hamiltonian, considering only the interactions between s-states which give rise to the lowest bands of silicon. After neglecting s-p hopping parameters in the original $\widehat{H}_k$, a smaller 2×2 matrix block can be decoupled and diagonalized using VQE on the QPU. Figure 8.1 shows the two-gate circuit acting on a single qubit, often referred to as the mean field ansatz [160], which has been used in the experiment. In principle, to ensure that finding the true minimum is possible, circuits must be designed to span every state allowed by the operating qubits, unless the space is restricted by physical arguments, such as fermionic commutation relations in the UCC strategies [161]. The ansatz below takes a pure state $|0\rangle$ and applies two rotations described by the angles $\vec{\theta} = (\theta, \phi)$. A polar rotation brings the qubit into a superposition of $|0\rangle$ and $|1\rangle$

states, while an azimuthal rotation scans the sphere's latitude. The two rotations produce a state represented by the following wave function:

$$|\psi(\theta,\phi)\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}sin\left(\frac{\theta}{2}\right)|1\rangle \tag{8.10}$$

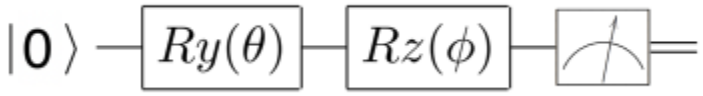


Figure 8.1: Mean-field circuit acting on a single qubit has been employed to determine the lowest bands of silicon. It consists of a polar rotation (R_y) followed by an azimuthal rotation (R_z). In last step, the expectation value of $\widehat{H}_k$ is measured.

The band structure has been computed along a high symmetry line X—Γ—L by repeating the whole algorithm for each of the k-points. Figure 8.2(a) and (b) report the two-band electronic structure evaluated on the quantum machines of IBM (red squares) and Rigetti (green circles), complemented by data from the classical diagonalization (black solid line). In addition, we present the results obtained via quantum-classical algorithm performed on QSS (blue squares) and QVM (yellow circles). While the latter directly follow the bands calculated classically, the quantum data reveal tiny deviations that can be noticed around the high-symmetry points Γ and L for Rigetti and IBM, respectively. The sources of errors in the experiment can be manyfold. The probabilistic aspect can obviously play a role, despite a large number of measurements (8192) taken for each parameterization. Importantly, simulations of noise on QVM have revealed that any gate noise or readout error tends to increase the measured energy, shifting the expectation value toward different eigenstates. As described in the next sections, we have attempted to characterize and reduce the effects of errors arising from the qubit manipulation.

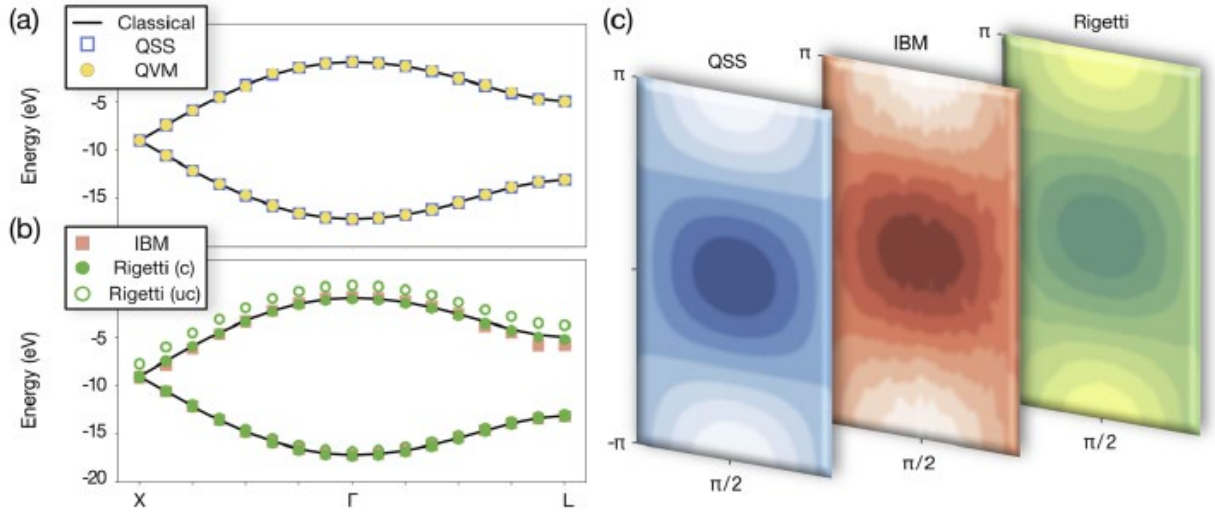


Figure 8.2: (a) The two-band electronic structure of silicon computed along X-Γ-L line using classical diagonalization (black solid line) and hybrid quantum-classical algorithm on quantum state simulator (blue squares) and quantum virtual machine (yellow circles). (b) Same as (a) realized on the QPUs of IBM (red squares) and Rigetti (green circles). We report the data from Rigetti before and after correcting for the readout errors, marked as open and closed circles, respectively. (c) Energy expectation value sampled over the entire parameter space $[-\pi,\pi]$ in the azimuthal angle and $[0,\pi]$ in the polar angle on QSS (blue), IBM (red), and Rigetti (red). Darker (brighter) colors denote lower (higher) values of the energy expectation value.

We note that the standard optimization routines have not been here applied. Instead, we have used the mean-field circuit to measure a dense grid of parameter angles in order to find the minimum expectation value. Sampling the entire parameter space provides a visual tool for analyzing the structure of parameter space. Figure 8.2(c) shows examples of the expectation value surface computed for one selected point $k = \frac{\pi}{4a}\langle 1,1,1\rangle$. Three subsequent panels report the surfaces obtained analytically on QSS (blue), and experimentally on IBM (red) and Rigetti (green). The two latter have been smoothed by minimizing the root-mean-square error across all data points. Again, the data collected on IBM reveals largest irregularities in the energy contour lines, especially compared with the analytical surface evaluated on QSS.

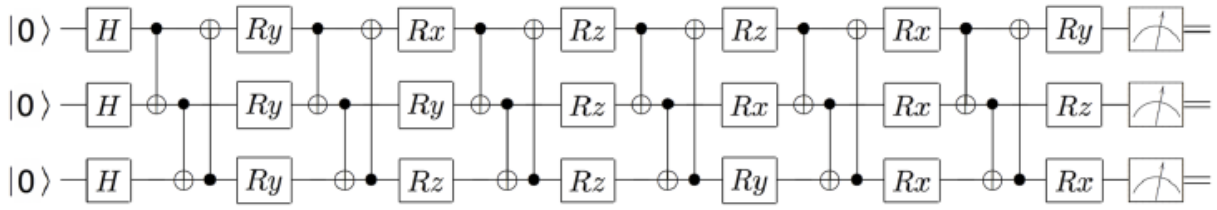


Figure 8.3: The circuit used to diagonalize the 8x8 Hamiltonian. Each qubit is initialized as a pure zero state.

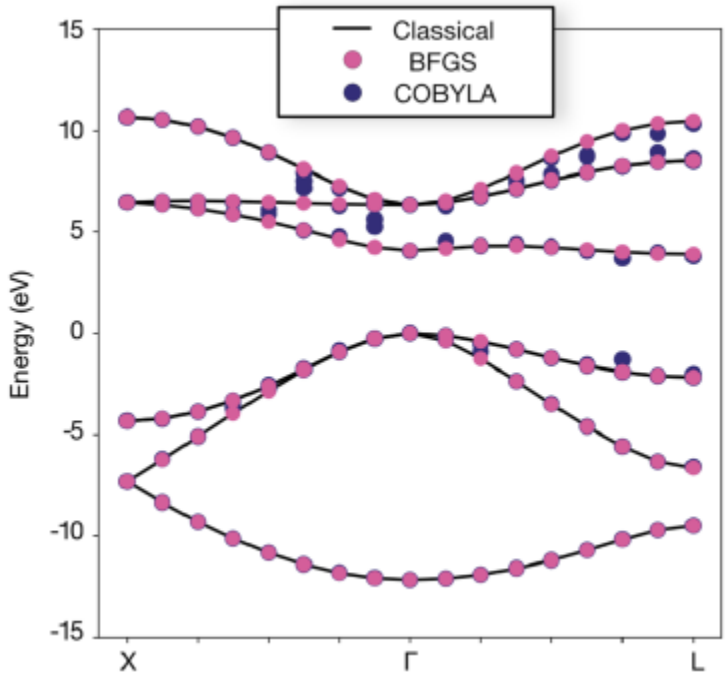


Figure 8.4: Electronic structure of silicon computed via hybrid classical/quantum algorithm on QSS. Different optimization routines BFGS and COBYLA are compared on analytic surface. Black solid lines denote the bands calculated classically.

The second approach, employed to diagonalize full 8×8 Hamiltonian, relies on QSS. Figure 8.3 presents a robust three-qubit circuit that we have designed to variationally minimize the expectation value of $\widehat{H}_k$ at any k-point and each level of excitation. The set of twelve parameters $\vec{\theta} = (\theta_1, \theta_2, \ldots, \theta_{12})$ in this ansatz, measured in the Pauli word basis from the Hamiltonian decomposition defined in equation 8.4, are varied to minimize the energy expectation values. Figure 8.4 displays the electronic structure computed using this circuit,

demonstrating that it is indeed capable of representing the silicon Hamiltonian anywhere along the k-line. Although small discrepancies are again visible, the overall agreement with the bands calculated classically seems to be sufficient. We note that now the results do not depend on external factors that can perturb the behavior of qubits. The deviations are related to the optimization procedures whose proper choice is essential to correctly determine the energy spectrum.

Several classical optimization routines have been tested in conjunction with the three-qubit circuit used for the evaluation of full electronic structure. Minimizing a function in parameter space of twelve dimensions is rather challenging and requires a compromise between the number of measurements and the smoothness of the space being optimized. We have found that the Broyden—Fletcher—Goldfarb—Shanno (BFGS) and Constrained Optimization BY Linear Approximation (COBYLA) routines yielded the most accurate results. The former requires fewer function evaluations to reach a minimum, but it suffers from instability due to the rough surface in parameter space. The latter, being a direct search method, entirely omits the idea of gradient decent which makes it more robust against becoming trapped in a local minimum. Even though it may provide more reliable global minima [162], it occasionally fails to settle on the correct set of parameters. Figure 8.4 clearly shows that especially the excited energy levels are sensitive to fluctuations in the determined parameters. The comparison of both routines, BFGS and COBYLA, eventually indicates the superior performance of the former, at least in the present case.

8.7 Additional Remarks on Measuring Expectation Values

While the previous section was entirely focused on the realization and results of

quantum experiments, the measurements of expectation values need a more detailed discussion. The quantities we have measured on the quantum computer are the expectation values $\langle\hat{\sigma}_k\rangle$, where the operator $\hat{\sigma}_k$ is an n-length Pauli word consisting of an $\hat{I}$, $\hat{X}$, $\hat{Y}$, or $\hat{Z}$ for each qubit. They depend on the state $|\psi\rangle$ of the qubits and could be written as the integral $\langle\psi|\hat{\sigma}_k|\psi\rangle$. Because we do not know $|\psi\rangle$, we must measure the state of each qubit in the computational basis, resulting in a single bitstring (e.g. $|00101\rangle$). Repeating the measurement a large number of times M, we construct the expectation value $\langle\hat{\sigma}_k\rangle$ from the ensemble of bitstrings. In the following paragraphs, we will first consider the single-qubit case $\hat{\sigma}_k = \hat{Z}$, then the multi-qubit case where $\hat{\sigma}_k$ consists only of $\hat{I}$ and $\hat{Z}$ operators and last, the general case including $\hat{X}$ and $\hat{Y}$ operators.

The Pauli operator $\hat{Z}$ can be written in a matrix form:

$$\hat{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

It is a diagonal matrix with eigenvalues $+1$ corresponding to the state $|0\rangle$, and -1 corresponding to the state $|1\rangle$. The expectation value $\langle\hat{Z}\rangle$ is the average of these two eigenvalues, weighted by the number of measurements in each state. If p is the probability that we measure $|0\rangle$ rather than $|1\rangle$, the expectation value $\langle\hat{Z}\rangle$ is given by:

$$\langle Z\rangle = (+1)p + (-1)(1-p) = 2p - 1$$

Now, consider an operator $\hat{A}$ defined as a Kronecker product of $\hat{I}$ and $\hat{Z}$ operators, each acting on their own qubit. It is a degenerate operator with half the eigenvalues $+1$ and half -1. Because its matrix form is diagonal, each bitstring we measure corresponds exactly to an eigenstate. The parity (± 1) of a given bitstring z is precisely the parity of the substring z' which omits any index corresponding to an $\hat{I}$ operator in $\hat{A}$. For example, if $\hat{A} = \hat{I}_5\hat{Z}_4\hat{Z}_3\hat{I}_2\hat{Z}_1$ and $z =$

$|00101\rangle$, the substring z' leaves off the seconds and fifth indices: $z' = |011\rangle$. This string has a weight of two, which is an even parity and therefore corresponds to the eigenvalue $+1$. The expectation values $\langle A \rangle$ is once again an average of $+1$ and -1, weighted by the frequency of the bitstrings corresponding to each of the two states.

Last, let us consider a general Pauli word$\hat{\sigma}_k$. Half of its eigenvalues are again $+1$ and half -1, but bitstrings in the computational basis do not correspond exactly to the eigenstates. We therefore need to diagonalize $\hat{\sigma}_k$. Let $\hat{A}_k$ be the Pauli word which replaces all $\hat{X}$ and $\hat{Y}$ in $\hat{\sigma}_k$ by $\hat{Z}$, and the operator $\hat{U}_k$ changes the basis so that $\hat{\sigma}_k = \hat{U}_k^\dagger \hat{A}_k \hat{U}_k$. Then, for each expectation value we have $\langle \hat{\sigma}_k \rangle = \langle \hat{U}_k^\dagger \hat{A}_k \hat{U}_k \rangle$. This is equivalent to measuring the expectation value $\langle \hat{A}_k \rangle$ in a new state $|\psi'\rangle = \hat{U}_k |\psi\rangle$. Thus, we may apply at the end of the variational circuit the sequence of gates representing $\hat{U}_k$, and then apply the methods of the previous paragraph to evaluate $\langle \hat{\sigma}_k \rangle$. One example of $\hat{U}_k$ could be an operator applying the Hadamard gate H to each qubit corresponding to an $\hat{X}$ operator in $\hat{\sigma}_k$, and the sequence of gates HSZ to each qubit corresponding to a $\hat{Y}$ operator.

8.8 Error Analysis and Mitigation

Quantum error correction, or more often error mitigation is essential for a reliable attainment of computations on a real QPU [163-165]. The quantum measurement, an integral element of any algorithm, is by itself probabilistic. In particular, expectation values of an operator are estimated over a large number (M) of discrete measurements. On a noiseless quantum computer, the variance in the expectation value of the Hamiltonian is limited by:

$$\langle \epsilon^2 \rangle \leq \frac{\overline{\mathrm{E}^2}}{\mathrm{M}} \tag{8.11}$$

where $\overline{E^2}$ is the average of the squared energy. It defines an uncertainty and can be resolved to an arbitrary level of precision by increasing the number of measurements.

Importantly, the qubits may accumulate errors either due to the imprecise manipulation or interactions with environment. One of the major sources of errors that we have identified while collecting the data from the quantum processors is the readout error, emerging due to a certain probability that a qubit in a true $|0\rangle$ state is measured as a $|1\rangle$ or vice versa. Repeated measurements of prepared $|0\rangle$ or $|1\rangle$ states reveal transition rates w_{01} and w_{10}, defined as the probability that $|0\rangle$ is erroneously measured as $|1\rangle$, or $|1\rangle$ is measured as $|0\rangle$, respectively. Moreover, the application of a particular circuit element may result in an imperfect transformation of the qubit state. The so-called gate noise is typically classified as a separate source of error but for the purpose of this study we have assumed it to be intrinsic to the readout error.

The procedure of error mitigation is based on the computation of transition rates w_{01} and w_{10} and deriving an appropriate expression to correct the measured expectation values. In order to estimate these rates, we have explicitly prepared the state $|0\rangle$ ($|1\rangle$) $100{,}000$ times and counted how many $|1\rangle$s ($|0\rangle$s) were measured, which determines the probability that a bit flip occurs on a readout for the given computational state of each qubit. The transition rates need to be measured and updated often to ensure that the correction scheme remains effective across the duration of the trails. In fact, they are calculated every time before the optimization step is reached to take into account changes in the behavior of a specific qubit. Figure 8.5 reports the transition rates w_{01} and w_{10} evaluated for each qubit while computing band energies. The transition rates are sampled once per minute across the duration of a 50 minute

run. The rates corresponding to a flip from $|1\rangle$ to $|0\rangle$ seem to oscillate with a period of roughly 18 minutes, suggesting that environmental effects indeed modulate the behavior of qubits.

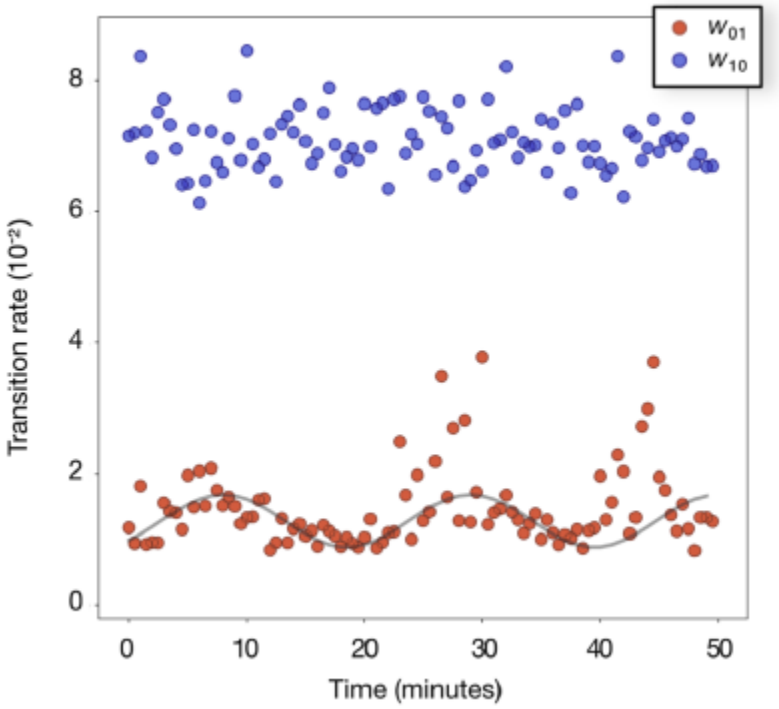


Figure 8.5: The transition rates estimated for a qubit on Rigetti's QPU. Blue circles denote the rates from state $|1\rangle$ to state $|0\rangle$, while red circles denote the rates from state $|0\rangle$ to $|1\rangle$. The fitted trend in transitions suffering from less noise is marked with a grey line. We believe these transitions to arise due to the environmental coupling.

The measured expectation value, on a single qubit, can be corrected using the following expression:

$$\langle \hat{\sigma}_c \rangle = \frac{\langle \hat{\sigma} \rangle - p^-}{1 - p^+} \tag{8.12}$$

with $p^{\pm}$ defined in terms of the transition probabilities for the single qubit, $p^{\pm} = w_{10} \pm w_{01}$. The procedure can easily be generalized to any number of qubits measured in the computational basis [166], as follows:

$$\langle Z \dots Z \rangle = \sum_{z \epsilon \mathbb{Z}_2^n} p(z) \prod_{i=1}^{n} \frac{(-1)^{z_i} - p^-}{1 - p^+} \tag{8.13}$$

where z_i is the ith element of bitstring z, and z is among the set of bitstrings of length $n(\mathbb{Z}_2^n)$. The fraction of measured bitstrings resulting in z is denoted as $p(z)$. The corrections have been successfully applied to the quantum computation of two-band electronic structure performed on Rigetti. Figure 8.2(b) shows a comparison between the corrected and uncorrected data points (closed and open circles, respectively), demonstrating that the errors have been significantly reduced.

8.9 Summary and Perspectives

In summary, we have computed the band structure of silicon along high symmetry lines in the momentum space using quantum machine accessible via cloud. In order to perform quantum simulations beyond the tractability of modern supercomputers, we need to establish methods of translating a desired physical system to the language of qubits founded with quantum logic gates. The VQE algorithm adapted from quantum chemistry seems to be suitable for electronic structure computation and remarkably, is able to leverage even minimal quantum resources, as demonstrated by the results discussed in this work. In analogy to early quantum chemistry computation tackling the problems with known analytical solutions, we have selected the electronic structure of silicon which is considered trivial in materials science. The presented studies can be thus regarded as a first step towards scalable electronic structure quantum computation that would not be limited to a specific interaction or one particular quantum system. Even though the analyzed Hamiltonian was quite simple, we are convinced that adding interactions, field effects, or corrective terms will be possible in the nearest future.

8.10 Acknowledgements

We thank Marco Fornari, Ilaria Siloi, and Virginia Carnevali for useful discussions. We acknowledge support from the US Department of Energy through the grant Q4Q: Quantum Computation for Quantum Prediction of Materials and Molecular Properties (DE-SC0019432). We are also grateful to IBM and Rigetti Computing for providing quantum resources.

CHAPTER 9

A SYSTEMATIC VARIATIONAL APPROACH TO BAND THEORY IN A QUANTUM COMPUTER*

Recent advances in qubit fidelity and hardware availability have driven efforts to stimulate molecular systems of increasing complexity in a quantum computer and motivated us to design quantum algorithms for solving the electronic structure of periodic crystalline solids. To this effect, we present a hybrid quantum-classical algorithm based on Variational Quantum Deflation [Higgot *et al., Quantum, 2019,* ***3****, 156*] and Quantum Phase Estimation [Dobšíc̆ek *et al., Phys. Rev. A, 2007,* ***76****, 030306*(R)] to solve the band structure of any periodic system described by an adequate tight-binding model. We showcase our algorithm by computing the band structure of a simple-cubic crystal with one s and three p orbitals per site (a simple model for Polonium) using simulators with increasingly realistic levels of noise and culminating with calculations on IBM quantum computers. Our results show that the algorithm is reliable in a low-noise device, functional with low precision on present-day noisy quantum computers, and displays a complexity that scales as $\Omega(M^3)$ with the number M of tight-binding orbitals per unit-cell, similarly to its classical counterparts. Our simulations offer a new insight into the "quantum" mindset applied to solid state systems and suggest avenues to explore the potential of quantum computing in materials science.

9.1 Introduction

Band structures are the fundamental toolbox of materials science in the characterization

* This chapter is reproduced with permission from The Royal Society of Chemistry, from the paper: *A systematic variational approach to band theory in a quantum computer* by Kyle Sherbert, Frank T. Cerasoli, and Marco Buongiorno Nardelli, submitted to the journal Physical Chemistry Chemical Physics in April 2021.

and discovery of the electronic properties of crystalline solids. Such periodic systems are typically considered infinite in extent and thus appear to require a large number of resources to adequately approximate. Band theory resolves this difficulty by adopting the single-electron approximation, in which each electron independently interacts with an effective potential produced by all other electrons and atomic centers. Under this approximation, a periodic Hamiltonian becomes separable in reciprocal space, reducing the system at any particular momentum $\vec{k}$ to the complexity of a single unit cell. In this way, the eigenstates of an electron with momentum $\vec{k}$ can be efficiently calculated in a classical computer; the energies of each eigenstate along a path through reciprocal space from the band structure of the material. Integrating the band structure provides early insight into structural, electronic, optical, and thermal properties of a material [167].

Band structures are an extremely useful tool but are limited by the single-electron approximation. Exchange and correlation effects can be treated through *ad hoc* correction terms to the effective field, but these methods fail when applied to highly-correlated systems. Quantum computers have the potential to surpass band theory by imposing fermionic statistics onto qubits and including electron correlation terms directly in the system Hamiltonian. For example, Quantum Phase Estimation (QPE) is a quantum algorithm which extracts eigenstate energies from a simulated system on a noise-resilient quantum computer at a low cost of computational resources [168,169]. However, quantum circuits designed for the QPE algorithm tend to require longer coherence times than are available in the era of Noisy Intermediate-Scale Quantum (NISQ) devices, so recent efforts focus on hybrid algorithms which balance quantum and classical resource costs. In particular, a more popular algorithm for molecular

ground-state energy calculations is the Variational Quantum Eigensolver (VQE) [170-181]. Many variants of VQE have arisen in recent literature, including methods such as Variational Quantum Deflation (VQD) capable of exploring excited states [173].

These algorithms were originally developed for molecular simulation, but they are readily applicable to materials by adopting the tight-binding framework in which the periodic system is expanded in a basis of suitably constructed atomic orbitals [167,173]. One may simulate a periodic system of N unit cells, each consisting of M orbitals (see for example Cade et al. [175]), where M is typically comparable to the number of orbitals considered in a single molecular simulation, but N is large enough to approximate infinity. Alternatively, one may adopt a plane-wave basis, for which a quantum circuit is available to efficiently diagonalize the kinetic and potential operators directly [176]. In either approach, the size of the basis, and therefore the number of qubits, must be very large to accurately represent the periodic system. The quantum resources required to simulate a material will thus tend to be many times larger than those required for a molecule, and generally larger than the size of quantum computers available today.

In this work, we offer an easier transition to adapt quantum algorithms for materials science, by implementing correlation-free band structure calculations on NISQ-era quantum computers. We show how the single-electron approximation accommodates a systematic approach to apply VQE to any periodic system. We do not expect our approach to offer any quantum advantage; rather, our purpose is to help materials scientists think the *quantum* way, motivating new, resource-efficient approaches to studying highly correlated systems. By considering the simplest available model, we can provide lower bounds on resource complexity

and give insight into the practical difficulties materials scientists may expect when implementing quantum algorithms.

We have previously considered this topic, employing a VQE-base algorithm to iteratively calculated the band energies of a tight-binding silicon model (chapter 8). We now apply recent developments in the literature [173,177,178] to extend and improve upon our previous work. In section 9.2, we briefly outline the essential ideas we have taken from the literature. In section 9.3, we present our robust procedure for accurately calculating the band structure of any periodic system with a quantum computer. In section 9.4, we demonstrate our procedure applied to a simple-cubic lattice, presenting data from a quantum simulator and preliminary results from IBM's **ibmq_athens** and **ibmq_santiago** cloud devices. In section 9.5, we discuss the algorithmic complexity of our procedure and highlight the steps which may or may not be improved in later work.

9.2 Background

In this section we briefly outline some essential techniques actively studied in the quantum computing literature. In particular, while QPE provides a robust strategy for measuring eigenenergies with minimal classical resources, its performance suffers greatly form the imperfect fidelity of NISQ devices. As such, we will focus mostly on VQE and its close cousin VQD when measuring eigenspectral, applying QPE when available as an optional refinement.

9.2.1 Quantum Phase Estimation

In the QPE algorithm [168,169], a set of qubits (the "state register") are first prepared into an eigenstate $|\psi\rangle$ of a unitary operator $\widehat{U}$, such that $\widehat{U}|\psi\rangle = e^{2\pi i\phi}|\psi\rangle$. The unitary

operator U is then repeatedly applied as a controlled quantum circuit so that the phase shift ϕ is encoded into another set of qubits (the "readout register"). Measurements on the readout register give the binary expansion of ϕ. In molecular simulations one selects $\widehat{U} \equiv \exp(i\widehat{H}\tau)$, the operator which evolves a system with Hamiltonian $\widehat{H}$ by a unit time τ. If one first transforms $\widehat{H}$ to guarantee that all possible energies E fall within the interval $[0,2\pi/\tau)$, the measured phases ϕ map directly onto an eigenstate $|\psi\rangle$ and its eigenvalue ϕ has an exact binary expansion, but it retains some probability of success when both conditions are relaxed. Thus, QPE can be adapted to discover eigenstates and eigenvalues *a priori*, at the cost of additional rounds of measurement.

Generally, $\widehat{H}$ is given as a weighted sum of non-commuting Pauli words (section 9.3.1). An exact circuit for $\widehat{U} \equiv \exp(i\widehat{H}\tau)$ is not readily available, but can be closely approximated by Suzuki-Trotter expansion, which factors $\widehat{U}$ into many small-time slices [179]. The number of time slices scales polynomially with the accuracy required, and the depth of each time slice depends on the number of commuting groups in $\widehat{H}$. For this reason, QPE is extremely susceptible to errors arising from the low gate fidelity and short coherence times which plague NISQ devices. Alternative approaches scale more favorably with error at the cost of ancilla qubits [180,181].

9.2.2 Variational Quantum Eigensolver

In the VQE algorithm [170,171], one begins with a Hamiltonian $\widehat{H}$, represented as a weighted sum of non-commuting Pauli words, and a parameterized quantum circuit $\hat{V}(\vec{\theta})$, the "ansatz", which prepares a set of qubits into an arbitrary state (section 9.3.2). The ansatz is

applied on an ensemble of states such that qubit measurements give the expectation value of each Pauli word in $\widehat{H}$. The weighted sum of expectation values gives the energy $E(\theta)$ of the arbitrary state prepared – this procedure is called "operator estimation" [171,182]. The parameters θ are varied by a classical optimization routine until E is minimized. According to the variational principle, this minimum is exactly the ground-state energy of the system when the ansatz $\widehat{V}(\vec{\theta})$ is robust (ie. It spans the full Hilbert space of the system), and if the classical optimization succeeds in producing the global minimum.

The algorithmic complexity of VQE depends on several factors. Measuring the expectation value of a Pauli word is a stochastic process, requiring a large number of measurements on the order of $O(\epsilon^{-2})$ for an acceptable sampling noise ϵ. These ensembles are usually measure for each Pauli word in $\widehat{H}$, although recent advances reduce the sive of the ensemble by simultaneously measuring each commuting group of Pauli words [183] or by "classically shadowing" [184] the quantum circuit to require only a logarithmic number of measurements. Like in QPE, the efficiency of the algorithm is determined by the complexity of $\widehat{H}$, and every element of the ensemble requires a unique application of the ansatz, meaning that the circuit depth and gate count should be kept minimal. The dimension of the ansatz also determines the efficiency and efficacy of the classical optimization. For all these reasons, VQE tends to be impractical for perfectly robust ansatze, and much of the literature focuses on methods for constructing effective ansatze accounting for system symmetries and hardware limitations [177,185-188]. Because circuit depth and gate count are kept low, VQE is well-suited to NISQ devices.

9.2.3 Variational Quantum Deflation

Variational Quantum Deflation (VQD) [173] is one approach for extending the VQE algorithm to explore excited states in addition to the ground-state. VQD begins as a typical VQE run to locate the ground-state, and the ground-state parameterization $\vec{\theta}_0$ is recorded. The variational process is then repeated with an additional term in the optimization routine's cost function, which gives the overlap between the current ansatz and the ground-state, weighted by a factor β. States similar to the ground-state will be shifted into a higher effective energy, so that the optimization routine considers them unfavorable. Meanwhile, higher-energy eigenstates must be orthogonal to the ground-state, so their overlap contribution will be zero. Therefore, the next lowest energy that can be found is the first excited state. This process is repeated for each energy level, adding a new overlap term for each eigenenergy already found. Each overlap can be evaluated as the expectation value of a single commuting group of Pauli words in the Hamiltonian, so that the total number of additional measurements after finding M eigenvalues is $\Theta(M^2)$. If $\widehat{H}$ consists of $\Omega(M)$ commuting groups, measured for each of M energy levels, then the additional cost of the overlap circuits is negligible.

9.3 Method

Our objective is to calculate the band structure of a periodic system, as described by a tight-binding Hamiltonian $\widehat{H}$ of the form:

$$\widehat{H} = \sum_{\alpha,\beta} t_{\alpha\beta} c_{\alpha}^{\dagger} c_{\beta} \tag{9.1}$$

Each $c_j^{\dagger}$ and c_j represent a creation and annihilation (ladder) operator on an atomic orbital ϕ_j, centered on a coordinate $\vec{r_j}$ in the crystal. The hopping parameters $t_{\alpha,\beta}$ denote the energy cost

of an electron transition from orbital ϕ_β to orbital ϕ_α. They are calculated from the overlap integrals between each pair of orbitals ϕ_α and ϕ_β, or they are selected to fit empirical observations. A general tight-binding Hamiltonian may also include multi electron correlations such as $t_{\alpha\beta\gamma\delta}c_\alpha^\dagger c_\beta^\dagger c_\gamma c_\delta$, but we neglect these terms in this work.

Our strategy is to transform equation 9.1 into reciprocal space and to apply VQD to solve for each eigenenergy at each momentum $\vec{k}$ along the desired path through reciprocal space. When sufficient quantum resources are available, we refine each band energy with QPE. Our procedure for mapping a single-electron periodic system onto a set of qubits is derived in section 9.3.1. The variational ansatz we have selected, suitable for any band structure calculation, is described in section 9.3.2. Details of implementing the quantum algorithm are presented in section 9.3.3. Finally, we provide a step-by-step schematic of our algorithm and its relation to VQE in figure 9.2.

9.3.1 Qubit Mapping

The Hamiltonian in equation 9.1 consists of ladder operators acting on atomic orbitals. The Hamiltonians appearing in the quantum algorithms of section 9.2 consist of Pauli words acting on qubits. We define a "Pauli word" $\widehat{P_i}$ as an operator acting independently on each qubit with either the identity $\hat{I}$ or one of the Pauli spin matrices $\hat{X}$, $\hat{Y}$, $\hat{Z}$. Pauli words are a natural choice for representing physical operators in a quantum computer because their expectation values can be readily measured and their unitary time evolution $exp\left(i\hat{P}_i t\right)$ can be readily implemented as a quantum circuit [189]. Our goal in this section is to map our atomic orbitals onto a qubit basis, and our Hamiltonian to a weighted sum of Pauli words:

$$\widehat{H} = \sum_{\alpha}\sum_{\beta} t_{\alpha\beta} c_{\alpha}^{\dagger} c_{\beta} \rightarrow \sum_{\mathrm{i}} \mathrm{a_i} \hat{P}_i \tag{9.2}$$

9.3.1.1 Qubit Basis

The simplest conceivable mapping between orbitals and qubits is to identify each orbital with its own qubit. The qubit state $|1\rangle$ represents an occupied orbital, while $|0\rangle$ is empty. There are however an infinite number of orbitals in an infinite crystal, and quantum computers with an infinite number of qubits are beyond our engineering capabilities. We therefore reinterpret $\widehat{H}$ as the Hamiltonian of an arbitrarily large supercell with periodic boundary conditions, consisting of N unit cells, each with M orbitals.

$$\widehat{\mathrm{H}} = \sum_{\vec{\nu}}^{\mathrm{N}} \sum_{\vec{\nu}'}^{\mathrm{N}} \sum_{\alpha=0}^{\mathrm{M-1}} \sum_{\beta=0}^{\mathrm{M-1}} t_{\alpha\beta}^{\vec{\delta}} c_{\vec{\nu}'\alpha}^{\dagger} c_{\vec{\nu}\beta} \tag{9.3}$$

Hopping parameters are now dependent on the orbitals α, and β and the displacement vector $\vec{\delta} \equiv \vec{\mathrm{r}}_{\vec{\nu}'\alpha} - \vec{\mathrm{r}}_{\vec{\nu}\beta}$ between their atoms. As $\vec{\delta}$ increases, $t_{\alpha\beta}^{\vec{\delta}}$ tends to vanish, permitting a nearest-neighbor approximation in which one considers only the smallest $\vec{\delta}$.

Equation 9.3, when supplemented with two-electron correlation terms, is the form typically considered when applying quantum algorithms to periodic systems, requiring a total of MN qubits. In the single-electron approximation, however, we can reduce the size of the system to only M qubits by transforming into reciprocal space. Reciprocal space orbitals are characterized by their own ladder operators $\tilde{c}_{\vec{k}j}^{\dagger}$ and $\tilde{c}_{\vec{k}j}$, related to $c_{\vec{\nu}'\alpha}^{\dagger}$ and $c_{\vec{\nu}\beta}$ by Fourier transform:

$$c_{\vec{\nu}'\alpha}^{\dagger} = \frac{1}{\sqrt{N}} \sum_{\vec{k}'} e^{i\vec{k}'\cdot\vec{r}_{\vec{\nu}'\alpha}} \tilde{c}_{\vec{k}'\alpha}^{\dagger} \tag{9.4a}$$

$$c_{\vec{v}\beta} = \frac{1}{\sqrt{N}} \sum_{\vec{k}} e^{-i\vec{k}\cdot\vec{r}_{\vec{v}\beta}} \tilde{c}_{\vec{k}\beta} \tag{9.4b}$$

Substituting equations 9.4 into equations 9.3, we obtain

$$\hat{H} = \sum_{\vec{k}}^{N} \sum_{\vec{k}'}^{N} \sum_{\alpha=0}^{M-1} \sum_{\beta=0}^{M-1} \left(\frac{1}{N} \sum_{\vec{v}} \sum_{\vec{v}'} t_{\alpha\beta}^{\vec{\delta}} \, e^{i\left(\vec{k}'\cdot\vec{r}_{\vec{v}'\alpha} - \vec{k}\cdot\vec{r}_{\vec{v}\beta}\right)} \tilde{c}_{\vec{k}'\alpha}^{\dagger} \tilde{c}_{\vec{k}\beta} \right) \tag{9.5}$$

We simply this by recalling $\vec{r}_{\vec{v}'\alpha} = \vec{r}_{\vec{v}\beta} + \vec{\delta}$. Then $\vec{r}_{\vec{v}'\alpha}$ becomes a common factor of each $\vec{k}$ in the exponential, and we may exploit the orthogonality relation $\frac{1}{N}\sum_{\vec{v}'} e^{i(\vec{k}'-\vec{k})\cdot\vec{r}_{\vec{v}'\alpha}} = \delta_{\vec{k}'\vec{k}}$. Summing over $\delta_{\vec{k}'\vec{k}}$, we obtain $\hat{H} = \sum_{\vec{k}} \hat{H}_{\vec{k}}$, where

$$\hat{H}_{\vec{k}} \equiv \sum_{\alpha=0}^{M-1} \sum_{\beta=0}^{M-1} H_{\alpha\beta}\left(\vec{k}\right) \tilde{c}_{\vec{k}\alpha} \tilde{c}_{\vec{k}\beta} \tag{9.6}$$

$$H_{\alpha\beta}\left(\vec{k}\right) \equiv \sum_{\vec{\delta}} t_{\alpha\beta}^{\delta} e^{i\vec{k}\cdot\vec{\delta}} \tag{9.7}$$

Each momentum $\vec{k}$ contributes an independent subsystem with only M orbitals, whose eigenenergies may be solved independently. Classically, the values $H_{\alpha\beta}\left(\vec{k}\right)$ in equation 9.7 form an $M \times M$ Hermitian matrix whose eigenvalues can be efficiently calculated with standard linear algebraic techniques in $\Theta(M^3)$ time. This work instead considers how to calculate these eigenvalues the "quantum" way.

We focus on a specific $\hat{H}_{\vec{k}}$ for the remainder of this section, with the understanding that our procedure must be repeated for each momentum $\vec{k}$ along the path of interest in reciprocal space. Equation 9.3 has a form very similar to equation 9.1, except that it acts on the reciprocal-space orbitals rather than atomic orbitals. We therefore adopt a "reciprocal-orbital" basis, in which each reciprocal-space orbital is identified with its own qubit.

9.3.1.2 Hamiltonian Mapping

Having transformed our Hamiltonian into reciprocal space (equation 9.6), we must now consider mapping each ladder operator to a set of Pauli words. The ladder operators must satisfy the following:

$$\tilde{c}|0\rangle = 0 \qquad \tilde{c}|1\rangle = |0\rangle \tag{9.8a}$$

$$\tilde{c}^{\dagger}|0\rangle = |1\rangle \qquad \tilde{c}^{\dagger}|1\rangle = 0 \tag{9.8b}$$

Meanwhile, the Pauli spin operators $\hat{X}$, $\hat{Y}$, $\hat{Z}$ act on a qubit's basis states in the following way:

$$\hat{X}|0\rangle = |1\rangle \qquad \hat{X}|1\rangle = |0\rangle \tag{9.9a}$$

$$-i\hat{Y}|0\rangle = |1\rangle \qquad i\hat{Y}|1\rangle = |0\rangle \tag{9.9b}$$

$$\hat{Z}|0\rangle = |0\rangle \qquad -\hat{Z}|1\rangle = |1\rangle \tag{9.9c}$$

It is easy to verify that the following mapping suffices for a single qubit:

$$\tilde{c} \rightarrow \frac{1}{2}\left(\hat{X} + i\hat{Y}\right) \tag{9.10a}$$

$$\tilde{c}^{\dagger} \rightarrow \frac{1}{2}\left(\hat{X} - i\hat{Y}\right) \tag{8.10b}$$

In multi-electron systems, one typically adopts the Jordan-Wigner transformation, which retains the form of equations 9.10 but appends a $\hat{Z}$ operation on $\Theta(M)$ other qubits to enforce fermionic antisymmetry. Alternatively, one may adopt the Bravyi-Kitaev transformation, which requires operations on only $\Theta(logM)$ qubits but uses a non-intuitive basis and involves non-adjacent interactions more difficult to simulate on certain qubit architectures. We refer the reader to Seely et al. [189] for an excellent introduction to both transforms. However, because we are considering single-electron systems, there *are* no other fermions to exchange with, and we may use equations 9.10 directly, so that each ladder operator acts on only $\Theta(1)$ qubits.

We may rewrite equation 9.6 to exploit the Hermiticity of $\widehat{H}_{\vec{k}}$.

$$\widehat{H}_{\vec{k}} = \sum_{\alpha} H_{\alpha\alpha} c_{\alpha}^{\dagger} c_{\alpha} + 2 \sum_{\alpha} \sum_{\beta>\alpha} Re\left(H_{\alpha\beta} c_{\alpha}^{\dagger} c_{\beta}\right) \tag{9.11}$$

Since the transpose term $H_{\beta\alpha} c_{\beta}^{\dagger} c_{\alpha} = \left(H_{\alpha\beta} c_{\alpha}^{\dagger} c_{\beta}\right)^{\dagger}$. Applying equations 9.10 and noting $\hat{X}^2 = \hat{Y}^2 = \hat{I}$, $-i\hat{X}\hat{Y} = i\hat{Y}\hat{X} = \hat{Z}$:

$$\begin{aligned} \widehat{H} \rightarrow \frac{1}{2} \sum_{\alpha} H_{\alpha\alpha} \left(\hat{I} - \hat{Z}_{\alpha}\right) + \frac{1}{2} \sum_{\alpha} \sum_{\beta>\alpha} Re\left(H_{\alpha\beta}\right) \left(\hat{X}_{\alpha} \hat{X}_{\beta} + \hat{Y}_{\alpha} \hat{Y}_{\beta}\right) \\ + \frac{1}{2} \sum_{\alpha} \sum_{\beta>\alpha} Im\left(H_{\alpha\beta}\right) \left(\hat{Y}_{\alpha} \hat{X}_{\beta} - \hat{X}_{\alpha} \hat{Y}_{\beta}\right) \end{aligned} \tag{9.12}$$

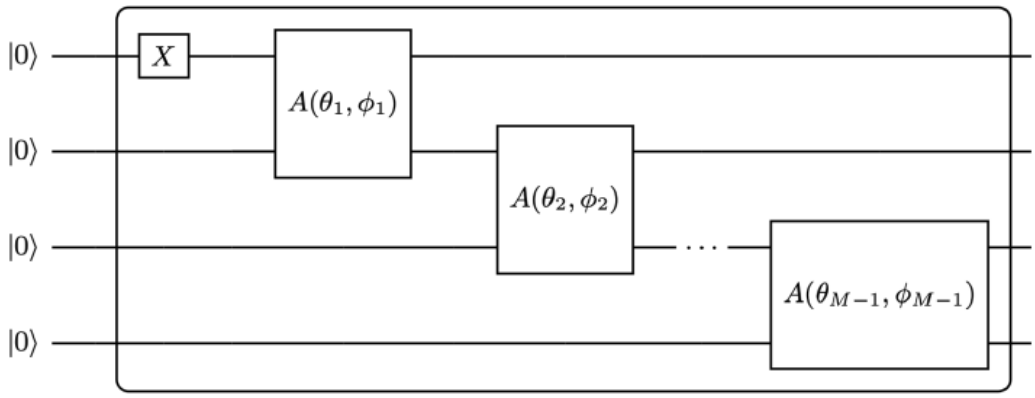
Equation 9.12 provides the weighted sum of Pauli words required in the quantum algorithms of section 9.2.

Equation 9.12 consists of $\Theta(M^2)$ Pauli words. The complexity of each algorithm in section 9.2 is determined in part by the number of commuting groups in $\widehat{H}$. In equation 9.12, all terms of the form $\hat{Z}_{\alpha}$, $\hat{X}_{\alpha}\hat{X}_{\beta}$, and $\hat{Y}_{\alpha}\hat{Y}_{\beta}$ each form commutative groups. Therefore, when $\widehat{H}_{\vec{k}}$ has no imaginary part, the energy can be determined with just 3 rounds of measurement. When $\widehat{H}_{\vec{k}}$ does have an imaginary part, we note that for fixed α, $\hat{Y}_{\alpha}\hat{X}_{\beta>\alpha}$ and $\hat{X}_{\alpha}\hat{Y}_{\beta>\alpha}$ each form commutative groups, so in general we have $\Theta(M)$ commuting groups. Finally, we note that each of these commuting groups are *qubit-wise* commutative, meaning that each index of all Pauli words in the set has either the same spin operator or the identity. This accommodates a particularly simple procedure for measuring expectation values of each set simultaneously, requiring no additional overhead in the measurement circuit.

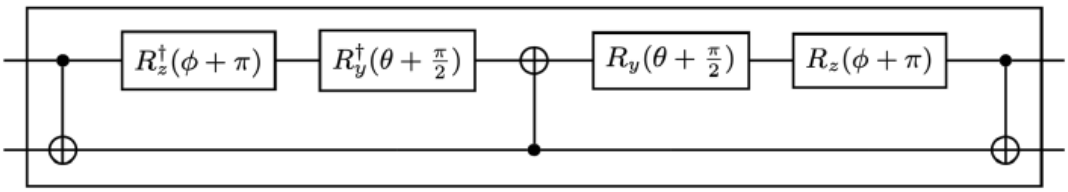
9.3.2 Ansatz

The VQE and VQD algorithms require an ansatz – a parameterized quantum circuit $\hat{V}\left(\vec{\theta}\right)$

preparing a trial state $\Psi(\vec{\theta}) = \hat{V}(\vec{\theta})|0\rangle$ for energy measurements. A quantum circuit to span the full Hilbert space of M quibts requires $2(2^M - 1)$ parameters, and it will not generally have an efficient decomposition into one- and two-qubit gates. However, most applications to molecular simulation consider a system with fixed number of electrons. In the orbital basis, or in our reciprocal orbital basis, one need only consider that subset of Hilbert space spanned by the basis states whose Hamming weights match the number of electrons in our system. For example, in band structure calculations we consider just one electron, so we need only consider the space spanned by $|10\ldots\rangle$, $|010\ldots\rangle$, etc…



(a) The ansatz $\hat{V}(\vec{\theta})$



(b) The particle-number preserving $A(\theta, \phi)$ gate from Gard et al. [177]

Figure 9.1: (a) The ansatz $\hat{V}(\vec{\theta})$ suitable for any band structure calculation. (b) One constituent $A(\boldsymbol{\theta}, \boldsymbol{\phi})$ gate. Each qubit is initialized in the $|\mathbf{0}\rangle$ state; the output is an arbitrary superposition of states with a single qubit in the $|\mathbf{1}\rangle$ state.

Gard et al. [177] provide a procedure for generating variational ansatz which conserve particle number, which is particularly simple when the particle number is 1. We begin with M

qubits labeled 0 through $M-1$ in the state $|0\rangle$. First, we apply an $\hat{X}$ gate to qubit 0, to set our ansatz with a single filled orbital. Then we apply the entangling parameterized A gate [177] such that each qubit is entangled directly or indirectly with qubit 0 (see figure 9.1). This ansatz requires $M-1$ A gates, each contributing two independent parameters, for a total of $\Theta(M)$ gates and parameters. The circuit is compatible with any quantum architecture exhibiting linear qubit connectivity and has a depth of $\Theta(M)$. Alternatively, in a fully connected device, the A gates could be applied with a "divide-and-conquer" strategy, reducing the circuit depth to $\Omega(logM)$.

Rather than assigning each orbital to its own qubit, we *could* assign each orbital to an individual basis state, requiring only $\Theta(logM)$ qubits total. This is the approach of chapter 8. While this is more efficient in the number of qubits, it must explore states with an arbitrary Hamming weight. The number of parameters required to span the space of interest is unchanged, and a suitable Hamiltonian mapping will generally form the maximum number $3^{log_2 M} = M^{log_2 3}$ of commuting sets, and is more difficult to reduce based on symmetries in the Hamiltonian (for example our observations in section 9.3.1 that a real $\widehat{H}_{\vec{k}}$ results in $\Theta(1)$ rounds of measurements).

9.3.3 Band Structure Calculations

With our ansatz $\hat{V}(\vec{\theta})$ (figure 9.1) and qubit Hamiltonian $\widehat{H}_{\vec{k}}$ (equations 9.7 and 9.12) prepared, we are ready to implement VQD for each momentum $\vec{k}$ along a path through reciprocal space. This path is usually constructed from high-symmetry segments in the crystal's First Brillouin Zone, because this proves sufficient to calculate many properties of interest. As

briefly described in section 9.2, the idea is to vary the trial state prepared by our ansatz until the energies $E \equiv \langle \hat{H} \rangle$ is minimized. We repeat the optimization for each band energy, adding additional terms to the cost function proportional to the overlap between the trail state and each previously found eigenstate, weighted by the constant factor β.

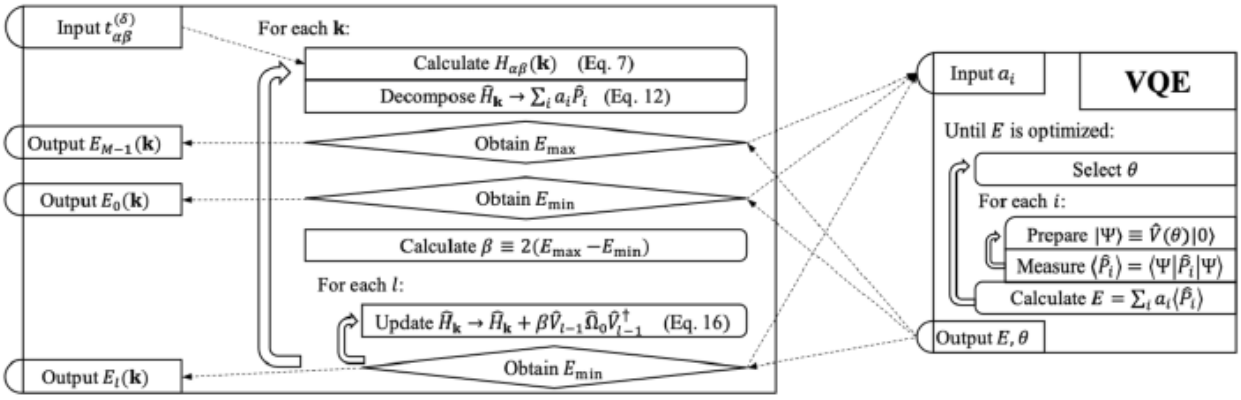


Figure 9.2: A schematic of our algorithm and its relation to VQE. Our algorithm takes tight-binding parameters $\boldsymbol{t_{\alpha\beta}^{(\vec{\delta})}}$ as input and outputs each band energy $\boldsymbol{E_l(\vec{\theta})}$. Optionally, each band energy may be refined with QPE. The operator $\boldsymbol{\hat{\Omega}_0}$ is the sum of all Pauli words spelled with letters $\boldsymbol{\hat{I}}$ and $\boldsymbol{\hat{Z}}$. The operator $\boldsymbol{\hat{V}}$ is the quantum circuit presented in figure 9.1.

The expectation values $\langle \hat{H} \rangle$ of a generic observable cannot be directly measured in the quantum computer. Rather, the expectation value of each Pauli word $\hat{P}_i$ are measure dindependently, and the energy is evaluated from the weighted sum $\langle \hat{H} \rangle = \sum_{\mathrm{i}} \mathrm{a_i} \langle \hat{\mathrm{P}}_i \rangle$, with weights a_i taken from equation 9.12. Obtaining the Pauli expectation values $\langle \hat{\mathrm{P}}_i \rangle$ is also somewhat indirect. First, the Pauli word $\hat{P}_i$ should be transformed so that it contains only letters $\hat{I}$ or $\hat{Z}$ – let us refer to the modified Pauli word as $\hat{Q}_i$. In practice, the transformation is easily accomplished by applying a "basis rotation" gate to each qubit before measurement. Next, each qubit is measured to be in one of the two computational basis states $|0\rangle$ or $|1\rangle$. The bitstring obtained from concatenating the state of each qubit is itself an eigenstate of $\hat{Q}_i$, with

eigenvalue $+1$ or -1. This procedure is applied to a large *ensemble* of qubits, each prepared independently with the ansatz and basis rotation gates. The expectation value $\langle \hat{P}_i \rangle$ is the average of all the eigenvalues of $\hat{Q}_i$ measured across the ensemble.

The ensemble necessarily has a finite size S, introducing an energy variance on the order of $\epsilon^2\ O(1/S)$. In practice, the ensemble is usually prepared in sequence, resetting a single register of qubits after each round of measurement, relegating the sampling error ϵ a parameter in the time complexity of any VQE-based algorithm. Fortunately, the same ensemble may be used to calculate the expectation values of any Pauli word which is *qubit-wise* commutative with $\hat{P}_i$. For simplicity, we assign $S = 8096$ for each commuting group in this work, although advanced methods exist which optimally distribute measurements to minimize the sampling error ϵ [182].

Many popular optimization routines (e.g. SLSQP, BFGS) are gradient-based, and they have difficulty converging to the correct value in the presence of sampling noise. Therefore, we use COBYLA, a simplex-based algorithm implemented in the **SciPy** Python package, which we have empirically noted to give good results. We randomly generate our initial guess for the parameters $\vec{\theta}$, and we use the default tolerance parameters implemented by **SciPy**. These choices are by far the simplest, but they are by no means optimal, and our results may be improved greatly by a more careful choice of optimization routine [190].

Before we can implement the deflation procedure, we must select the constant β suitable for "deflating" each band energy. We do this with a systematic procedure, first *maximizing* the energy of our system to find the highest possible energy E_{max}. We then minimize the energy to find E_0 and $\Delta \equiv E_{max} - E_0$. In theory, $\beta = \Delta$ is a sufficiently high

number to guarantee each eigenstate is projected sufficiently out of the optimization in later steps. In practice, we take $\beta = 2\Delta$ to insure against errors in the sampling and optimization process.

Higgot et al. [173] offer several strategies for computing the overlap, offering robustness against error at the const of ancilla qubits or additional optimization steps. In this work, we choose the simplest, evaluating the overlap with an eigenstate $\Psi(\vec{\theta}_l)$ by preparing the trial state $\Psi(\vec{\theta})$ and applying the adjoint circuit $\hat{V}_l^\dagger \equiv \widehat{V^\dagger}(\vec{\theta}_l)$. The probability of measuring the bitstring $0 \dots 0$ gives the overlap $\left|\langle\Psi(\vec{\theta})|\Psi(\vec{\theta}_l)\rangle\right|^2$. In practice, the probability of measuring bitstring $0 \dots 0$ is equivalent to the expectation value of an operator $\Omega_0 \equiv \sum_i \hat{Q}_i$, the sum of all unique Pauli words spelled with the letters $\hat{I}$ and $\hat{Z}$ (eg. $\hat{I}\hat{I}\hat{I}$, $\hat{I}\hat{I}\hat{Z}$, … $\hat{Z}\hat{Z}\hat{Z}$). All such operators are qubit-wise commutative and can be estimated with a single round of measurements. Therefore, we can implement the deflation procedure conveniently in the **qiskit** Python package provided by IBM, by solving for each band energy and then adding to our Hamiltonian the deflation operator $\beta\hat{V}_l\hat{\Omega}_0\hat{V}_l^\dagger$.

Initializing the Hamiltonian $\hat{H}_0 \equiv \hat{H}_{\vec{k}}$, our procedure can be formally summarized as follows:

$$\vec{\theta}_l \equiv argmin\langle 0|\hat{V}^\dagger(\vec{\theta})\hat{H}_l\hat{V}(\vec{\theta})|0\rangle \tag{9.13}$$

$$\hat{V}_l \equiv \hat{V}(\vec{\theta})|0\rangle \tag{9.14}$$

$$E_l \equiv \langle 0|\widehat{\mathrm{V}_1^\dagger}\hat{H}_l\hat{V}_l|0\rangle \tag{9.15}$$

$$\hat{H}_{l+1} \equiv \hat{H}_l + \beta\hat{V}_l\hat{\Omega}_0\hat{V}_l^\dagger \tag{9.16}$$

Each E_l we find is recorded as the energy of the lth band at momentum $\vec{k}$, and we repeat the

procedure for each $\vec{k}$ in our selected path.

Optimization routines do not always converge to the true minimum, and errors incurred early in the deflation procedure can propagate unfavorably to higher bands. Therefore, we include an optional QPE refinement to our algorithm, which applies QPE refinement to state $\Psi \equiv \hat{V}_l|0\rangle$. QPE has the effect of selecting the dominant eigenstate of Ψ_l and giving the corresponding eigenenergy with high precision. Thus, as long as the optimization procedure is "good enough", we may update our energy calculations with the result of the QPE experiment. We have used the iterative version of QPE implemented in **qiskit**. Details of the algorithm can be found in Dobšíček et al. [169].

9.4 Results

To demonstrate our procedure, we consider a basic model for a material in a simple cubic lattice structure (see figure 9.3). Each atom has s, p_x, p_y, and p_z orbitals ($M = 4$), with a large energy gap between the s and p orbitals. This may be considered a rough model for elemental Polonium, although more accurate models should take into account the relativistic effects and Coulomb interaction between orbitals located on the same atom [190,192]. The exact eigenenergies of our model at specific k-points along a high-symmetry path are calculated using standard linear algebraic techniques to diagonalize the matrix elements in equation 9.7 . We compare this band structure to the results from the quantum algorithm presented in section 9.3 in four different levels of simulation.

1. **Statevector** – quantum operations are simulated with unitary matrices, and expectation values are calculated exactly.
2. **Sampling** – expectation values are now calculated by sampling from a probability distribution.

3. **Noisy** – quantum operations and measurements are now applied with an error rate drawn from real quantum devices.

4. **Calibrated** – the same noisy simulator is used, but classical post-processing steps are applied to mitigate the error.

We also present preliminary results from IBM quantum devices.

9.4.1 Statevector Simulator

To validate our algorithm's capability of producing the correct band structure, we model the state of an n qubit system as a complex statevector and quantum operations as unitary matrices acting on the Hilbert space spanned by the 2^n dimensional basis vectors. Expectation values are evaluated analytically. Such a simulation gives the ideal behavior of a quantum computer, with perfect qubit fidelity and no sampling variance.

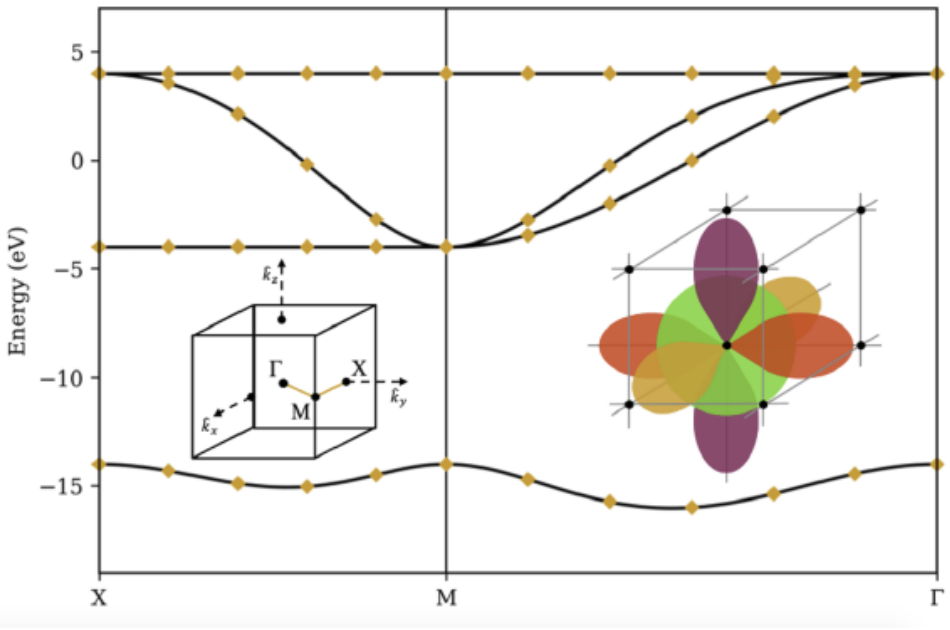


Figure 9.3: Statevector Simulator – The band structure of a simple cubic lattice with s and p orbitals (right inset) along the high-symmetry path XMΓ through the lattice's First Brillouin Zone (left inset). Solid curves denote classical (exact) diagonalization. Diamonds denote the median optimization result from applying our method on a noiseless statevector simulator 32 times with a different random seed. Bars (only visible between the third and fourth bands at nearly-degenerate momenta) denote interquartile ranges. Hopping parameters are 2 eV between adjacent s and p orbitals and 2 eV between colinear p orbitals. Each s orbital has self-energy of -14 eV to generate a large band gap.

Figure 9.3 summarizes the results of over 32 randomly-seeded optimization runs, marking the median optimization with a diamond and the interquartile range with a bar. For a few momenta where the third and fourth bands are very close together, optimization tends to locate the wrong eigenstate, giving a small variance in results. For every other point, the diamond coincides perfectly with the classical solution and the bar is absent, demonstrating that our ansatz is robust, the deflation procedure is mathematically sound, and that our choice of optimization routine (COBYLA) is generally consistent in converging to the correct values on a smooth surface.

9.4.2 Sampling Simulator

We now consider long-term viability of our procedure by retaining perfect qubit fidelity but simulating realistic measurement. The same unitary matrices as in the statevector simulator are applied to an ensemble of states, which are "measured" by sampling from the resulting probability distribution a finite number of times. While mathematically equivalent, the sampling noise resulting from the stochastic measurement process can make the energy surface bumpy, which can have a detrimental effect on the optimization step. We have selected the COBYLA optimization algorithm because it is resistant to these bumps; nevertheless, the anomalous variance observed at nearly degenerate points in the statevector simulator is now commonplace. Figure 9.4 shows our results on the noiseless qubit simulator over 32 randomly-seeded runs, clearing marking the median (asterisk) and mean (X) for both optimization (left) and QPE refinement (right). The smaller dots denote the results of individual trials.

The optimization results are extremely accurate and precise on the high-symmetry momenta but deviate slightly on the intermediate points. In fact, the high symmetry points in

this particular model each happen to have matrices $H_{\alpha\beta}(\vec{k})$ (equation 9.7) which are already diagonalized, and the resulting cost function yields a well-behaved surface which is reliably optimized, even in the presence of noise. Averaged results on the intermediate points still tend to be quite good, but individual trials can exhibit a large variance. However, the optimization does succeed in finding a point *close enough* to a correct eigenstate that the QPE refinement consistently extracts the dominant eigenvalue with a high precision. The median QPE results prove to be as accurate as is permitted by the finite binary expansion calculated by the algorithm.

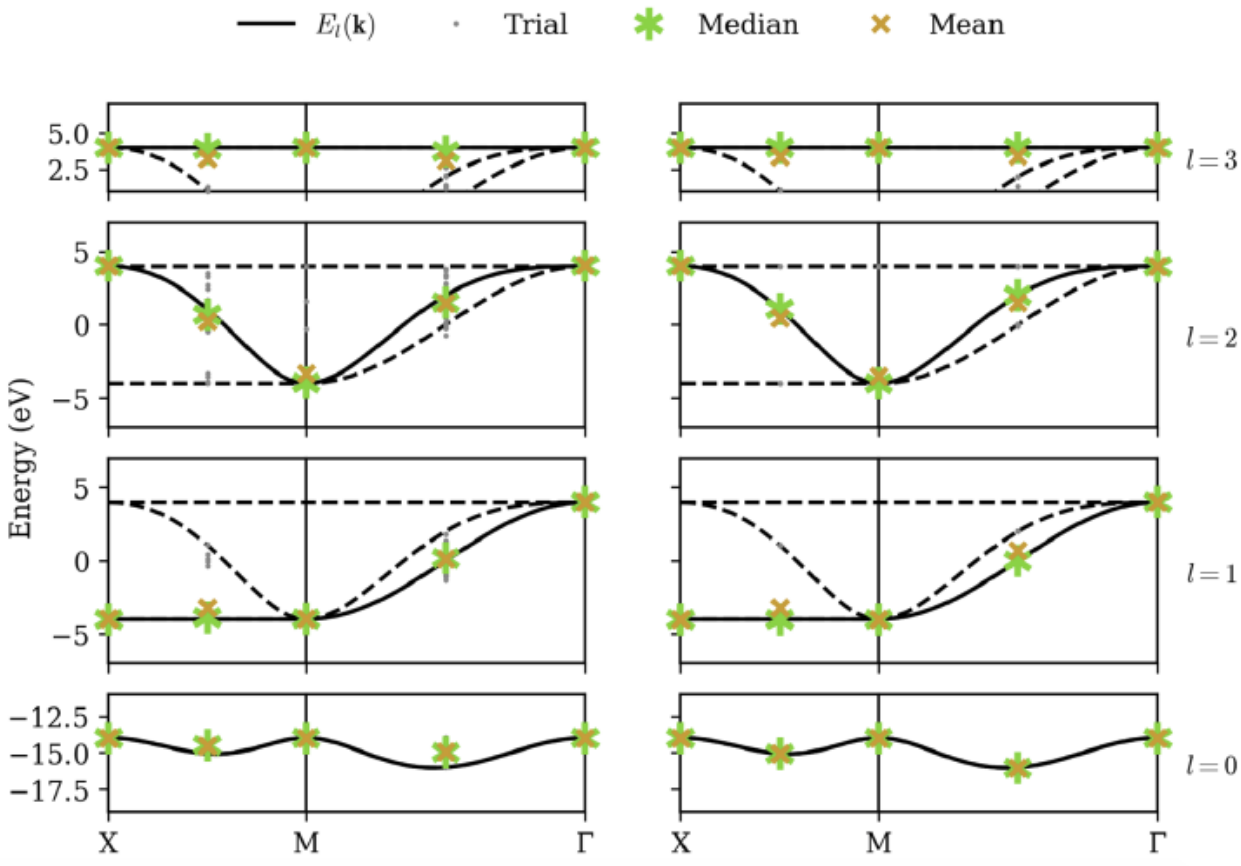


Figure 9.4: Sampling Simulator – Our method applied in the presence of sampling noise (high-fidelity qubits). The left column shows raw optimization results; the right column shows the energy obtained by QPE refinement. Gray dots denote the results from each of 32 trials, with each band given on its own row. The asterisk and X denote the median and mean, respectively.

9.4.3 Noisy Simulator

We now consider the realistic application of our procedure on present-day quantum computers, which suffer from relatively short coherence times and are vulnerable to a number of error sources. This makes practical computations extremely difficult, even in systems requiring relatively few qubits. We model error sources with a simulator by occasionally introducing "bit flips" after applying a unitary operation or when measuring a qubit.

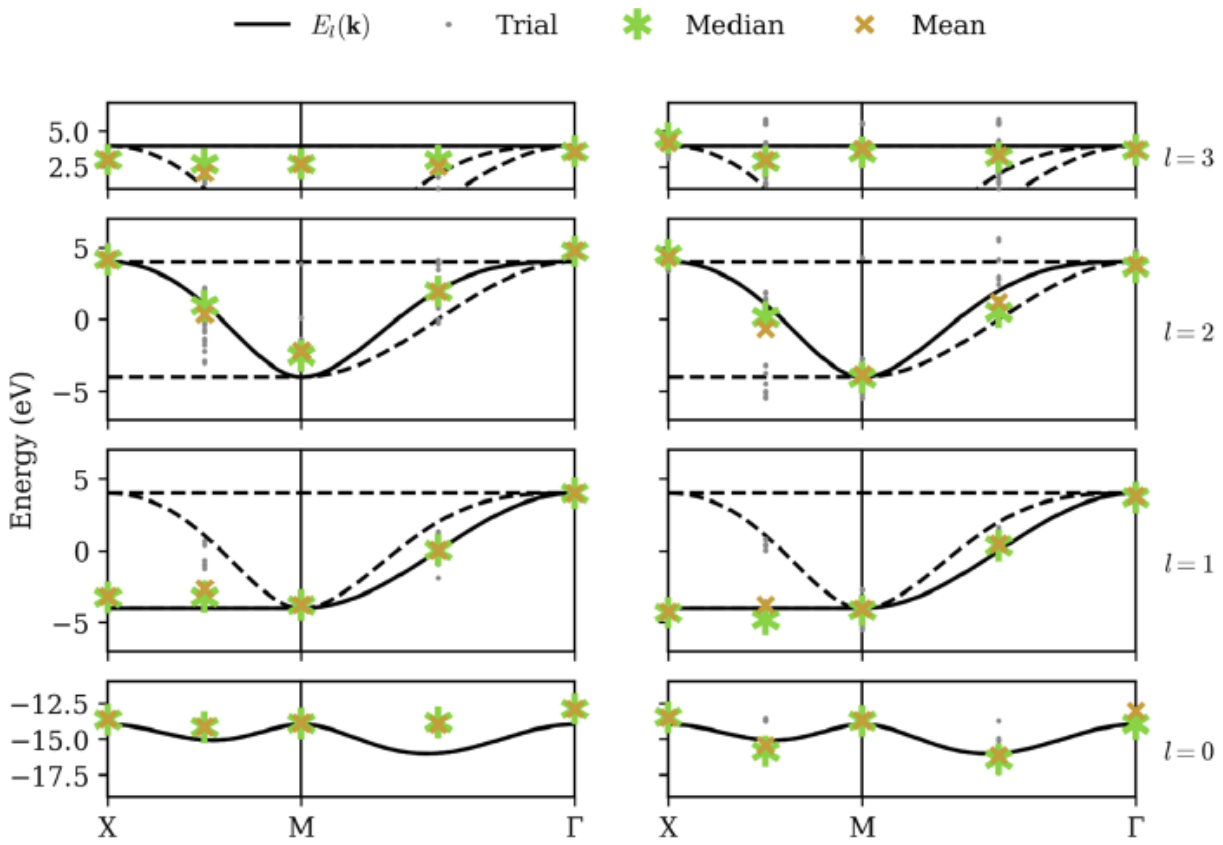


Figure 9.5: Noisy Simulator – Our method applied while simulating low-fidelity qubits, without calibration. The left column shows raw optimization results; the right column shows the energy obtained by QPE refinement. Gray dots denote the results from each of 32 trials, with each band given on its own row. The asterisk and X denote the median and mean, respectively.

Figure 9.5 shows our results on a simulator emulating the error rates characteristic of IBM's **ibmq_athens** quantum computer. Qubit noise has a clearly negative impact on the quality of results. Lowest-band optimization results tend to suffer a large systematic shift, characteristic

of coherent noise in a quantum computer. Additionally, while average QPE refinement often improves energy estimates, its results are now clustered with some variance around each nearby band, and on occasion (e.g. the third band between M and Γ) the mean optimization result is more accurate. This is symptomatic of the long circuit requirements for QPE and supports the widely-believed notion that variational algorithms are better suited to NISQ devices.

9.4.4 Calibrated Simulator

Although automated error correction procedures, based on redundant qubit registers and applied during calculation, are the most promising path toward practical quantum computation, several classical post-processing methods have already proven successful in mitigation error. Errors modeled by bit-flips in the measurement process ("readout error") can be mitigated in part or entirely, at the cost of additional calibration circuits [193]. Errors modeled by bit-flips as each unitary operation is applied ("gate error") tend to result in systematic distortions of the energy surface, as we have seen in the optimization results of figure 9.5. Theses distortions can be mitigated by applying Zero-Noise Extrapolation (ZNE) [178] in which the same measurements are repeated several times, each time modifying the circuit to incur *more* noise. These measurements can then be extrapolated to a hypothetical circuit with zero noise, using Richardson extrapolation or a similar method.

Figure 9.6 shows our results on a noisy simulator, applying readout calibration and ZNE for each energy evaluation during the optimization. ZNE offers noticeable improvement in the highest and lowest bands (calculated independently), but appears less impactful on the intermediate bands (calculated after deflation), perhaps even *increasing* variance in the third

band. This may be explained by noting that ZNE is designed to assume *systematic* error, and this is what we tend to observe when we can rely on the variational principle, where energies cannot in principle be measured below the ground-state energy.

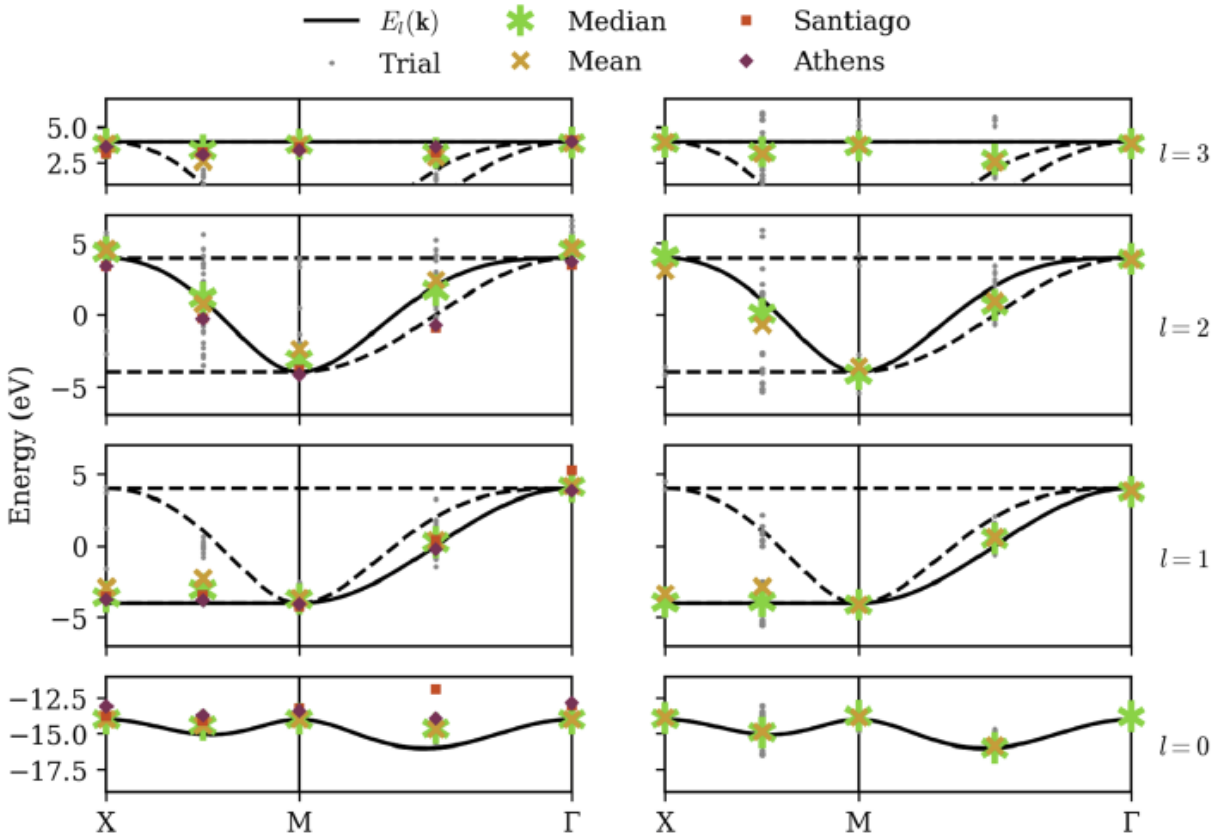


Figure 9.6: Calibrated Simulator – Our method applied while simulating low-fidelity qubits, along with rudimentary calibration. The left column shows raw optimization results; the right column shows the energy obtained by QPE refinement. Gray dots denote the results from each of 32 trials, with each band given its own row. The asterisk and X denote the median and mean, respectively. The squares and diamonds on the left denote the energies measured on quantum devices `ibmq_santiago` and `ibmq_athens` respectively, using the least-error optimization results obtained with the calibrated simulation data. Device architecture constrains the length of quantum circuits, and QPE results for either device could not be obtained.

This is not *always* true because our energy estimates the linear combinations of stochastically evaluated Pauli expectation values, and on occasion we do observe trials which appear above the highest band, but these points are relatively rare, and the average values on the highest and lowest bands are shifted inwards. However, the deflation circuits $\hat{V}_0$ are somewhat

different for each trial, depending on exactly what eigenstate was selected for the lowest band, and this, coupled with the coherent qubit error, has the effect of inducing a *random* noise on the intermediate bands. This explains why the intermediate bands seem to suffer a larger variance but reasonable average values. We note that many other error mitigation techniques besides readout calibration and ZNE have been proposed in the literature, and our results can likely be improved greatly by implementing more of them. Nevertheless, the best solution to combat random error remains averaging over more and more trials.

In addition to statistics from a calibrated simulator, figure 9.6 also shows data from the IBM devices **ibmq_athens** and **ibmq_santiago**. These are calibrated energy measurements of the eigenstates given by the least-error optimization runs on the (calibrated) noisy simulator. Results are generally consistent with the simulator, but our error mitigation is evidently even less effective on real devices. Furthermore, implementing the controlled-unitary operations necessary for the QPE procedure on a linear architecture introduces and overwhelming amount of overhead in the form of additional SWAP gates, making the QPE refinement part of our algorithm completely intractable on these devices.

9.5 Discussion

We have presented an application of VQD to calculate the band structure of a periodic system. This algorithm is hypothetically successful in producing accurate results on a device with low noise and is functional to a limited extend on current NISQ devices. In this section, we carefully analyze the complexity of the algorithm. The classical approach to band strcutre includes up to equation 9.7, at which point the calculated values $H_{\alpha\beta}(\vec{k})$ are arranged into a Hermitian matrix. The matrix can be diagonalized using row-reduction or a similar technique in

$\Theta(M^3)$ steps, where M is the number of atomic orbitals per unit cell. This is the standard against which we must compare our quantum algorithm.

Quantum resources are employed in the VQD phase of our algorithm during the operator estimation procedure, for every evaluation of the energy $E \equiv \langle \widehat{H}_{\vec{k}} \rangle$. Each application of the ansatz from figure 9.1 requires M qubits, $\Theta(M)$ entangling gates, and has a depth between $\Theta(logM)$ and $\Theta(M)$ layers, depending on qubit architecture. The Hamiltonian in equation 9.12 has $\Theta(M)$ commuting groups, even including additional terms from the deflation procedure (equation 9.16). "Our implementation requires an ensemble size of $O(\epsilon^{-2})$ for each commuting group in $\widehat{H}$ to obtain an expectation value accurate within ϵ, but since ϵ does not scale with M, we omit in in the present analysis. The ensemble states may be prepared sequentially, for a worst-case (linear architecture) execution time on the order of $\Theta(M^2)$. Alternatively, the ensemble states may be prepared in parallel, decreasing execution time at the cost of additional qubits. In the best case, implementing "classical shadowing" [184] reduces the number of required measurements to $\Theta(log\ M)$, and a fully-connected architecture permits a circuit depth as low as $\Theta(logM)$, bringing our algorithm into a sub-polynomial quantum resource requirement. However, the operator estimation procedure is still bounded by the number of Pauli words $\Theta(M^2)$ when measurement results are assembled into the energy $E(\vec{\theta}) = \sum_i a_i \langle \Psi(\vec{\theta}) | \hat{P}_i | \Psi(\vec{\theta}) \rangle$.

Operator estimation is repeated for each function evaluation in the optimization procedure. The number of function evaluations required depends on the optimization routine selected and the shape of the energy surface, so it is difficult to estimate. In general it may be expected to scale polynomially with the number of ansatz parameters, in our case $\Theta(M)$. Thus,

we include a factor of $\Theta(M^c)$, where $c \geq 1$ depends on the optimization. The optimization is repeated for each of M energy levels; therefore, the VQD phase of our algorithm has a total run-time on the order of $\Theta(M^{3+c})$.

An optional QPE phase may be implemented to estimate the eigenvalue to an arbitrary binary precision t [169]. The implementation of QPE we have used requires $M + 1$ qubits and $\Omega(t)$ rounds of measurement (see Dobšíček et al. [169] for a tighter bound). Each round applies a quantum circuit approximating a unitary operator $\widehat{U}_j = exp(i\widehat{H}\tau_j)$. Each time slice in the Suzuki Trotter expansion of $\widehat{U}_j$ on a linear architecture requires an entangling gate count of $\Theta(M^3)$ and a circuit depth of $\Theta(M^2)$ [194]. QPE is repeated for each of M energy levels, setting the best case run-time of the QPE phase of our algorithm on the order of $\Theta(M^3)$. Note also that the simulation time τ_j scale exponentially with the accuracy of the phase estimation procedure, and the number of time-slices must scale accordingly to maintain an accurate $\widehat{U}_j$. Thus, QPE tends to incur too much overhead for practical application on present-day NISQ devices.

Altogether, evaluating the band energies for each momentum $\vec{k}$ requires $\Omega(M^3)$ time steps, comparable to the classical approach. Even with a "perfect" optimizer in which the optimal parameters $\vec{\theta}_l$ are produced instantly ($c = 0$), the complexities of operator estimation and QPE alone exhibit the same scale as classical diagonalization and incur significantly greater overhead from the finite accuracy ϵ. While in this form band structure calculations are not a strong candidate for quantum advantage, quantum computers *are* expected to provide a superior edge when including electron correlation terms such as $t_{\alpha\beta\gamma\delta}c_\alpha^\dagger c_\beta^\dagger c_\gamma c_\delta$ in the Hamiltonian, which introduce factors of exponential complexity in the classical approach. However, such terms also appear to force us to abandon several simplifications we have made.

First, transforming into reciprocal space no longer enables H to be separated into subsystems of size M, meaning many more qubits are required to accurately simulate a periodic system. Second, considering multiple electrons forces us to adopt a qubit mapping which enforces fermionic antisymmetry, greatly increasing the number of commuting groups in the Hamiltonian. Finally, our ansatz dimension, entangling gates, and circuit depth can no longer remain linear in the number of qubits while simultaneously remaining robust. Our hope is that this work will inspire similar simplifications to those that we have made here, while remaining applicable to highly-correlated systems.

9.6 Conclusion

In this work, we have presented a systematic algorithm for evaluating band structures on a quantum computer. We have demonstrated the viability of implementing this algorithm in noiseless qubits systems, and we have demonstrated several of the difficulties faced when implementing it on present-day NISQ. Given the analogy to the classical band structure problem, our algorithm evidently generalizes to solving the eigenvalues of any Hermitian matrix. Finally, we have demonstrated how state-of-the-art quantum algorithms can be applied with drastically lower resource requirements to correlation-free materials and motivated similar approaches for highly-correlated systems less accessible to classical computing.

9.7 Acknowledgements

We thank Oliviero Andreussi, Itay Hen, Rosa Di Felice, Marco Fornari, Ilaria Siloi, Virginia Carnevali, and Anooja Jayaraj for useful discussions. We acknowledge support for the U.S. Department of Energy through the grant *Q4Q: Quantum Computation for Quantum Prediction*

of Materials and Molecular Properties (DE-SC0019432). We are also grateful to IBM for providing quantum computing hardware and software.

CHAPTER 10

CONCLUSION

The discovery of novel materials has accelerated greatly in the most recent decades. Materials scientists use computational tools to study molecules and materials that would be difficult to obtain in a lab and to study large families of materials exhibiting wide ranges of phenomena. Software packages for computing materials properties, such as Quantum ESPRESSO, AFLOW, AFLOWπ, and PAOFLOW, are used for high-throughput materials genomics and to study new phenomena in materials.

PAOFLOW is a relatively new software package, which constructs tight-binding Hamiltonians for efficient first-principles analysis of materials. In this work PAOFLOW is described thoroughly, and a user manual is provided for using its various routines and features. High throughput studies of monolayer and layered group-IV monochalcogenides reveal multiferroic properties and the ability to tune bandgaps with applied stresses and strains. PAOFLOW is also used to examine SnTe monolayers, which are found to exhibit the persistent spin helix (PSH) state. The PSH is a favorable property for creating spintronic gate transistors, and functional design of such a transistor is proposed in this work. PAOFLOW has proved itself as a valuable tool for identifying next generation materials and performing high-through case studies of material families.

Quantum computers have provided a new avenue for the modeling of molecules and materials. Harnessing the power of superposition promises efficient simulation of correlations in fermionic systems. Even in the NISQ era, where very few qubits are available with limited connectivity and low fidelity, researchers are developing methods of gaining computational

advantage over classical machines. Quantum algorithms, like QPE and UCC, are proposed for obtaining eigenspectra in Hamiltonians and for describing many body systems variationally. Hybrid quantum-classical algorithms have become popular in the most recent years, leveraging CPUs for optimization and using QPUs only when a computational advantage is viable. This work demonstrated a proof-of-concept strategy for diagonalizing tight-binding Hamiltonians for silicon, computing its band structure on a quantum processor with the VQE algorithm. Additionally, a procedure for representing LCAO systems and computing electronic structures of general solid-state systems on quantum processors is proposed and verified. Materials simulation on quantum computers is just recently gaining traction as a viable strategy, but the methods proposed here will hopefully pave the way for future studies regarding electronic structure and electronic correlations in molecules and solids.

Computational tools are invaluable for the classification and design of novel materials. They accelerate materials discovery, provide early insight to new phenomena, and enable affordable methods of experimenting with condensed matter. These computational techniques, along with development of increasingly powerful computing machines, will continue to benefit scientific and industrial organizations by providing low-cost methods of materials research and by permitting analysis of diverse materials features or phenomenon.

REFERENCES

[1] J. J. Thomson, *XL. Cathode Rays*, Philosophical Magazine Series 5, 293—316 (1897).

[2] N. W. Ashcroft, and N. D. Mermin, *Solid State Physics*, Harcourt College Publishers, (1976).

[3] D. Van Delft, and Peter Kes, *The discovery of superconductivity*, Physics Today 63, 38 (2010).

[4] R. B. Laughlin, *Anomalous Quantum Hall Effect: An Incompressible Quantum Fluid with Fractionally Charged Excitations*, Phys. Rev. Lett 50, 1395 (1983).

[5] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, *Theory of Superconductivity*, Phys. Rev. 108, 1175 (1957).

[6] P. Hohenberg, and W. Kohn, *Inhomogeneous Electron Gas*, Physical Review 136, B864 (1964).

[7] W. Kohn, L. J. Sham, *Self-Consistent Equations Including Exchange and Correlation Effects*, Physical Review 140, A1133 (1965).

[8] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. De Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. MartinSamos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, *Quantum ESPRESSO: a modular and open-source software project for quantum simulations of materials*, Journal of Physics: Condensed Matter 21, 395502 (2009).

[9] P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. Buongiorno Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A. Dal Corso, S. De Gironcoli, P. Delugas, R.A. DiStasio Jr, A. Ferretti, A. Floris, G. Fratesi, G. Fu- gallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H.-Y. Ko, A. Kokalj, E. Kucukbenli, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N.L. Nguyen, H. V. Nguyen, A. Otero-de-la-Roza, L. Paulatto, S. Ponć e, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A.P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, and S. Baroni, *Advanced capabilities for materials modelling with Quantum ESPRESSO*, Journal of Physics: Condensed Matter 29, 465901 (2017).

[10] P. W. Shore, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, J.Sci.Statist.Comput. 26, 1484 (1997).

[11] A. Robert, P. K. Barkoutsos, S. Woerner, et al. *Resource-efficient quantum algorithm for protein folding*, npj Quantum Inf 7, 38 (2021).

[12] M. Zinner, F. Dahlhause, P. Boehme, J. Ehlers, L. Bieske, and L. Fehring, *Quantum computing's potential for drug discovery: Early stage industry dynamics*, Drug Discovery Today, 1359-6446 (2021).

[13] M. Born, and J.R. Oppenheimer, *On the Quantum Theory of Molecules*, Translated by S. M. Blinder, Ann. Physik 84, 458 (1927).

[14] J. P. Perdew, K. Burke, and M. Ernzerhof, *Rationale for mixing exact exchange with density functional approximations*, American Institute of Physics 105, 9982 (1996).

[15] T. van Mourik, et al. *Density functional theory across chemistry, physics and biology, Philosophical transactions*. Series A, Mathematical, physical, and engineering sciences vol. 372, 2011 (2014).

[16] W. E. Pickett, *Pseudopotential methods in condensed matter applications*, Computer Physics Reports 9, 115-198 (1989).

[17] L. A. Agapito, A. Ferretti, A. Calzolari, S. Curtarolo, and M. Buongiorno Nardelli, *Effective and accurate representation of extended Bloch states on finite Hilbert spaces*, Phys. Rev. B 88, 165127 (2013).

[18] L. A. Agapito, S. Ismail-Beigi, S. Curtarolo, M. Fornari, and M. Buongiorno Nardelli, *Accurate tight-binding Hamiltonian matrices from ab initio calculations: Minimal basis sets*, Physical Review B 93, 035104–9 (2016).

[19] L. A. Agapito, M. Fornari, D. Ceresoli, A. Ferretti, S. Curtarolo, and M. Buongiorno Nardelli, *Accurate tight-binding Hamiltonians for two-dimensional and layered materials*, Physical Review B 93, 125137–8 (2016).

[20] S. Curtarolo, W. Setyawan, G. L. W. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko, and D. Morgan, *AFLOW: An automatic framework for high-throughput materials discovery*, Computational Materials Science 58, 218–226 (2012).

[21] S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R. H. Taylor, L. J. Nelson, G. L. W. Hart, S. Sanvito, M. Buongiorno Nardelli, N. Mingo, and O. Levy, *AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations*, Comput. Mater. Sci. 58, 227–235 (2012).

[22] A. R. Supka, T. E. Lyons, L. Liyanage, P. D'Amico, R. Al Rahal Al Orabi, S. Mahatara, P. Gopal, C. Toher, D. Ceresoli, A. Calzolari, S. Curtarolo, M. Buongiorno Nardelli, and M. Fornari, *AFLOWpi: A minimalist approach to high-throughput ab initio calculations including the generation of tight-binding hamiltonians*, Computational Materials Science 136, 76–84 (2017).

[23] L. E. Givon, T. Unterthiner, N. B. Erich- son, D. W. Chiang, E. Larson, L. Pfister, S. Dieleman, G. R. Lee, S. van der Walt, B. Menn, T. M. Moldovan, F. Bastien, X. Shi, J. Schlu¨ter, B. Thomas, C.

Capdev- ila, A. Rubinsteyn, M. M. Forbes, J. Frelinger, T. Klein, B. Merry, L. Pastewka, S. Tay- lor, F. Wang, and Y. Zhou, *scikit-cuda 0.5.1: a Python interface to GPU-powered libraries* (2015), doi:10.5281/zenodo.40565. http://dx. doi.org/10.5281/zenodo.40565.

[24] W. Setyawan and S. Curtarolo, *High-throughput electronic band structure calculations: Challenges and tools*, Comput. Mater. Sci. 49, 299–312 (2010).

[25] Y. Yao, L. Kleinman, A. H. MacDonald, J. Sinova, T. Jungwirth, D.-s. Wang, E. Wang, and Q. Niu, *First Principles Calculation of Anomalous Hall Conductivity in Ferromagnetic bcc Fe*, Physical Review Letters 92, 037204–4 (2004).

[26] G. S. Krinchik and V. A. Artem'ev, *Magneto- optical properties of Ni, Co, and Fe in the ultra-violet, visible, and infrared parts of the spectrum*, Sov Phys JETP 28, 1080.

[27] T. Kimura, Y. Otani, T. Sato, S. Takahashi, and S. Maekawa, *Room-Temperature Reversible Spin Hall Effect*, Phys. Rev. Lett. 98, 156601 (2007).

[28] G. Y. Guo, S. Murakami, T. W. Chen, and N. Nagaosa, *Intrinsic Spin Hall Effect in Plat- inum: First-Principles Calculations*, Physical Re- view Letters 100, 096401–4 (2008).

[29] M. Graf and P. Vogl, *Electromagnetic fields and dielectric response in empirical tight-binding theory*, Physical Review B 51, 4940 (1995).

[30] L. A. Agapito, S. Curtarolo, and M. Buongiorno Nardelli, *Reformulation of DFT+U as a Pseudo-hybrid Hubbard Density Functional for Accelerated Materials Discovery*, Physical Review X 5, 011006 (2015).

[31] P. Gopal, M. Fornari, S. Curtarolo, L. A. Agapito, L. S. I. Liyanage, and M. Buongiorno Nardelli, *Improved predictions of the physical properties of Zn- and Cd-based wide band-gap semiconductors: A validation of the ACBN0 functional*, Physical Review B 91, 245202 (2015).

[32] E. Abate and M. Asdente, *Tight-Binding Calculation of 3d Bands of Fe with and without Spin-Orbit Coupling*, Phys. Rev. 140, A1303–A1308
(1965).

[33] L. Fu, C. Kane, and E. Mele, *Topological insulators in three dimensions*, Physical Review Letters 98, 106803 (2007).

[34] J. R. Yates, X. Wang, D. Vanderbilt, and I. Souza, *Spectral and Fermi surface properties from Wannier interpolation*, Physical Review B 75, 195121– [25] 11 (2007).

[35] P. D'Amico, L. Agapito, A. Catellani, A. Ruini, S. Curtarolo, M. Fornari, M. Buongiorno Nardelli, and A. Calzolari, *Accurate ab initio tight-binding Hamiltonians: Effective tools for electronic transport and optical spectroscopy from first principles*, Phys. Rev. B 94, 165166 (2016).

[36] N. A. Mecholsky, L. Resca, I. L. Pegg, and M. Fornari, *Theory of band warping and its effects on thermoelectronic transport properties*, Physical Review B 89, 155131 (2014).

[37] M. Gradhand, D. V. Fedorov, F. Pientka, P. Zhan, I. Mertig, and B. L. Györffy, *First-principle calculations of the Berry curvature of Bloch states for charge and spin transport of electrons*, Journal of Physics: Condensed Matter 24, 213202-24 (2012).

[38] M. Buongiorno Nardelli, *Electronic transport in extended systems: Application to carbon nanotubes*, Physical Review B 60, 7828–7833 (1999).

[39] A. Calzolari, N. Marzari, I. Souza, and M. Buongiorno Nardelli, *Ab initio transport properties of nanostructures from maximally localized Wannier functions*, Physical Review B 69, 035108 (2004).

[40] D. S. Fisher and P. A. Lee, *Relation between conductivity and transmission matrix*, Physical Re- view B 23, 6851–6854 (1981).

[41] S. Datta, *Electronic Transport in Mesoscopic Systems* (Cambridge University Press, 1997).

[42] Vu Thi Ngoc Huyen, Yuki Yanagi, and Michi-To Suzuki, *Spin and anomalous Hall effects emerging from topological degeneracy in Dirac fermion system CuMnAs*, Phys. Rev. B (2021).

[43] Shao, Ding-Fu and Gurung, Gautam and Zhang, Shu-Hui and Tsymbal, Evgeny Y., *Dirac Nodal Line Metal for Topological Antiferromagnetic Spintronics*, Physical Review Latters 122, 077203 (2019).

[44] T. Nan, C.X. Quintela, J. Irwin, G. Gurung, D.F. Shao, J. Gibbons, N. Campbell, K. Song, S.-Y. Choi, L. Guo, R.D. Johnson, P. Manuel, R.V. Chopdekar, I. Hallsteinsen, T. Tybell, P.J. Ryan, J.-W. Kim, Y. Choi, P.G. Radaelli, D.C. Ralph, E.Y. Tsymbal, M.S. Rzchowski, and C.B. Eom, *Controlling spin current polarization through non- collinear antiferromagnetism*, Nature Communi- cations 2041-1723, 4671 (2020).

[45] McHugh, Oliver L. W. and Goh, Wen Fong and Gradhand, Martin and Stewart, Derek A., *Impact of impurities on the spin Hall conductivity in β- W*, Phys. Rev. Materials 4, 094404 (2020).

[46] Gurung, Gautam and Shao, Ding-Fu and Paudel, Tula R. and Tsymbal, Evgeny Y*., Anomalous Hall conductivity of noncollinear magnetic antiperovskites*, Phys. Rev. Materials 3, 04409 (2019).

[47] T. Hagiwara, K. Suekuni, P. Lemoine, A.R. Supka, R. Chetty, E. Guilmeau, B. Raveau, M. Fornari, M. Ohta, R. Al Rahal Al Orabi, H. Saito, K. Hashikuni, and M. Ohtaki, *Key Role of d0 and d10 Cations for the Design of Semi- conducting Colusites: Large Thermoelectric ZT in $Cu_{26}Ti_2Sb_6S_{32}$ Compounds*, Chemistry of Mate- rials 33, 3449–3456 (2021).

[48] V. Pavan Kumar, P. Lemoine, V. Carnevali, G. Gu ́elou, O.I. Lebedev, P. Boullay, B. Raveau, R. Al Rahal Al Orabi, M. Fornari, C. Prestipino, D. Menut, C. Candolfi, B. Malaman, J. Juraszek, E.

Guilmeau, *Ordered sphalerite derivative $Cu_5Sn_2S_7$: a degenerate semiconduc- tor with high carrier mobility in the Cu–Sn–S diagram*, J. Mater. Chem. A 9, 10812–10826 (2021).

[49] M. Kawamura, *FermiSurfer: Fermi-surface viewer providing multiple representation schemes*, Computer Physics Communications 239, 197-203 (2019).

[50] D. J. Campbell, J. Collini, J. Sławińska and Au- tieri, C. Autieri, L. Wang, K. Wang, B. Wilfong, Y. S. Eo, P. Neves, D. Graf, E. E. Rodrigues, N. P. Butch, M. Buongiorno Nardelli, and J. Paglione, *Topologically driven linear magnetoresistance in helimagnetic FeP*, npj Quantum Materials 6, 38 (2021).

[51] D. Gresch, G. Aut`es, O. V. Yazyev, M. Troyer, D. Vanderbilt, B. A. Bernevig, A. A. Soluyanov, *Z2Pack: Numerical implementation of hybrid Wannier centers for identifying topological materials*, Phys. Rev. B. 95, 075146 (2017).

[52] C. L. Kane and E. J. Mele, *Quantum Spin Hall Effect in Graphene*, Phys. Rev. Lett. 95, 226801 (2005).

[53] J. C. Slater and G. F. Koster, *Simplified LCAO Method for the Periodic Potential Problem*, Phys. Rev. 94, 1498–1524 (1954).

[54] R. Farris, M. B. Maccioni, A. Filippetti, and V. Fiorentini, *Theory of thermoelectricity in Mg_3Sb_2 with an energy- and temperature-dependent relaxation time*, Journal of Physics: Conference Series 1226, 012010 (2019).

[55] C. Jacoboni, *Theory of electron transport in semi- conductors: a pathway from elementary physics to nonequilibrium Green functions*, Journal of Physics: Cond. Matter 31, 065702 (2018).

[56] B. Ridley, *Polar-optical-phonon and electron-electron scattering in large-bandgap semiconductors*, (Journal of Physics: Condensed Matter) 10, 6717 (1998).

[57] W. Setyawan and S. Curtarolo, *High-throughput electronic band structure calculations: Challenges and tools*, Comput. Mater. Sci. 49, 299–312 (2010).

[58] J. E. Hirsch, *Spin Hall Effect,* Phys. Rev. Lett. 83, 1834 (1999).

[59] Y. K. Kato, R. C. Myers, A. C. Gossard, and D. D. Awschalom, *Observation of the Spin Hall Effect in Semiconductors*, Science 306, 1910 (2004).

[60] J. Wunderlich, B. Kaestner, J. Sinova, and T. Jungwirth, *Experimental Observation of the Spin-Hall Effect in a Two-Dimensional Spin-Orbit Coupled Semiconductor System*, Phys. Rev. Lett. 94, 047204 (2005).

[61] J. Sinova, D. Culcer, Q. Niu, N. A. Sinitsyn, T. Jungwirth, and A. H. MacDonald, *Universal Intrinsic Spin Hall Effect*, 2004 Phys. Rev. Lett. 92, 126603 (2004).

[62] J. Sinova, S. O. Valenzuela, J. Wunderlich, C. H. Back, and T. Jungwirth, , Rev. Mod. Phys. 87, 1213 (2015).

[63] K. Fujiwara, Y. Fukuma, J. Matsuno, H. Idzuchi, Y. Niimi, Y. Otani, and H. Takagi, *5d iridium oxide as a material for spin-current detection*, Nat. Commun. 4, 2893 (2013).

[64] Y. Yao, and Z. Fang, *Spin Changes of Intrinsic Spin Hall Effect in Semiconductors and Simple Metals: First-Principles Calculations*, Phys. Rev. Lett. 95, 156601 (2005).

[65] G. Y. Guo, *Ab initio calculation of intrinsic spin Hall conductivity of Pd and Au*, J. Appl. Phys. 105, 07C701 (2009).

[66] C. Stamm, C. Murer, M. Berritta, J. Feng, M. Gabureac, P. M. Oppeneer, and P. Gambardella, *Magneto-Optical Detection of the Spin Hall Effect in Pt and W Thin Films*, Phys. Rev. Lett. 119, 087203 (2017).

[67] L. Liu, C. -F. Pai, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman, *Spin-Torque Switching with the Giant Spin Hall Effect of Tantalum*, Science 336, 555 (2012).

[68] C. -F. Pai, L. Liu, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman, *Spin transfer torque devices utilizing the giant spin Hall effect of tungsten*, Appl. Phys. Lett. 101**,** 122404 (2012).

[69] M. Morota, Y. Niimi, K. Ohnishi, D. H. Wei, T. Tanaka, H. Kontani, T. Kimura, and Y. Otani, *Indication of intrinsic spin Hall effect in 4d and 5d*, Phys. Rev. B 83, 174405 (2011).

[70] A. S. Patri, K. Hwang, H. -W. Lee, and Y. B. Kim, *Theory of Large Intrinsic Spin Hall Effect in Iridate Semimetals*, Sci. Rep. 8, 8052 (2018).

[71] W. Feng, Y. Yao, W. Zhu, J. Zhou, W. Yao, and D. Xiao, *Intrinsic spin Hall effect in monolayers of group-VI dichalcogenides: A first-principles study*, Phys. Rev. B 86, 165108 (2012).

[72] L. Matthes, S. Kufner, J. Furthmuller,and F. Bechstedt, *Intrinsic spin Hall conductivity in one-, two- and three-dimensional trivial and topological systems*, Phys. Rev. B 94, 085410 (2016).

[73] L. C. Gomes and A. Carvalho, *Phosphorene analogues: Isoelectric two-dimensional group-IV monochalcogenides with orthorhombic structure*, Phys. Rev. B 92, 085406 (2015).

[74] L. C. Gomes, A. Carvalho, and A. H. Castro Neto, *Enhanced piezoelectricity and modified dielectric screening of two-dimensional group-IV monochalcogenides*, Phys. Rev. B 92, 214103 (2015).

[75] L. Huang, F. Wu, and J. Li, *Structural anisotropy results in strain-tunable electronic and optical properties in monolayer GeX and SnX (X = S, Se, Te)*, J. Chem. Phys. 144, 114708 (2016).

[76] X. Ding-Jiang, T. Jiahui, H. Jin-Song, H. Wenping, G. Yu-Guo, and W. Li-Jun, *Anisotropic Photoresponse Properties of Single Micrometer-Sized GeSe Nanosheet*, Adv. Mater. 24, 4528 (2012).

[77] L. Li, Z. Chen, Y. Hu, X. Wang, T. Zhang, W. Chen, andQ. Wang, *Single-Layer Single-Crystalline SnSe Nanosheets*, J. Am. Chem. Soc. 135, 1213 (2013).

[78] J. R. Brent, D. J. Lewis, T. Lorenz, E. A. Lewis, N. Savjani, S. J. Haigh, G. Seifert, B. Derby, and P. O'Brien, *Tin(II) Sulfide (SnS) Nanosheets by Liquid-Phase Exfoliation of Herzenbergite: IV-VI Main Group Two-Dimensional Atomic Crystals*, J. Am. Chem. Soc. 137, 12689 (2015).

[79] K. Chang et al., *Discovery of robust in-plane ferroelectricity in atomic-thic SnTe*, Science 353, 274 (2016).

[80] K. Chang *et al., Enhanced Spontaneous Polarization in Ultrathin SnTe Films with Layered Antipolar Structure,* Adv. Mater. 31, 1804428 (2018).

[81] R. Fei, W. Li, J. Li and L. Yang, *Giant piezoelectricity of monolayer group IV monochalcogenides: SnSe, SnS, GeSe, and GeS*, Appl. Phys. Lett. 107, 173104 (2015).

[82] H. Wang, and X. Qian, *Two-dimensional multiferroics in monolayer group IV monochalcogenides*, 2D Mater. 4, 015042 (2017).

[83] M. Wu, and X. C. Zeng, *Intrinsic Ferroelectricity and/or Multiferroicity in Two-Dimensional Phosphorene and Phosphorene Analogues*, Nano Lett. 16, 3236 (2016).

[84] G. Shi, and E. Kioupakis, *Anisotropic Spin Transport and Strong Visible-Light Absorbance in Few-Layer SnSe and GeSe*, Nano Lett. 15, 6926 (2015).

[85] C. Chowdhury, S. Karmakar, and A. Datta, *Monolayer Group IV-VI Monochalcogenides: Low-Dimensional Materials for Photocatalytic Water Splitting*, J. Phys. Chem. C 121, 7615 (2017).

[86] L. Xu, M. Yang, S. J. Wang, and Y. P. Feng, *Electronic and optical properties of the monolayer group-IV monochalcogenides MX (M = Ge, Sn; X = S, Se, Te)*, Phys.Rev.B 95, 235434 (2017).

[87] A. S. Rodin, L. C. Gomes, A. Carvalho, and A. H. Castro Neto, *Valley physics in tin (II) sulfide*, Phys. Rev. B 93, 045431 (2016).

[88] J. P. Perdew, K. Burke, and M. Ernzerhof, *Generalized Gradient Approximation Made Simple*, Phys. Rev. Lett. 77, 3865 (1996).

[89] G. Kresse, and D. Joubert, From ultrasoft pseudopotentials to the projector augmented-wave method, Phys. Rev. B 59, 1758 (1999).

[90] A. D. Corso, *Pseudopotentials periodic table: From H to Pu*, Comput. Mater. Sci. 95, 337 (2014).

[91] G. Y. Guo, Y. Yao, and Q. Niu, *Ab initio Calculation of the Intrinsic Spin Hall Effect in Semiconductors*, Phys. Rev. Lett. 94, 226601 (2005).

[92] S. Barraza-Lopez, T. P. Kaloni, S. P. Poudel, and P. Kumar, *Tuning the ferroelectric-to-paraelectric transition temperature and dipole orientation of group-IV monochalcogenide monolayers*, Phys. Rev. B 97, 024110 (2018).

[93] K. Liu, J. Lu, S. Picozzi, L. Bellaiche, and H. Xiang, *Intrinsic Origin of Enhancement of Ferroelectricity in SnTe Ultrathin Films*, Phys. Rev. Lett. 121, 027601 (2018).

[94] S. Ohya, A. Yamamoto, T. Yamaguchi, R. Ishikawa, R. Akiyama, L. D. Anh, S. Goel, Y. K. Wakabayashi, S. Kuroda, and M. Tanaka, *Observation of the inverse spin Hall effect in topological crystalline insulator SnTe using spin pumping*, Phys. Rev. B 96, 094424 (2017).

[95] J. Sławińska, H. Aramberri, M. Munoz, and J. Cerda, *Ab initio study of the relationship between spontaneous polarization and p-type doping in quasi-freestanding graphene on H-passivated SiC surfaces*, Carbon 93, 88 (2015).

[96] B. Huang et al*., Layer-dependent ferromagnetism in a van der Waals crystal down to the monolayer limit,* Nature 546, 270 (2017).

[97] C. Gong et al., Discovery of intrinsic ferromagnetism in two-dimensional van der Waals crystals, Nature 546, 265 (2017).

[98] Y. Xu, D. D. Awschalom, and J. Nitta, *Handbook of Spintronics* (Springer Publishing Company, Incorporated, 2015).

[99] S. Datta and B. Das, *Electronic analog of the electro-optic modulator*, Applied Physics Letters 56, 665 (1990).

[100] H. C. Koo, J. H. Kwon, J. Eom, J. Chang, S. H. Han, and M. Johnson, *Control of spin precession in a spin-injected field effect transistor*, Science 325, 1515 (2009).

[101] J. Wunderlich, B.-G. Park, A. C. Irvine, L. P. Zarbo, E. Rozkotova, P. Nemec, V. Novak, J. Sinova, and T. Jungwirth, *Spin Hall Effect Transistor*, Science 330, 1801 (2010).

[102] W. Y. Choi, H.-j. Kim, J. Chang, S. H. Han, A. Abbout, H. B. M. Saidaoui, A. Manchon, K.-J. Lee, and H. C. Koo, *Ferromagnet-Free All-Electric Spin Hall Transistors*, Nano Letters 18, 7998 (2018).

[103] J. Schliemann, J. C. Egues, and D. Loss, *Nonballistic Spin-Field-Effect Transistor*, Phys. Rev. Lett. 90, 146801 (2003).

[104] B. A. Bernevig, J. Orenstein, and S.-C. Zhang, *Exact SU(2) Symmetry and Persistent Spin Helix in a Spin-Orbit Coupled System*, Phys. Rev. Lett. 97, 236601 (2006).

[105] C. Betthausen, T. Dollinger, H. Saarikoski, V. Kolkovsky, G. Karczewski, T. Wojtowicz, K. Richter, and D. Weiss, *Spin-Transistor Action via Tunable Landau-Zener Transitions*, Science 337, 324 (2012).

[106] M. Kohda, V. Lechner, Y. Kunihashi, T. Dollinger, P. Olbrich, C. Schönhuber, I. Caspers, V. V. Bel'kov, L. E Golub, D. Weiss, K. Richter, J. Nitta, and S. D. Ganichev, *Gate-controlled persistent spin helix state in (In,Ga)As quantum wells*, Phys. Rev. B 86, 081306 (2012).

[107] M. Kohda and G. Salis, *Physics and application of persistent spin helix state in semiconductor heterostructures*, Semiconductor Science and Technology 32, 073002 (2017).

[108] D. Di Sante, P. Barone, R. Bertacco, and S. Picozzi, *Electric Control of the Giant Rashba Effect in Bulk GeTe*, Advanced Materials 25, 509 (2013).

[109] S. Picozzi, *Ferroelectric Rashba semiconductors as a novel class of multifunctional materials*, Frontiers in Physics 2, 10 (2014).

[110] M. Liebmann, C. Rinaldi, D. Di Sante, J. Kellner, C. Pauly, R. N. Wang, J. E. Boschker, A. Giussani, S. Bertoli, M. Cantoni, L. Baldrati, M. Asa, I. Vobornik, G. Panac- cione, D. Marchenko, J. Sanchez-Barriga, O. Rader, R. Calarco, S. Picozzi, R. Bertacco, and M. Morgenstern, *Giant Rashba-Type Spin Splitting in Ferroelectric GeTe(111)*, Advanced Materials 28, 560 (2016).

[111] C. Rinaldi, S. Varotto, M. Asa, J. Sławińska, J. Fujii, G. Vinai, S. Cecchi, D. Di Sante, R. Calarco, I. Vobornik, G. Panaccione, S. Picozzi, and R. Bertacco, *Ferroelectric Control of the Spin Texture in GeTe*, Nano Letters 18, 2751 (2018).

[112] J. Sławińska, D. Di Sante, S. Varotto, C. Rinaldi, R. Bertacco, and S. Picozzi, *Fe/GeTe(111) heterostructures as an avenue towards spintronics based on ferroelectric Rashba semiconductors*, Phys. Rev. B 99, 075306 (2019).

[113] E. Plekhanov, P. Barone, D. Di Sante, and S. Picozzi, *Engineering relativistic effects in ferroelectric SnTe*, Phys. Rev. B 90, 161108 (2014).

[114]H. Wang, P. Gopal, S. Picozzi, S. Curtarolo, M. Buongiorno Nardelli, and J. Sławińska, *Spin Hall effect in prototype Rashba ferroelectrics GeTe and SnTe*, npj Computational Materials 6, 7 (2020).

[115] H. Lee, J. Im, and H. Jin, *Harnessing the giant out-of-plane Rashba effect and the nanoscale persistent spin helix via ferroelectricity in SnTe thin films*, arxiv:1712.06112 (2018).

[116] L. L. Tao and E. Y. Tsymbal, *Persistent spin texture enforced by symmetry*, Nature Communications 9, 2763 (2018).

[117] M. A. U. Absor and F. Ishii, Intrinsic persistent spin helix state in two-dimensional group-IV monochalcogenide MX monolayers (M = Sn or Ge and X = S, Se, or Te), Phys. Rev. B 100, 115104 (2019).

[118] W. Wan, C. Liu, W. Xiao, and Y. Yao, *Promising ferroelectricity in 2D group IV tellurides: a first-principles study*, Applied Physics Letters 111, 132904 (2017).

[119] C. Yang, Y. Liu, G. Tang, X. Wang, and J. Hong, *Non-monotonic thickness dependence of Curie temperature and ferroelectricity in two-dimensional SnTe film*, Applied Physics Letters 113, 082905 (2018).

[120] Y. Wang, C. Xiao, M. Chen, C. Hua, J. Zou, C. Wu, J. Jiang, S. A. Yang, Y. Lu, and W. Ji, *Two-dimensional ferroelectricity and switchable spin-textures in ultra-thin elemental Te multilayers*, Materials Horizons 5, 521 (2018).

[121] T. P. Kaloni, K. Chang, B. J. Miller, Q.-K. Xue, X. Chen, S.-H. Ji, S. S. P. Parkin, and S. Barraza-Lopez, *From an atomic layer to the bulk: Low-temperature atomistic structure and ferroelectric and electronic properties of SnTe films*, Phys. Rev. B 99, 134108 (2019).

[122] K. Chang and S. S. P. Parkin, *The growth and phase distribution of ultrathin SnTe on graphene*, APL Materials 7, 041102 (2019).

[123] Z. Fu, M. Liu, and Z. Yang, *Substrate effects on the in-plane ferroelectric polarization of two-dimensional SnTe*, Phys. Rev. B 99, 205425 (2019).

[124] A. W. Cummings, J. H. Garcia, J. Fabian, and S. Roche, *Giant Spin Lifetime Anisotropy in Graphene Induced by Proximity Effects*, Phys. Rev. Lett. 119, 206601 (2017).

[125] J. H. Garcia, M. Vila, A. W. Cummings, and S. Roche, *Spin transport in graphene/transition metal dichalcogenide heterostructures*, Chem. Soc. Rev. 47, 3359 (2018).

[126] C. Autieri, P. Barone, J. Sławińska, and S. Picozzi, *Persistent spin helix in Rashba-Dresselhaus ferroelectric $CsBiNb_2O_7$*, Phys. Rev. Materials 3, 084416 (2019).

[127] H. Djani, A. C. Garcia-Castro, W.-Y. Tong, P. Barone, E. Bousquet, S. Picozzi, and P. Ghosez, *Rationalizing and engineering Rashba spin-splitting in ferroelectric oxides*, npj Quantum Materials 4, 51 (2019).

[128] R. P. Feynman, *Simulating physics with computers*, Int. J. Theor. Phys. 21, 467–488 (1982).

[129] P. W. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM Journal on Computing 5, 1484-1509 (1997).

[130] J. S. Bell, *On the Einstein Podolsky Rosen Paradox*, Phys. Pub. Co. 1, 195-200 (1964).

[131] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, W. K. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*, Phys. Rev. Lett 70, 1895 (1993).

[132] W. K. Wootters, and W. H. Zurek, *A single quantum cannot be cloned*, Nature 299, 802-803 (1982).

[133] F. Arute, K. Arya, R. Babbush et al., *Quantum supremacy using a programmable superconducting processor*. Nature 574, 505–510 (2019).

[134] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, *Hardware-efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets*, Nature 549, 242 (2017).

[135] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, *A variational eigenvalue solver on a quantum processor,* Nature Communications 5, 4213 (2014).

[136] B. Bauer, S. Bravyi, M. Motta, G. Chan, *Quantum Algorithms for Quantum Chemistry and Quantum Materials Science*, Chemical Reviews 22, 12685–12717 (2020).

[137] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition, 10th ed*. (Cambridge University Press, USA, 2011).

[138] D. S. Abrams and S. Lloyd, *Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors,* Physical Review Letters 83, 5162 (1999).

[139] L. K. Grover, *Quantum Mechanics Helps in Searching for a Needle in a Haystack*, Phys. Rev. Lett. 79, 325 (1997).

[140] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, *Quantum computational chemistry*, Rev. Mod. Phys. 92, 015003 (2020).

[141] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. John- son, M. KieferovÃą, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, *Quantum Chemistry in the Age of Quantum Computing*, Chemical Reviews 119, 10856 (2019).

[142] R. P. Feynman, *Quantum Mechanical Computers*, Optics News 11, 11 (1985).

[143] S. Lloyd, *Universal Quantum Simulators*, Science 273, 1073 (1996).

[144] K. Brown, W. Munro, and V. Kendon, *Using Quantum Computers for Quantum Simulation*, Entropy 12, 2268 (2010).

[145] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, *Could Quantum Computing of an Atomic Nucleus*, Phys. Rev. Lett. 120, 210501 (2018).

[146] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head- Gordon, *Simulated Quantum Computation of Molecular Energies*, Science 309, 1704 (2005).

[147] Y. Nam et al., *Ground-state energy estimation of the water molecule on a trapped-ion computer*, npj Quantum Information 6, 33 (2020).

[148] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, *Low-Depth Quantum Simulation of Materials*, Phys. Rev. X 8, 011044 (2018).

[149] B. Bauer, D. Wecker, A. J. Millis, M. B. Hastings, and M. Troyer, *Hybrid Quantum-Classical Approach to Correlated Materials*, Phys. Rev. X 6, 031045 (2016).

[150] J. M. Kreula, L. García-Álvarez, L. Lamata, S. R. Clark, E. Solano, and D. Jaksch, *Few-qubit quantum-classical simulation of strongly correlated lattice fermions*, EPJ Quantum Technology 3, 11 (2016).

[151] I. Rungger, N. Fitzpatrick, H. Chen, C. H. Alderete, H. Apel, A. Cowtan, A. Patterson, D. M. Ramo, Y. Zhu, N. H. Nguyen, E. Grant, S. Chretien, L. Wossnig, N. M. Linke, and R. Duncan, *Dynamical mean field theory algorithm and experiment on quantum computers*, arxiv:1910.04735 (2019).

[152] Y. Yao, F. Zhang, C.-Z. Wang, K.-M. Ho, and P. P. Orth, *Gutzwiller Hybrid Quantum-Classical Computing Approach for Correlated Materials*, arxiv:2003.04211 (2020).

[153] D. J. Chadi and M. L. Cohen, *Tight-Binding Calculations of the Valence Bands of Diamond and Zincblende Crystals*, Phys. Stat. Sol (b) 68, 405 (1974).

[154] A. F. Izmaylov, T.-C. Yen, R. A. Lang, and V. Vertelet- skyi, *Unitary Partitioning Approach to the Measurement Problem in the Variational Quantum Eigensolver Method*, Journal of Chemical Theory and Computation 16, 190 (2020).

[155] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, *A variational eigenvalue solver on a photonic quantum processor*, Nature Communications 5, 4213 (2014).

[156] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru- Guzik, New Journal of Physics 18, 023023 (2016). Guzik, *Quantum Chemistry in the Age of Quantum Computing*, Chemical Reviews 119, 10856 (2019).

[157] P. J. J. O'Malley et al., *Scalable Quantum Simulation of Molecular Energies*, Phys. Rev. X 6, 031007 (2016).

[158] O. Higgott, D. Wang, and S. Brierley, *Variational Quantum Computation of Excited States*, Quantum 3, 156 (2019).

[159] J. Lee, W. J. Huggins, M. Head-Gordon, and K. B. Whaley, *Generalized Unitary Coupled Cluster Wave functions for Quantum Computation*, Journal of Chemical Theory and Computation 15, 311 (2019).

[160] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. C. Ben- jamin, *Variational quantum algorithms for discovering Hamiltonian spectra*, Phys. Rev. A 99, 062304 (2019).

[161] I. G. Ryabinkin, T.-C. Yen, S. N. Genin, and A. F. Iz- maylov, *Qubit Coupled Cluster Method: A Systematic Approach to Quantum Chemistry on a Quantum Computer*, Journal of Chemical Theory and Computation 14, 6317 (2018).

[162] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, *Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz*, Quantum Science and Technology 4, 014008 (2018).

[163] J. A. Nelder and R. Mead, *A Simplex Method for Funcion Minimization*, Comput. J. 7, 308 (1965).

[164] S. Endo, S. C. Benjamin, and Y. Li, *Practical Quantum Error Mitigation for Near-Future Applications*, Phys. Rev. X 8, 031027 (2018).

[165] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, *Error mitigation extends the computational reach of a noisy computational processor*, Nature 567, 491 (2019).

[166] S. Zhang, Y. Lu, K. Zhang, W. Chen, Y. Li, J.-N. Zhang, and K. Kim, *Error-mitigated quantum gates exceeding physical fidelities in a trapped-ion system*, Nature Communications 11, 587 (2020).

[167] A. J. McCaskey, Z. P. Parks, J. Jakowski, S. V. Moore, T. D. Morris, T. S. Humble, and R. C. Pooser, *Quantum chemistry as a benchmark for near-term quantum computers*, npj Quantum Information 5, 99 (2019).

[168] G. Grosso and G. Parravicini, *Solid State Physics* (Elsevier Science, 2000).

[169] A. Aspuru-Guzik, *Simulated quantum computation of molecular energies*, Science 309, 1704–1707 (2005).

[170] M. Dob˘si˘cek, G. Johansson, V. Shumeiko, and G. Wendin, *Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark*, Physical Review A 76, 10.1103/physreva.76.030306 (2007).
[170] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik,

and J. L. O'Brien, *A variational eigenvalue solver on a photonic quantum processor, Nature Communications* 5, 4213 (2014).

[171] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru- Guzik, *The theory of variational hybrid quantum- classical algorithms*, New Journal of Physics 18, 023023 (2016).

[172] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, *Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets*, Nature 549, 242 (2017).

[173] O. Higgott, D. Wang, and S. Brierley, *Variational Quantum Computation of Excited States*, Quantum 3, 156 (2019).

[174] J. C. Slater and G. F. Koster, *Simplified lcao method for the periodic potential problem*, Phys. Rev. 94, 1498 (1954).

[175] C. Cade, L. Mineh, A. Montanaro, and S. Stanisic, *Strategies for solving the fermi-hubbard model on near-term quantum* computers (2019), arXiv:1912.06007 [quant-ph].

[176] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, Low-depth quantum simulation of materials, Physical Review X 8, 011044 (2018).

[177] B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, *Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm*, npj Quantum Information 6, 10 (2020).

[178] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, *Digital zero noise extrapolation for quantum error mitigation*, 2020 IEEE International Conference on Quantum Computing and Engineering (QCE) 10.1109/qce49297.2020.00045 (2020).

[179] N. Hatano and M. Suzuki, *Finding Exponential Product Formulas of Higher Orders*, 68, 37 (2005), arXiv:0506007 [math-ph].

[180] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, *Simulating hamiltonian dynamics with a truncated taylor series*, Physical Review Letters 114, 10.1103/PhysRevLett.114.090502 (2015), arXiv:1412.4687.

[181] A. Kalev and I. Hen, *Simulating Hamiltonian Dynamics with an Off-diagonal Series Expansion* (2020), arXiv:2006.02539.

[182] N. C. Rubin, R. Babbush, and J. McClean, *Application of fermionic marginal constraints to hybrid quantum algorithms*, New Journal of Physics 20, 053020 (2018).

[183] T.-C. Yen, V. Verteletskyi, and A. F. Izmaylov, *Measuring all compatible operators in one series of a single-qubit measurements using unitary transformations* (2020), arXiv:1907.09386 [quant-ph].

[184] H.-Y. Huang, R. Kueng, and J. Preskill, *Predicting many properties of a quantum system from very few measurements*, Nature Physics 16, 1050–1057 (2020).

[185] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, *Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz*, Quantum Science and Technology 4, 014008 (2018).

[186] I. G. Ryabinkin, T.-C. Yen, S. N. Genin, and A. F. Iz- maylov, *Qubit coupled cluster method: A systematic approach to quantum chemistry on a quantum computer*, Journal of Chemical Theory and Computation 14, 6317 (2018).

[187] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, *An adaptive variational algorithm for exact molecular simulations on a quantum computer*, Nature Communications 10, 3007 (2019).

[188] T.-C. Yen, R. A. Lang, and A. F. Izmaylov, *Exact and approximate symmetry projectors for the electronic structure problem on a quantum computer*, The Journal of Chemical Physics 151, 164111 (2019).

[189] J. T. Seeley, M. J. Richard, and P. J. Love, The bravyi-kitaev transformation for quantum computation of electronic structure, The Journal of Chemical Physics 137, 224109 (2012).

[190] W. Lavrijsen, A. Tudor, J. Muller, C. Iancu, and W. De Jong, *Classical Optimizers for Noisy Intermediate-Scale Quantum Devices,* Proceedings - IEEE International Conference on Quantum Computing and Engineering, QCE 2020 , 267 (2020), arXiv:2004.03004.

[191] B. I. Min, J. H. Shim, M. S. Park, K. Kim, S. K. Kwon, and S. J. Youn, *Origin of the stabilized simple- cubic structure in polonium: Spin-orbit interaction versus Peierls instability*, Physical Review B - Condensed Matter and Materials Physics 73, 3 (2006).

[192] A. Silva and J. Van Wezel, *The simple-cubic structure of elemental Polonium and its relation to combined charge and orbital order in other elemental chalcogens*, arXiv 028, 1 (2017).

[193] P. J. Karalekas, N. A. Tezak, E. C. Peterson, C. A. Ryan, M. P. da Silva, and R. S. Smith, *A quantum- classical cloud platform optimized for variational hybrid algorithms*, Quantum Science and Technology 5, 024003 (2020).

[194] I. D. Kivlichan, J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G. K.-L. Chan, and R. Babbush, *Quantum simulation of electronic structure with linear depth and connectivity*, Physical Review Letters 120, 110501 (2018).