ABSTRACT

*Ab Initio* Formation Energy Calculations for Defect Complexes in Diamond, ZnSe and CdS for Room-Temperature Quantum Computing

Ethan W. Dickey

Director: Enrique P. Blair, Ph.D.

Quantum computers are beginning to demonstrate a potential for practical uses in data security, protein folding, artificial intelligence and machine learning, and economics. Current obstacles to reliable large-scale quantum computers include better decoherence times, improved error correction schemes, and consistent fabrication. Creating a qubit (quantum bit) that can exist at room temperature makes large progress in each of these obstacles while decreasing operational costs (by eliminating the need for cryogenic cooling). Diamond has shown promising results when a defect known as the Nitrogen Vacancy (NV) complex is introduced via doping into the crystal. However, diamond is expensive to fabricate and foundries that can do so are rare. ZnSe and CdS, by contrast, can be grown at lower temperatures and pressures than diamond, and do not require the expensive retooling of foundries for the higher pressure and temperature required in diamond fabrication. This study provides a methodology and computational structure with which to identify semiconductors with similar desirable electronic properties as the NV defect in diamond and identifies potential defects for the two specified semiconductors of interest. This work may guide experimental exploration of quantum technologies based on semiconductor defects and could lead to lower cost, room-temperature qubits that are easily fabricated using the vast infrastructure of the current semiconductor industry.

APPROVED BY DIRECTOR OF HONORS THESIS:

Dr. Enrique P. Blair, Department of Electrical Engineering

APPROVED BY THE HONORS PROGRAM:

Dr. Andrew Wisely, Interim Director

DATE: 4/22/2021

# *Ab Initio* FORMATION ENERGY CALCULATIONS FOR DEFECT COMPLEXES IN DIAMOND, ZnSe, AND CdS FOR ROOM-TEMPERATURE QUANTUM COMPUTING

A Thesis Submitted to the Faculty of

Baylor University

In Partial Fulfillment of the Requirements for the

Honors Program

By

Ethan W. Dickey

Waco, Texas

May 2021

# 0 TABLE OF CONTENTS

# 0 List of Figures

# 0 LIST OF TABLES

# ACKNOWLEDGMENTS

"If I have seen further than others, it is by standing upon the shoulders of giants."

~Issac Newton

"Indeed, I count everything as loss because of the surpassing worth of knowing Christ Jesus my Lord. For his sake I have suffered the loss of all things and count them as rubbish, in order that I may gain Christ and be found in him."

~Philippians 3:8-9

This thesis would not be possible without the people who surround and support me. I am especially thankful for my thesis advisor, Dr. Enrique Blair, for all of your technical help and practical wisdom, and for being willing to take me on as an undergraduate researcher all those years ago. Thank you to Dr. Bill Poucher and Dr. Linda Olafsen for being a part of my defense committee and for being willing to take time out of the busiest part of the semester during an unprecedented pandemic to read, listen and respond to my thesis and help me finish my Baylor career strong. Thank you to my friends who surround me and support me, especially when times get tough. And lastly, thank you to my family who have endured my many missed holidays and breaks, endless studying, and constant busyness and who have continued to support and love me through all of it.

"It takes a village to raise a child, but it takes an entire city to raise a college student. I thank God that y'all are my city."

~Denton Wood

# CHAPTER ONE

## 1 Introduction

A quantum computer is a type of computer which can perform certain tasks much faster than a classical computer. These tasks are executed with quantum bits, or qubits, a type of computational bit based more fundamentally on the laws of physics (via superposition). With the power that qubits bring comes an inescapable complexity in their implementation.

### *1.1 A Brief History*

From ion-traps to quantum annealing to superconducting transmons [1, 2, 3], the vast majority of modern attempts at quantum computation involve cooling the system to sub-1 Kelvin temperatures in a way that results in high-maintenance, low-coherence-time qubits. This process is expensive to maintain and even more expensive to perform long enough to get any useful computational power out of it. Due to an ever-increasing interest in the power of quantum computation (most directly the power to break RSA (Rivest–Shamir–Adleman) encryption, the encryption standard that the world runs on), investors have been less concerned about how to make quantum computers run at room temperature and more focused on how to increase the number of qubits we can practically use, as soon as possible. No one wants to be last in establishing hacker-proof communications. However, this leaves practical attempts at room-temperature quantum computation trailing far behind.

## *1.2 Motivation*

Attempts at creating a practical room-temperature qubit have been focused around the Nitrogen Vacancy (NV) point defect in diamond, which has shown promising results because of diamond's wide band gap and ability to house a deep-center defect (further discussed in Section 2.3). Fabrication of diamond, however, is an expensive process that very few of the existing foundries can perform. Research on the horizon focuses on identifying compound semiconductors that exhibit the same optimal electronic characteristics that the NV defect center in diamond does and can be easily fabricated by existing industrial infrastructure.

With the power of modern computation, density functional theory (DFT, [4, 5]) has developed from just an idea to a powerful tool in the computation of the electronic structure of many-body systems. With this theory, researchers can imagine and explore physical implementations of a qubit even before it is realized in a laboratory. This computational study seeks to provide guidance for both future studies and experimental work in identifying point defects in compound semiconductors useful for implementation of a qubit that are also easily fabricated within existing industrial infrastructure.

# CHAPTER TWO

## 2 Background

First-principles calculations are a way of accurately predicting the electronic structure of a material without the expense and logistics of fabrication and testing. Through this method, predictions can be made about which materials will be useful for quantum computing. Making these predictions requires a wide comprehension of DFT and the methods surrounding it. This chapter covers many of those topics, including present-day room-temperature qubits, why ZnSe and CdS were selected as candidate material hosts, stable charged defects, formation energy, chemical potential, and periodic supercell corrections. For further readings, see Ref. [6, 7].

### *2.1 Present-Day Room-Temperature Quantum Computing*

Room temperature quantum computers are *required* for any sort of mass production and distribution of a quantum chip. Similar to a Graphics Processing Unit (GPU), the Quantum Processing Unit (QPU) will be an accessory to classical computation, performing certain tasks much faster than a classical computer can while leaving the majority of computation to them.

The nitrogen vacancy (NV) center in a diamond lattice (a wide-band-gap semiconductor) is a promising implementation of a qubit because it can be initialized, measured, and manipulated at room-temperature for use in quantum information processing. The NV center is a compound defect comprised of two adjacent point defects, a carbon vacancy ($V_C$) and a nitrogen substitution ($N_C$), as seen in the left panel of Fig. 1. These defects can create a potential well and allow us to effectively trap an electron in these sites.

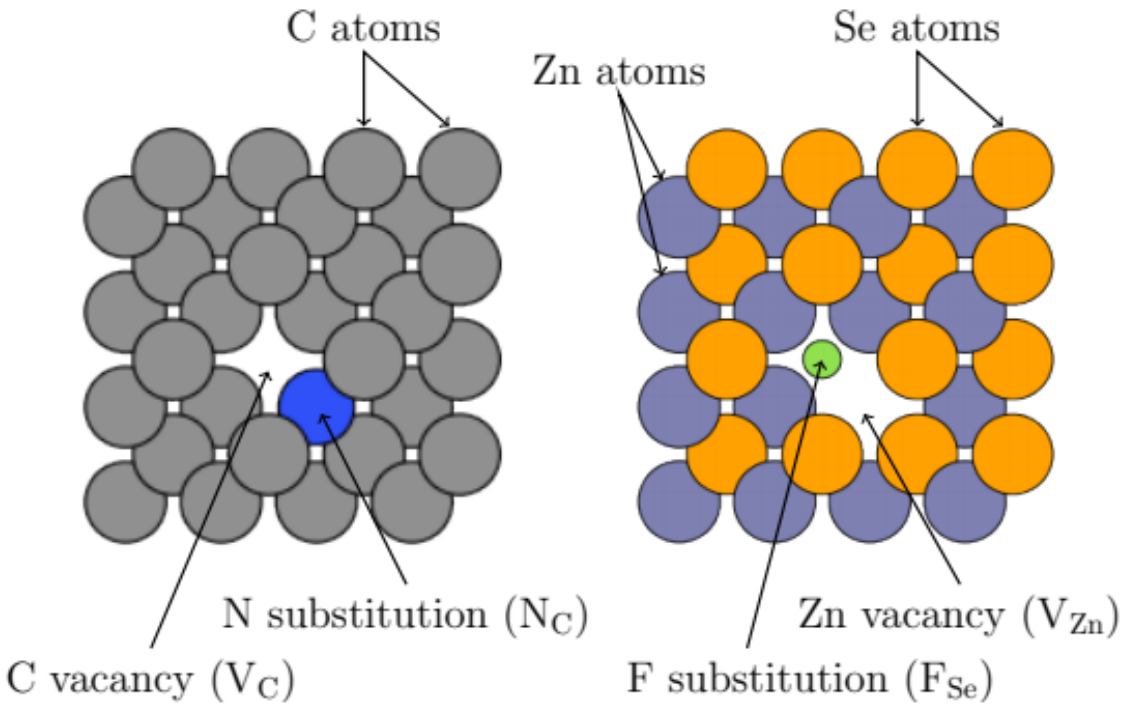


**Figure 1:** (Left) A C vacancy and a C-to-N substitution form an NV center in diamond. (Right) A Zn vacancy and a Se-to-F substitution form a $F_{Se}V_{Zn}$ defect complex in ZnSe.

The NV center in diamond is of particular interest because of the defect's robustness. The NV center can be initialized, manipulated, and measured with high fidelity at room temperature [6]. For a stable qubit, diamond hosts this defect complex. However, the two individual defects must also be studied as they make up the NV center.

Unfortunately, diamond synthesis is expensive because it requires temperatures and pressures beyond that of a normal semiconductor foundry. Other wide-band-gap semiconductors can be much more cost-effective to grow than diamond, and many can host defects with similar properties to the NV complex in diamond.

### *2.2 ZnSe and CdS*

Two particular wide-band-gap semiconductors of interest in this study are ZnSe and CdS. They were chosen because their wide band gap admits deep-level defects, allowing for optical interfaces to the defect states without causing interfering electronic states in the host [6]. Additionally, ZnSe may be isotopically purified to provide a spin-free host matrix, which minimizes spin-dephasing of defect-based qubits [8]. ZnSe has received less attention as a host matrix for NV-like defects than have diamond and SiC [9, 10]. While F and Cl defects have been studied in ZnSe [8], it is presently unknown what other charged defects may be useful for implementing qubits in ZnSe, nor under what conditions they are stable. While CdS has been studied for use in quantum dots [11], its usefulness as a wide-band-gap semiconductor host matrix for a qubit is presently unknown. However, CdS's electronic structure is very similar to that of ZnSe, so it was chosen for further study as well.

### *2.3 Stable Charged Defects*

While making point defects is a well developed and understood process [12], finding useful (stable) charged defects is a challenging process. Charged defects are point defects with a nonzero charge at the defect site. As mentioned in Section 2.1, properties of a point defect that make it useful to quantum computing include the ability to initialize, manipulate, and measure the defect with high fidelity at room temperature. Due to these qualifications, most point defects must be charged. In addition, there are two distinguishing characteristics of the NV center (specifically, the -1 charge state, $NV^{-1}$) in diamond that make it suitable for quantum computing. Firstly, since the $NV^{-1}$ center's bound states are highly lo-

calized (due to the wide band gap of diamond), they remain very isolated from potential sources of decoherence [6]. Secondly, the way the defect behaves in its excited state allows for high fidelity optical initialization and measurement. The presence of both of those properties at room temperature is what distinguishes it from other forms of qubits. Most present-day attempts that use solid state systems either require extremely cold temperatures for thermal equilibrium or can only be measured in an ensemble [6]. For further discussion on what characteristics a host material and a candidate defect center should exhibit to reproduce those two distinguishing characteristics, see Ref. [6].

It is possible to predict the overall charge states, $q$, of a semiconductor point defect which allow it to be useful for quantum computation the way the $NV^{-1}$ center in diamond is [6]. To be useful, the charged defect should have spin-conserving transitions that allow RF or optical interfacing to its electronic state.

Specifically, we define a compound semiconductor with constituent elements A and B as AB. Let AB have an A vacancy ($V_A$) adjacent to a B $\rightarrow$ X substitution ($X_B$). The defect defined as

$$X_B V_A^q \tag{1}$$

is said to have appropriate spin-conserving transitions if $n_e$, the number of electrons participating in its electronic state, is [6]

$$n_e \in \{4, 6\}. \tag{2}$$

Additionally, it can be shown that for all $q$ in Equation (1),

$$q = N_X - n_e, \tag{3}$$

where $N_X$ is the number of valence electrons for element X. For example, in ZnSe,

| Element Group ($N_X$) | $n_e = 4$ | $n_e = 6$ |
|---|---|---|
| 1 | -3 | -5 |
| 2 | -2 | -4 |
| 3 | -1 | -3 |
| 4 | 0 | -2 |
| 5 | +1 | -1 |
| 6 | +2 | 0 |
| 7 | +3 | +1 |

**Table 1:** The useful charge states $q$ for charged defects of the form $X_BV^q{}_A$. Here, $q$ is calculated for impurity X in various element groups.

a $F_{Se}V^q{}_{Zn}$ defect (right panel of Fig. 1) should have $q$ = 1 or $q$ = 3 to be stable, since $N_F$ = 7. This method can be quickly expanded to calculate the useful charge states $q$ for defect element X in any other groups on the periodic table. These predictions are listed in Table 1.

### *2.4 Defect Formation Energy*

After stable charge states have been identified, the doping conditions under which the charged defect $X_BV^q{}_A$ is stable must be identified. Doping is intentionally introducing defects (impurities) into a material and is key to controlling the electronic properties of bulk semiconductors [13]. Desired doping conditions may be determined from a plot of the formation energy of the system as a function of Fermi level for various defects in different charge states. These calculations yield insight into which defects are the most energetically favorable at certain Fermi levels, where Fermi level is the work required to add one electron to a solid-state body. The Fermi level in relation to the bands of a body is critical in determining electrical structure of that body. From the formation energy plot we can determine which regions the desired charged defects are stable in and from those regions, determine the doping conditions to place the Fermi level in that region.

The formation energy of the defect, $E^f[X_BV_A^q]$, may be calculated using first-

principles. This process can be executed using the supercell method [7] (as further discussed in Section 3.1.4) using density functional theory (DFT) software. The formation energy can be calculated as follows:

$$E^f[\mathrm{X}_B\mathrm{V}_A^q] = E_{bulk} - E[\mathrm{X}_B\mathrm{V}_A^q] - \sum_z n_Z\mu_Z + q(\varepsilon_\mathrm{F} + \varepsilon_\mathrm{VBM} + \Delta V), \tag{4}$$

where $E_{bulk}$ and $E[\mathrm{X}_B\mathrm{V}_A^q]$ represent the total energy of the pristine and defected supercells, respectively; $n_Z$ is the number of element Z atoms removed from the lattice to form the defect complex; $\mu_Z$ is the chemical potential of element Z; $\varepsilon_\mathrm{F}$ is the Fermi energy relative to $\varepsilon_\mathrm{VBM}$ in the range $\varepsilon_\mathrm{F} \in [0, E_g]$, where $E_g$ is the semiconductor's band gap, measured from the valance band maximum to the conduction band minimum; $\varepsilon_\mathrm{VBM}$ is the valence band maximum; and $\Delta V$ is a term that includes both the electrostatic correction term (further discussed in Section 2.6) and a term that aligns the Fermi levels of the pristine and defected supercells. Note that this equation becomes more complicated with compound semiconductors, a topic further elaborated on in Section 2.5.

Once the formation energy for each charge state of each defect at a range of Fermi levels has been calculated, the useful charge states can be extracted. Formally, defect $\alpha$ is more stable than defect $\beta$ if

$$E^f[\alpha] < E^f[\beta]. \tag{5}$$

Additionally, the slope of the plot $\varepsilon_\mathrm{F}$ vs $E^f[\alpha]$ is equal to $q$. By plotting $E^f[\alpha]$ for different $q$ values over the range $0 < \varepsilon_\mathrm{F} < E_g$, where charged defects are the most stable can be determined. If the lowest $E^f[\alpha]$ at a given $\varepsilon_\mathrm{F}$ is chosen as $E^f_\mathrm{min}[\alpha]$ and plotted vs $\varepsilon_\mathrm{F}$, the slope of $E^f_\mathrm{min}[\alpha]$ at $\varepsilon_\mathrm{F}$ indicates the most stable $q$ at that point. Thus, $\varepsilon_\mathrm{F}$, which can be controlled by the dopant concentration [14], provides a sort of "tuning knob" for selecting the defect's stable charge state $q$.

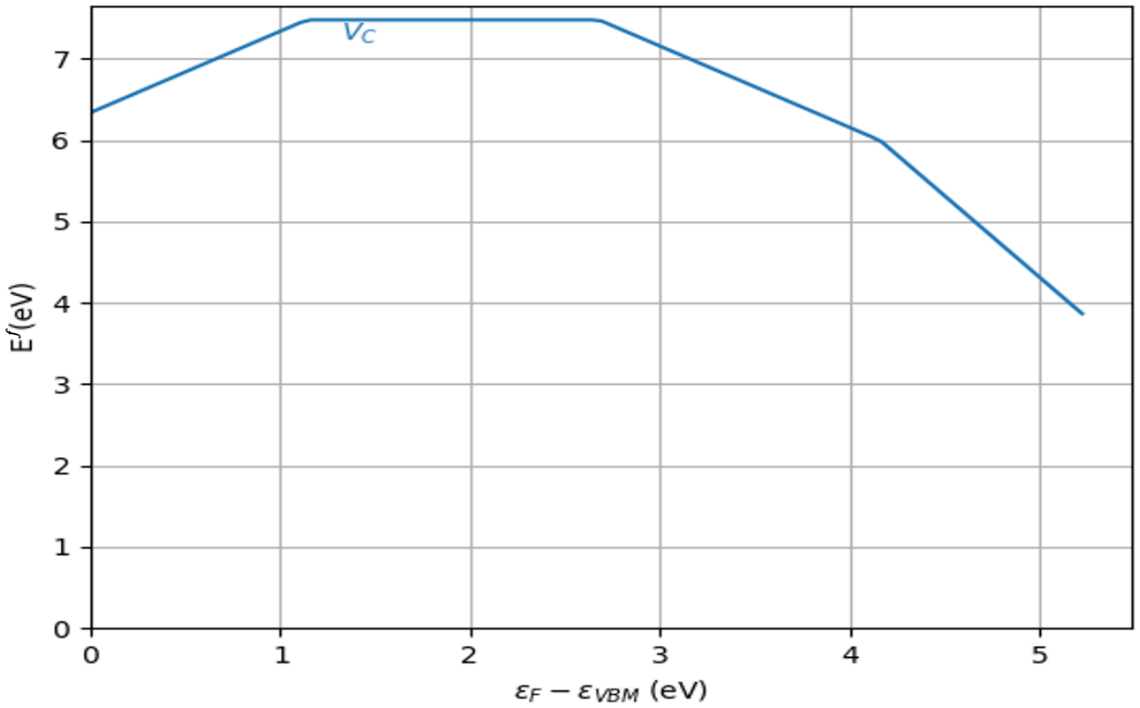


**Figure 2:** Defect formation energy in diamond as a function of the Fermi level, $\varepsilon_{\mathrm{F}}$, indicates that a $\mathrm{V_C}$ center in the $q = -1$ charge state is most stable for a crystal doped to around $2.6 < \varepsilon_{\mathrm{F}} < 4.2$ eV.

For example, Fig. 2 shows $E^f_{\mathrm{min}}$ for the carbon monovacancy ($\mathrm{V_C}$) in diamond. The slope of each of the components of this graph represent the charged state that is most stable in that region. For example, from relative Fermi level of 0 to around 1.2, the $q = +1$ charge state is most stable. This graph indicates that the $q = -1$ state can be selected by doping the Fermi level to around $2.6 < \varepsilon_{\mathrm{F}} < 4.2$ eV relative to the valance band maximum.

### *2.5 Chemical Potential*

The chemical potential of elementary semiconductors and molecules are relatively easy to calculate. For these, the chemical potential is the total energy of the elementary compound divided by the total number of atoms in it.

With compound semiconductors and molecules, it becomes more complicated. Essentially, we want to put bounds on what the chemical potential could be and try

to make those bounds as close as possible. For upper bounds, the process is simple. Let's use ZnSe as an example. For the chemical potential of Zn ($\mu_{Zn}$), we know that it must be less than or equal to the naturally occurring chemical potential of Zn, $\mu_{Zn[bulk]}$,

$$\mu_{Zn} \leq \mu_{Zn[bulk]}. \tag{6}$$

The same is true for Se,

$$\mu_{Se} \leq \mu_{Se[bulk]}. \tag{7}$$

Otherwise, bulk Zn and Se would be more stable than the ZnSe crystal and precipitation would occur, forming a bulk Zn or Se phase [8]. In addition, for a single unit cell of ZnSe, the sum of the chemical potentials of Zn and Se is equal to the ground state total energy of that system,

$$\mu_{Zn} + \mu_{Se} = E_{ZnSe}. \tag{8}$$

Finally, for a ZnSe crystal to be stable, we require that the energy of the system must be less than the energy of bulk Zn and bulk Se by themselves:

$$E_{ZnSe} < \mu_{Zn[bulk]} + \mu_{Se[bulk]}. \tag{9}$$

We express this difference using a term called the enthalpy of formation, $\Delta H_f[ZnSe]$, which is a measure of the stability of the system:

$$E_{ZnSe} = \mu_{Zn[bulk]} + \mu_{Se[bulk]} + \Delta H_f[ZnSe], \tag{10}$$

where $\Delta H_f[ZnSe]$ must be negative for the system to be stable.

Using equations (8) and (10), the variation in $\mu_{Zn}$ and $\mu_{Se}$ is often parameterized by the scalar $\lambda$ which varies between 0 and 1 [8], as follows:

$$\mu_{Zn} = \mu_{Zn[bulk]} + \lambda\Delta H \tag{11}$$

$$\mu_{Se} = \mu_{Se[bulk]} + (1-\lambda)\Delta H, \tag{12}$$

with $0 \leq \lambda \leq 1$, where

$$\begin{aligned} \lambda \to 0 &\implies \text{Zn-rich conditions, and} \\ \lambda \to 1 &\implies \text{Se-rich conditions.} \end{aligned} \tag{13}$$

Native (pristine) ZnSe has $\lambda = 0.5$.

The chemical potential of a pristine host material can be calculated by multiplying the total number of each atom by its chemical potential:

$$\mu_{ZnSe} = \sum_i n_i\mu_i = n_{Zn}\mu_{Zn} + n_{Se}\mu_{Se}. \tag{14}$$

The chemical potential of a defected host material (with impurity X) is calculated the same way:

$$\mu_{ZnSe/X} = \sum_i n_i\mu_i = n_{Zn}\mu_{Zn} + n_{Se}\mu_{Se} + n_{imp}\mu_{imp}. \tag{15}$$

Briefly back to formation energy, Santos *et al*. [8] simplifies the formation energy equation to

$$\Omega = E_{def} - E_{bulk} - \sum_i n_i\mu_i \tag{16}$$

for formation energy $\Omega$ of a defected compound semiconductor, where $n_i$ is the number of atoms of element i that are added or removed from the pristine crystal

to create the defect with its respective chemical potential $\mu_i$ [8]. For defected ZnSe (with one type of defect),

$$\Omega = E_{def} - E_{bulk} - n_{Zn}\mu_{Zn} - n_{Se}\mu_{Se} - n_{imp}\mu_{imp}. \tag{17}$$

Following the discussion from Santos *et al*. [8], substituting equations (11) and (12) back into the formation energy Equation (17) yields

$$\begin{aligned}\Omega(\lambda, \mu_{imp}, E_F) =& E_{def} - E_{bulk} \\ &- n_{Zn}(\mu_{Zn[bulk]} + \lambda\Delta H) \\ &- n_{Se}(\mu_{Se[bulk]} + (1-\lambda)\Delta H) \\ &- n_{imp}\mu_{imp} \\ &+ q(\varepsilon_{\mathrm{F}} + \varepsilon_{\mathrm{VBM}} + \Delta V),\end{aligned} \tag{18}$$

where the last term is added to account for charged defect-defect interactions and the electrostatic correction term.

### *2.6 Periodic Supercell Corrections*

For all of the big calculations required in Equation (18), a single unit cell will not suffice. Fig. 3 shows that the unit cell is a simple configuration that can be used to represent a lattice with a defect on the molecular level. More specifically, a unit cell is the smallest system of atoms that can be used to represent the entire lattice by repeating it periodically. Interactions between the atoms of the unit cell and its repeated images throughout the extended, infinite crystal arise because of periodic boundary conditions inherent in plane-wave basis DFT calculations.

Theoretically, solving the set of Schrödinger equations for a lattice requires calculation of an infinite number of wavefunctions for an infinite number of elec-

trons that extend throughout the entire solid. However, since particles in a perfect crystal lattice are arranged regularly, Bloch's Theorem can be used to express the wavefunctions of the solid as wavefunctions in reciprocal space (also known as momentum space or k-space). Through this transformation, the infinite number of wavefunctions for an infinite number of electrons gets reduced to the number of electrons in the unit cell [15].

When running the formation energy calculations, it becomes necessary to distance the defect in the unit cell from its images in the repeated lattice. This minimizes cell-image interactions, reflecting the likely physical reality of having defects in a diffuse limit which does not affect the Fermi level. If the image unit cell is used, defect-defect interaction is not only possible, but probable. The supercell solves this problem by simply increasing the amount of surrounding lattice that is given with the defect, as can be seen in Fig. 4. In practice, the supercell is made by repeating the pristine image unit cell N$\times$N$\times$N times and then introducing a single defect into the entire structure. This process can be carried to very high N, when required (Fig. 5).

In this periodic supercell method, defect-defect interaction avoidance has to be balanced with runtime. With higher N (in NxNxN repetition) comes quadratically higher runtimes, as an increase from 2x2x2 to 3x3x3 increases the number of atoms to consider by 19 ($8 \rightarrow 27$) times the number of electrons per atom. However, larger supercells also more naturally support ionic relaxation, which puts the system closer to its most relaxed state. Supercell corrections are used to compensate for image interactions in smaller supercells so that shorter runtimes can be achieved. This post-processing step is discussed further in Section 3.1.4.

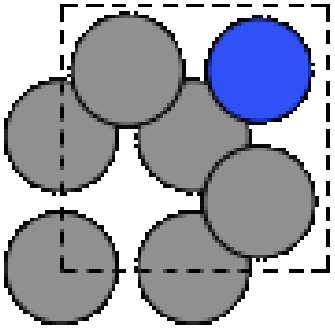

**Figure 3:** Diamond unit cell with a NV defect complex. The N substitution ($N_C$) is highlighted in blue and is adjacent to the C vacancy ($V_C$).

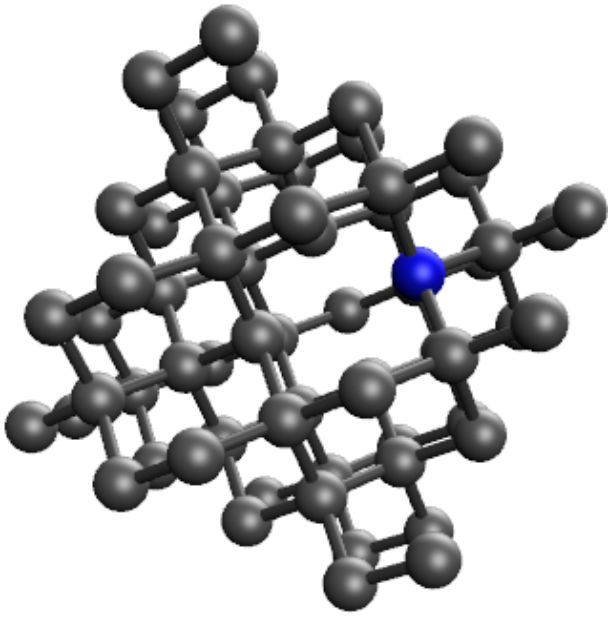

**Figure 4:** Diamond supercell with a NV defect complex. The N substitution ($N_C$) is highlighted in blue and is adjacent to the C vacancy ($V_C$).

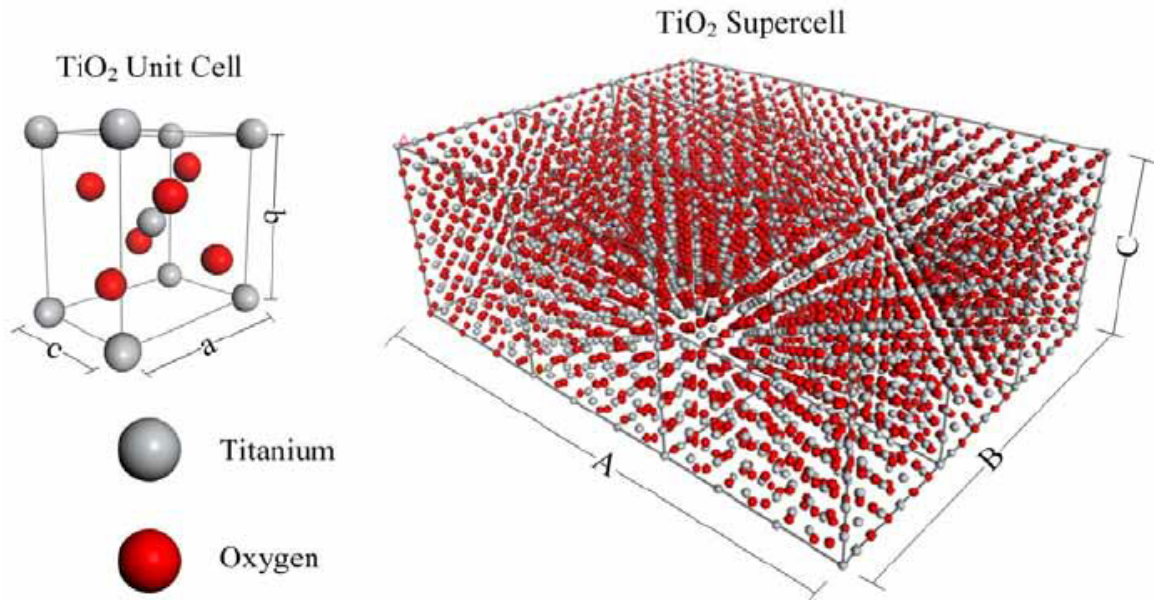


**Figure 5:** Titanium dioxide unit cell and supercell configurations used for molecular simulation. Retrieved from [16].

# CHAPTER THREE

## 3 Methodologies

This chapter explains the various methodologies that are used to gather and analyze the data relevant to this research. The methodologies will include the overall process and work flow, the VASP computing package, and convergence and optimization of system parameters.

### *3.1 Process*

The process through which formation energy plots are created is long and complicated. The following subsections will briefly discuss each important step along the way and conclude with a summarizing flowchart.

#### *3.1.1 Convergence*

To start formation energy calculations, first we need to calibrate known constants to our system. This serves two purposes: firstly, it allows us to verify that we are running the correct simulation and parameters. Secondly, it provides a fine tuning to what is set up in order to get better end results. If constants from outside the setup are used, when mixed with calculated values from inside the system, then oftentimes the resultant values (the simulation results that actually matter) are skewed (as there are two different data biases/perspectives going into it). This process is discussed more in Section 3.3.

#### *3.1.2 Lattice Structure*

The way in which a semiconductor naturally forms is important to the atomic structure fed into the calculations of that semiconductor's electronic structure.

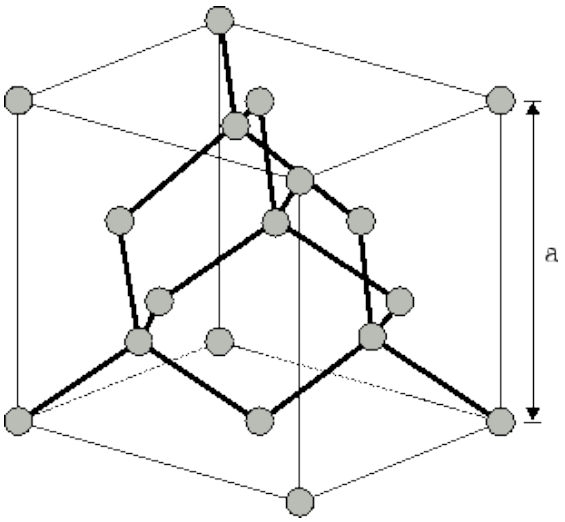


**Figure 6:** The arrangement of atoms in the diamond lattice structure. Retrieved from [17].

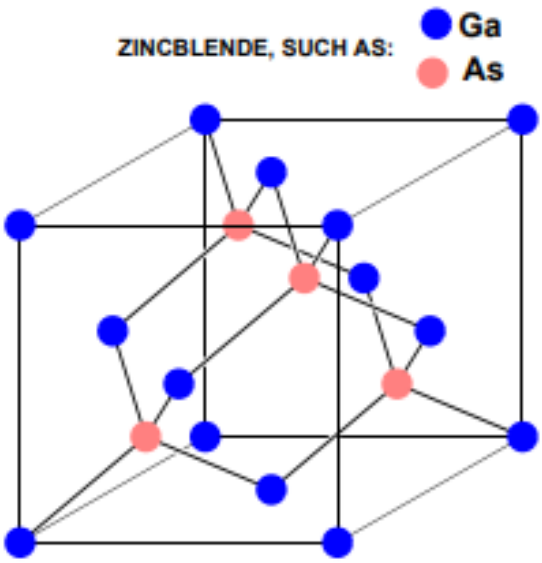


**Figure 7:** The arrangement of atoms in the zincblende lattice structure. Retrieved from [18].

Three elementary semiconductors (C, Si and Ge) form in the `diamond` lattice structure (as illustrated in Fig. 6). When running calculations on diamond, it is important to use the diamond lattice structure (clearly).

Additionally, many interesting compound semiconductors have the structure `zincblende`, which is very similar to the diamond lattice structure except it incorporates a second element, as illustrated in Fig. 7.

The reason this section must exist is because CdS does not conform to the typical structure of interesting semiconductors. Natural CdS exists in two forms: `hawleyite` and `greenockite` (Fig. 8, Fig. 9). When forming CdS using chemical precipitation methods, hawleyite (cubic) CdS is formed. This matches the pattern established above as hawleyite is a cubic zincblende form, as can be seen in Fig. 8.

### *3.1.3 Unit Cell Calculations*

Next, the simulation runs Self-Consistent Field (SCF) calculations on a unit cell to set up the more complicated simulations. These simple SCF calculations lay the groundwork for the band gap calculation.

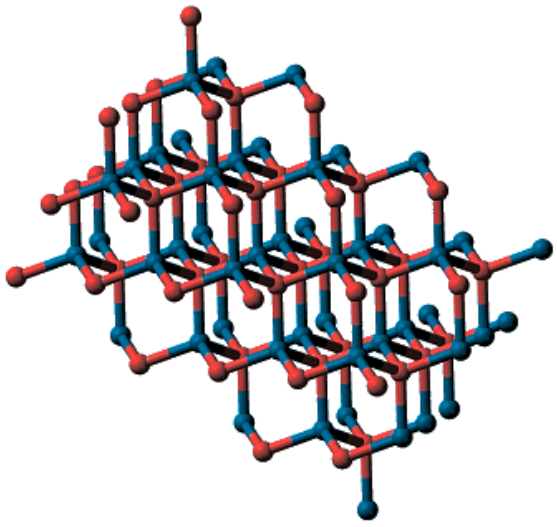

**Figure 8:** The arrangement atoms in the hawleyite (cubic) lattice structure. Retrieved from [19].

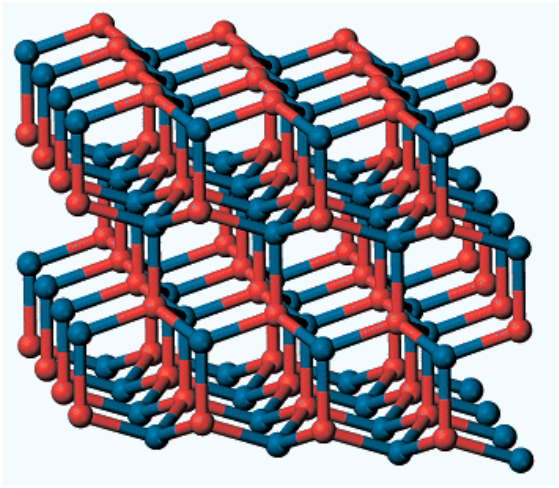

**Figure 9:** The arrangement atoms in the greenockite (hexagonal) lattice structure. Retrieved from [19].

The band gap is measured from the valance band maximum to the conduction band minimum (as seen in Fig. 10) and represents a range of energies that the host material's electrons cannot occupy. If researchers can make a defect whose energy state exists in this band gap, it would provide optimal conditions for a qubit. Calculating the band gap is one of the most important steps in this process as it restricts the formation energy plot, giving the exact bounds on what Fermi levels are possible (as going above or below this level would put the defect outside the band gap). However, it is well-known that DFT underestimates the band gap [20, 21]. There are two solutions to this problem: either correct the calculation by applying the GW Correction [22] to the Non-Self Consistent Field (NSCF) calculations or opt for using Hybrid Density Functionals instead of NSCF calculations. Hybrid density functionals add some fraction of the Hartree-Fock exchange in order to circumvent the expensive calculation of the GW Correction, leading to fast and accurate descriptions of the electronic structure and a better depiction of the band structure [23, 24].

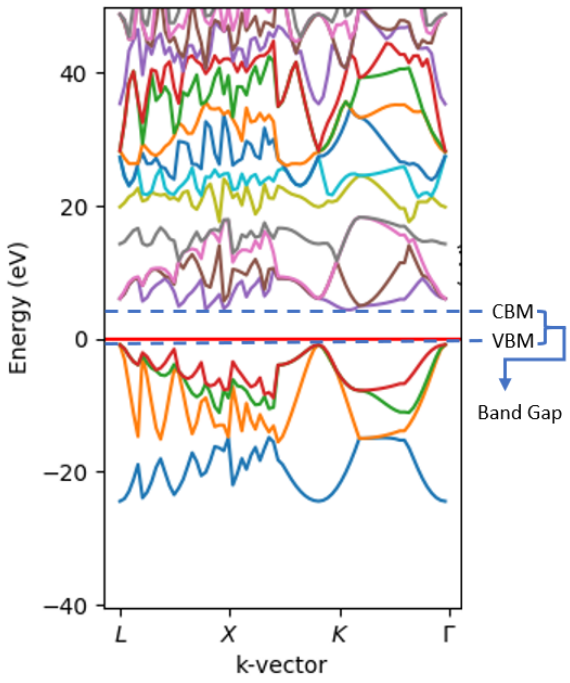


**Figure 10:** The band structure of diamond in k-space with Valance Band Maximum and Conduction Band Minimum labeled. Calculated using methods described in Section 4.1.1.

### *3.1.4 Supercell Calculations*

After obtaining the band gap from unit cell SCF and correction steps, the supercell calculations begin. Unit cell calculations are good enough for the band gap calculation because the lattice is pristine, so cell-cell interactions are kept to what is expected. However, when adding defects to a system, it is important to minimize the defect-defect interactions (caused by two defects being close enough to each other that they affect the other's relaxation calculations). This is solved by introducing supercells, which are N×N×N copies of the host material's unit cell, typically with a single defect. For these calculations, N ∈ [1, 2]; any larger values for N cause the calculations to take too long to run (weeks). Additionally, accurate calculations can be obtained with little relative error at those N values. A pristine supercell (one with no defects) is used to calculate the chemical potential of the bulk host material as discussed in Section 2.5.

From the pristine supercell, each point defect can be formed and the total energy of the system for each charge state is calculated (using the typical SCF method). Once each of these calculations has run, an electrostatic correction program called *sxdefectalign* [25] is run in order to correct for nonzero charge state defect-defect interactions. As the final step in the supercell calculations, the energy difference between the pristine supercell and each charge state for each defect is calculated (with the correction term added in). These energy differences are utilized when calculating the formation energy and are the primary output of this step.

### *3.1.5 Chemical Potential Reference Values*

Following the discussion from Section 2.5, the chemical potential of a defected host material is obtained from the sum of the chemical potentials for each of the component atoms (Equation (15)). The chemical potential of an individual material can be computed by taking the most naturally occurring form of the element, calculating its total energy, and dividing that by the total number of atoms in the material. This can be done with a simple pristine unit cell calculation. For example, in a C lattice (diamond) with a N vacancy (the $NV^{-1}$ defect), the chemical potential of N is calculated by taking its most naturally occurring form, $N_2$, and calculating its total energy. Dividing that value by the total number of N atoms in the lattice gives the chemical potential of N.

### *3.1.6 Post-Processing and Visualization*

The post processing begins with calculation of the formation energy for each charge state as defined in Equation (4). This step provides a set of formation energy lines as seen in Fig. 11. From these lines, the minimum of each formation energy at any given Fermi level is taken in order to form the final formation energy plot as seen in Fig. 12. On the x-axis is the Fermi level ($\varepsilon_F$) with respect to the valance

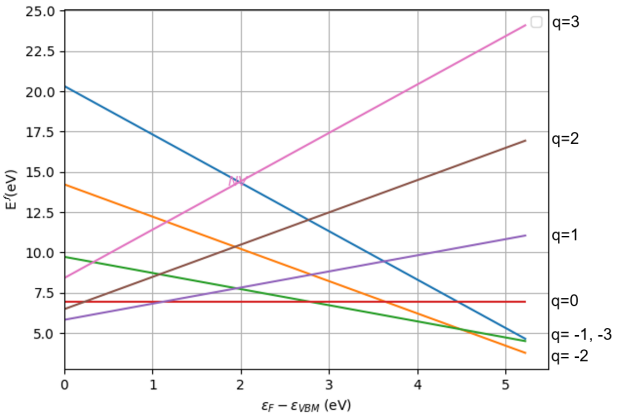


**Figure 11:** Formation energy as a function of Fermi level for each charge state for the NV defect in diamond. Each line represents a different charged state.

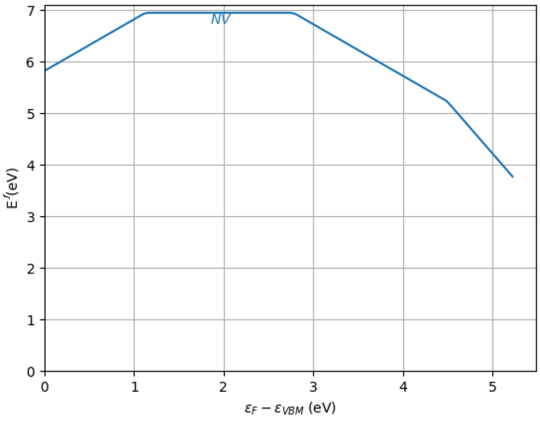


**Figure 12:** Formation energy plot for the NV defect in diamond. This represents the minimum trace of Fig. 11.

band maximum (VBM) of bulk diamond. On the y-axis is the formation energy of the system (as a result of taking the minimum formation energy of each charged defect at that point). These plots are further discussed in section 4.1.

### *3.1.7 Flowchart*

Each of the previous steps and what useful data comes out of each one is illustrated in Fig. 13.

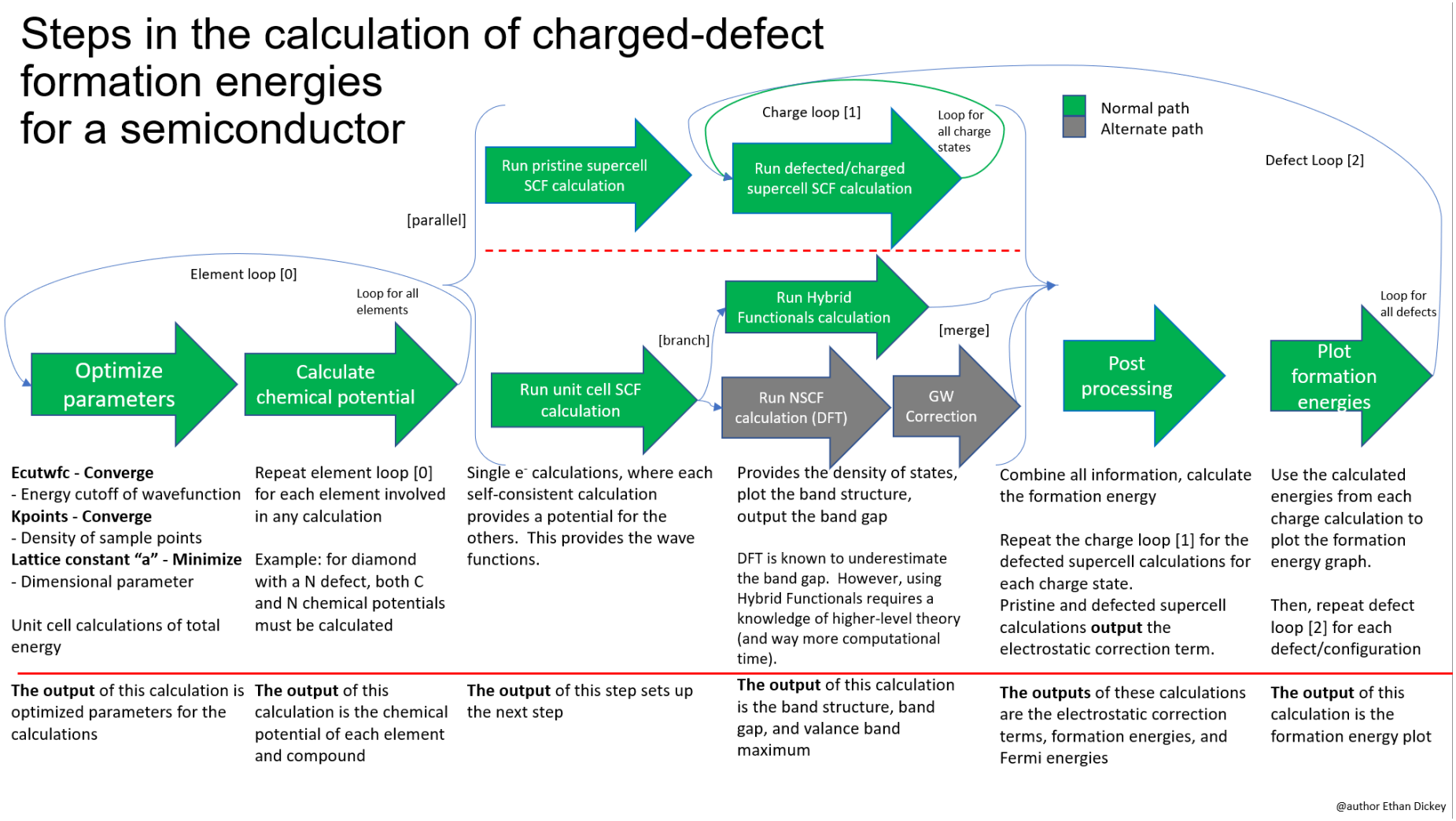


**Figure 13:** Steps in the calculation and plotting of formation energy.

### *3.2 VASP*

VASP (Vienna Ab initio Simulation Package) is a software package designed to model materials at the atomic scale using *ab initio* (first principles) calculations [26]. This software package is useful for calculating the electronic structure of a system using different methods (SCF, NSCF, Hybrid Functionals, etc.); in turn, total energy can be used to optimize and converge certain simulation parameters (lattice constant "a," sampling of the points in k-space "kpoints," plane wave energy cutoff, etc.).

The input files required to run a calculation with VASP are large and complicated, so John Kitchen's open-source VASP wrapper was adapted for our purposes [27, 28]. This software provides a Python interface for defining crystal structures, generating VASP input files for complex workflows, and for post-processing data.

An alternative software called PyCDT was considered for use originally, as it claims to "expedite the setup and post-processing of defect calculations with [VASP]" (Broberg *et al*. [29]). Diving into the software package, it did help with the setup successfully (generation of input files), but did not perform the band gap correction that was needed. It does support the NSCF calculations (see Section 3.1.3) which do a partial correction term calculation, but neither the full GW correction nor Hybrid Functionals calculation were found. Additionally, the software package was recently updated to a new version of Python and was not thoroughly tested. Several requests were made to the developers to fix said bugs, but there are no active contributors to the project so these errors were slow to be fixed. For all of these reasons, a simpler wrapper was chosen (John Kitchen's) and a shell outside of that was developed which incorporates all of the components of a proper formation energy calculation.

Modifications to the VASP Wrapper pertain to both input generation and the

output processing. Input generation functionality added includes pristine and defected supercell creation, supercell ionic relaxation, and number of valance electrons in a system calculation. Output processing tools mainly work with retrieving the band structure as well as calculating and plotting the formation energies.

### *3.3 Convergence and Optimizing System Parameters*

Convergence is the process of systematically guaranteeing that enough computational precision is used to obtain consistent results. A system will often not converge if the initial guesses of several parameters are too far off. Taking these system parameters, we can make our initial guess better by individually converging the total energy of the system with respect to each parameter.

These calculations are necessary to perform on each new system separately because every research group uses difference processing software to perform calculations. Well-known values can be used as a starting point, but need to be adapted to fit the current setup. Comparing certain resultant values to other research groups' values also allows validation of the system setup.

Converging and optimizing system parameters converges the total energy to the meV (less than 1 meV of change from one value to the next). At least three values are converged/optimized for every semiconductor that is tested:

1. The **plane wave energy cutoff (`"ecut"`)** is an upper bound on the plane-wave energies used to model the physical system. The electronic wavefunctions for these bound states are theoretically infinite due to the Fourier series that represents the plane wave basis set, but plane waves with lower kinetic energy typically have a much higher impact on the total energy [15]. Practically, we can see convergence of the total energy without going over 1000 eV. Formally, the plane wave energy cutoff is converged with respect to the

maximum total energy $TE$ such that

$$\text{for each } \Delta E_{cut} \geq 1 \text{ Ry}, \Delta TE < 1 \text{ meV}. \tag{19}$$

2. The **kpoints value $\mathtt{K_p}$** is converged for a similar issue. In momentum space (k-space), the original periodic lattice is transformed into the reciprocal lattice, an equivalent form that allows for a more careful study of the structure of the lattice. However, the wavefunctions much be calculated at each point in the region of k-space that the problem exists in (a continuum of points) [15]. Therefore, we use a value "kpoints" to determine the density of sampling that occurs in k-space.

   Formally, similarly to the plane wave energy cutoff, kpoints $K_p$ is converged with respect to the maximum total energy $TE$ such that

$$\Delta TE < 1 \text{ meV}, K_p \in \mathbb{N}^*. \tag{20}$$

3. The **lattice constant a** defines the spacing between atoms in the given lattice. The DFT prediction for $\mathtt{a}$ may be found by optimizing the total system energy a as a function of the lattice constant (as the system will most naturally rest in the least energetic state). Formally, the lattice constant is optimized with respect to the total energy such that at a point $x_m$,

$$\begin{gathered} TE(x_i) > TE(x_m) < TE(x_j) \\ \forall x_i, x_m, x_j \in \mathbb{R} \mid x_i < x_m < x_j. \end{gathered} \tag{21}$$

This is optimized to the meV, such that

$$\begin{aligned} TE(x_m) - TE(x_i) < 1 \text{ meV and} \\ TE(x_j) - TE(x_m) < 1 \text{ meV}. \end{aligned} \tag{22}$$

In addition to each semiconductor that is tested, each time a pristine (non-defected) calculation is run on a material, those parameters must be converged or optimized. Some examples include in the chemical potential calculation of N, Zn or S, where each element must be individually calibrated before their chemical potentials can be calculated.

Some systems require additional parameters to be converged. The primary example of this nitrogen. When calculating its chemical potential, it is necessary to evaluate nitrogen's total energy in its most relaxed state, which is the gaseous form $N_2$. Making $N_2$ requires specification of the **air gap** constant. This value helps specify how much space is between each $N_2$ center. In a lattice, it is not necessary to specify this because the periodicity of a lattice and the fact that it is a solid require the edges of each cell to be connected and form the lattice. In $N_2$, because it is gaseous, it is necessary to space out the $N_2$ centers to their most relaxed state (converged per the 1 meV standard).

# CHAPTER FOUR

## 4 Results and Analysis

This chapter shows results for the various calculations performed during the course of this research. It also provides a methodology and computational structure for future work, along with recommendations and explanations on where exactly to go. The calculations covered include diamond band structure and formation energy as well as ZnSe and CdS band structure. It then provides recommendations for the formation energy calculations on ZnSe and CdS along with a structure to perform these calculations.

### *4.1 Diamond Results*

#### *4.1.1 Band Structure*

The band structure of diamond was identified with DFT using a SCF calculation which was then interpreted and corrected using hybrid functionals. When plotting the band structure of a material, the full bands spectrum can be confusing to view (Fig. 14). The k-vector graph is a plot of the band structure in k-space, as discussed in Section 3.3. The DOS graph is a plot of the Density of States, which is not strictly important to this research (as the k-space band structure graph is sufficient), has applications to the broader chemistry of semiconductors [30].

However, the usefulness of the full bands spectrum plot is restricted as we just want to focus on the valence band maximum and conduction band minimum. Zooming in, Fig. 15 displays a closer look at the valance band maximum and conduction band minimum. From the plot, it can be gleaned that the semiconductor does have a wide band gap, spanning 5.226 eV.

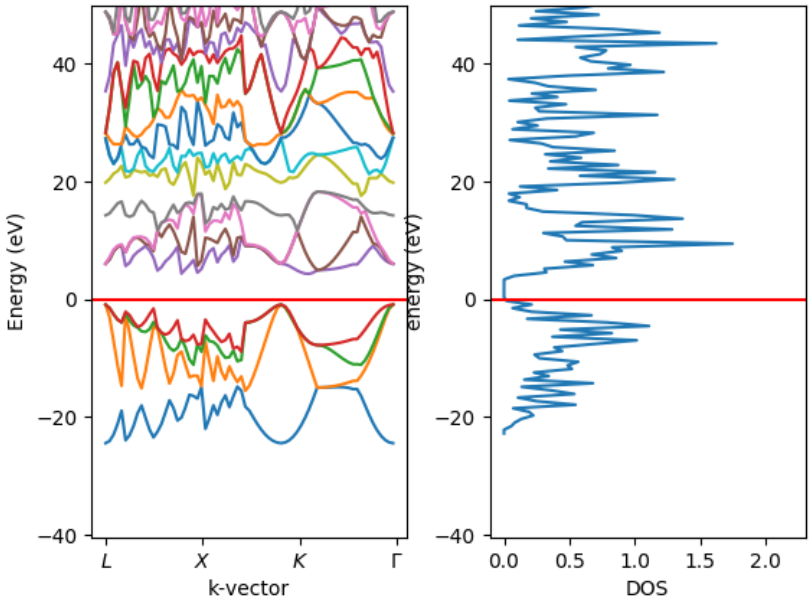


**Figure 14:** Limited bands spectrum of a pristine diamond lattice. Both the k-vectors and density of states plots are shown.

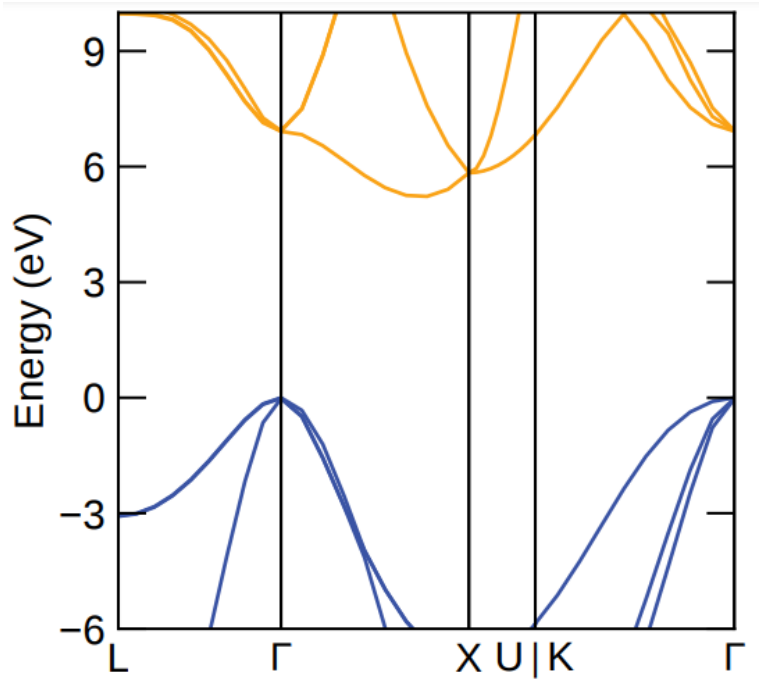


**Figure 15:** Limited bands spectrum of a pristine diamond lattice shown in k-space.

*4.1.2 Formation Energy*

When comparing formation energies across multiple defects, it is useful to plot all minimum energy lines on the same graph. This serves two purposes: firstly to see which has the lowest formation energy and secondly to see which charged defects are stable in the range [0, $E_g$] (where $E_g$ is the pristine semiconductor's band gap). The minimum energy lines are the most energetically favorable charge state at a given Fermi level for a defect. For a more in-depth explanation of this process, see Section 3.1.6.

Charge states in Fig. 16 are indicated by the slopes of the lines. For example, on the NV line, we can read off the charge states left to right as +1, 0, -1, and -2. As identified in table 1, the useful charge states for an element in group 5 (nitrogen) are $\pm 1$. Reading off the graph, the $NV^{-1}$ charge state is stable when doped between 2.76 and 4.49 eV. While the $NV^{+1}$ charge state is technically accessible, doping to Fermi levels approaching 0 with respect to the valance band maximum is much harder, so the -1 charge state is primarily targeted.

This calculation of NV in diamond formation energy serves to validate the existing setup and system parameters used. By referencing Weber *et al*. [6], comparisons can be made between formation energy plots and band gap values. Specifically, while the band gap is a little smaller than Weber *et al*., the formation energy plots are nearly identical. This is likely because Weber *et al*. used the more computationally-intensive GW correction while this research was performed with hybrid functionals. In this way, future results for semiconductors with no reference literature are also validated.

Fig. 16 also shows that the NV defect complex is energetically favorable to independent $V_C$ and $N_C$ defects. This is because the formation energy for the NV defect complex is less than the sum of independent $V_C$ and $N_C$ defects. The physical im-

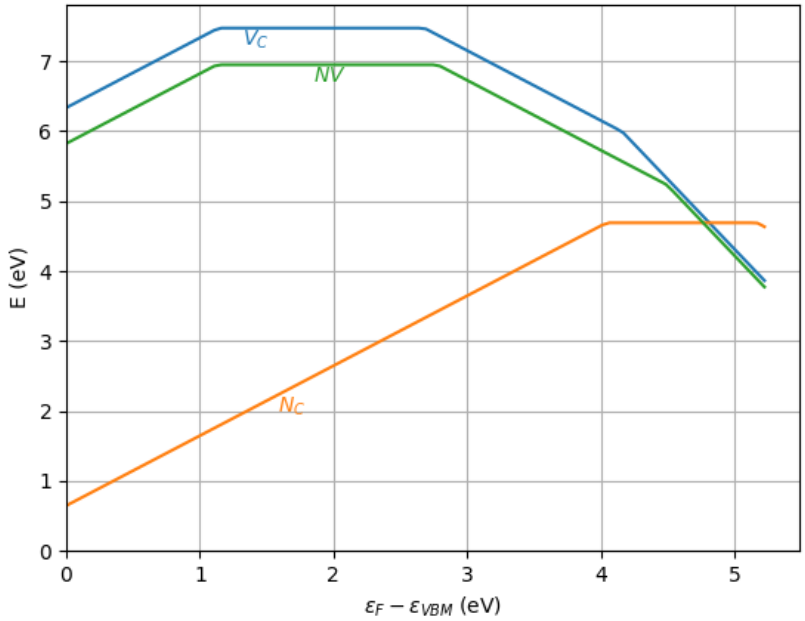


**Figure 16:** Formation energy plot for the nitrogen defect in diamond. $V_C$ represents the carbon vacancy, $N_C$ the nitrogen substitution for carbon, and $NV$ the carbon vacancy + nitrogen substitution. On the x-axis is the Fermi level ($\varepsilon_F$) in reference to the valance band maximum (VBM) of bulk diamond.

plication of this is that if both a $V_C$ and $N_C$ defect are introduced into the lattice, it is energetically favorable for them to "find" each other to form a NV defect.

## *4.2 ZnSe Results*

### *4.2.1 Band Structure*

The ZnSe band structure is obtained very similarly to diamond. Using a pristine unit cell, SCF and then NSCF calculations are run and then post-processed by the VASP Wrapper. Fig. 17 shows the bands structure for ZnSe, with the band gap equal to 1.137eV. The band gap calculations for ZnSe were not run with Hybrid Functionals (instead with SCF + NSCF), and so are subject to DFT's well-known underestimation of the band gap.

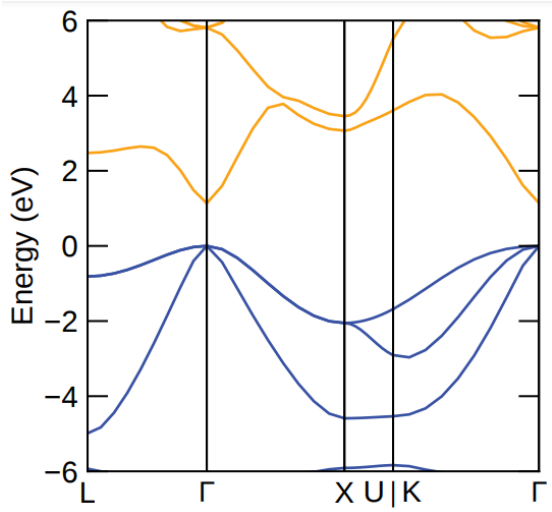


**Figure 17:** Limited bands spectrum of a pristine ZnSe lattice shown in k-space.

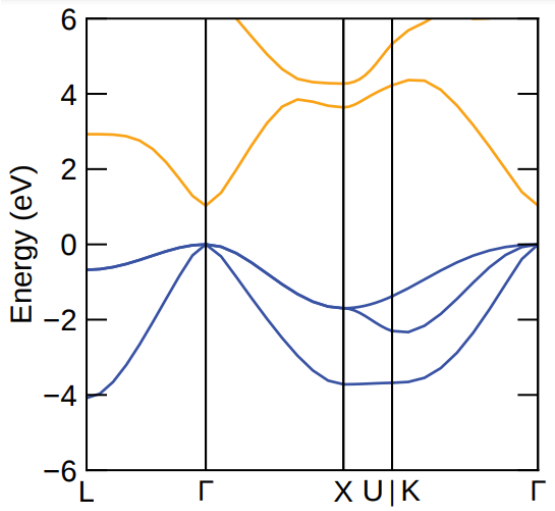


**Figure 18:** Limited bands spectrum of a pristine CdS Hawleyite lattice (as discussed in Section 3.1.2) shown in k-space.

## *4.3 CdS Results*

### *4.3.1 Band Structure*

Utilizing the methods described in Section 3.1.3, the band structure for CdS was calculated and is shown in Fig. 18. It should be noted that CdS's band structure appears very similar to ZnSe's band structure (Fig. 17). The band structures differ only slightly. One key difference between the two is that ZnSe's band gap is larger, but only by 10% as calculated here (and 17.5% with band gap corrections, as seen in literature [8, 31]). This is one of the reasons why both were chosen for this computational study.

The band gap for CdS is calculated to be 1.028 eV. Similar to ZnSe, CdS also was not run with Hybrid Functionals, and so is subject to DFT's well-known underestimation of the band gap. Hybrid Functionals are a complicated step to take, and so the method to calculate them has been shown for diamond's band gap in Appendix A. For additional readings on using Hybrid Functionals, see Ref. [32].

| Compound | A (B) | Impurities (X) | Charge Range (q) | Defect Formula |
|---|---|---|---|---|
| ZnSe | Zn (Se) | F | [-2, +3] | $F_{Zn}V_{Se}^{[-2,+3]}$ |
| | | Cl | [-2, +3] | $Cl_{Zn}V_{Se}^{[-2,+3]}$ |
| | Se (Zn) | F | [-2, +3] | $F_{Se}V_{Zn}^{[-2,+3]}$ |
| | | Cl | [-2, +3] | $Cl_{Se}V_{Zn}^{[-2,+3]}$ |
| CdS | Cd (S) | F | [-2, +3] | $F_{Cd}V_{S}^{[-2,+3]}$ |
| | | Cl | [-2, +3] | $Cl_{Cd}V_{S}^{[-2,+3]}$ |
| | S (Cd) | F | [-2, +3] | $F_{S}V_{Cd}^{[-2,+3]}$ |
| | | Cl | [-2, +3] | $Cl_{S}V_{Cd}^{[-2,+3]}$ |

**Table 2:** Selected defects ($X_BV_A^q$) and charge state ranges for ZnSe and CdS.

This project developed a complex workflow using Python tools for performing formation energy calculations in semiconductors for charged defects. It validates that workflow for the NV center in diamond at the level of DFT calculations and hybrid functionals. It also began to apply the workflow to analyze ZnSe and CdS systems.

### *4.4 Future Work*

In this project, future work includes formalizing the formation energy calculations for ZnSe and all of its intended defects. Additionally, once ZnSe has been thoroughly studied, CdS will be studied as a potential housing structure for room-temperature qubits. Since ZnSe and CdS have similar electronic structures, similar defects will be studied in both.

Future work on ZnSe and CdS begins with the calculation of Zn, Se, Cd and S chemical potentials within the confines of our current system. From there, defects such as F and Cl will be introduced according to table 2.

To study each defect, there is a 5-step process which builds on top of the research efforts already undertaken:

1. Calculate the chemical potential of the most naturally occurring form of the defect. For example, for the N defect in Diamond, the chemical potential of N is calculated using $N_2$. Alternatively, reference from literature can be drawn.

2. Modify the Python-based solver for the $V_C$ defect complex in diamond (Appendix B) for ZnSe/CdS. Plug the chemical potential into this solver for the electronic structure of ZnSe/CdS with a substitution defect. This set of calculations should include charge states in the range [-2, +3], for completeness and cohesion with all of the other defect calculations. Analyzing these results will help catch potential bugs as well as validate calculations.

3. Modify the Python-based multi-defect solver made for diamond (Appendix C) for ZnSe/CdS. Use this to solve for all 3 types of defects (host-vacancy, host-substitution, and defect-vacancy) across charge states in the range [-2, +3] for each target defect.

4. Utilize the results of step 3 to calculate and plot the formation energy for each defect.

5. Repeat steps 3 and 4 for each desired defect listed in Table 2.

After all of these steps, formation energy plots will be generated like the one in Fig. 16. Based on these plots, stable charge states for each defect can be identified and Fermi levels for the predetermined so-called useful charge states will be determined.

# CHAPTER FIVE

## 5 Conclusions

This work found formation energies for the NV defect complex in diamond, utilizing the $V_C$ and $N_C$ defect formation energies along the way. With these results, this work guides future theoretical and experimental exploration of semiconductors with electronic properties similar to that of the NV defect in diamond. Specifically, this work targets future exploration of ZnSe and CdS and provides a guide of how exactly to explore these compound semiconductors.

Through the calculation of formation energy for many defects and charge states, the determination of energetically favorable semiconductor/defect combinations can be made. The predetermined useful charged defects states can be used to make room-temperature qubits for quantum information processing (QIP). While QIP is still in its infancy, we are already starting to see potential applications in data security, protein folding, artificial intelligence/machine learning, and economics.

# APPENDICES

# APPENDIX A

## Diamond Unit Cell With Hybrid Functionals

This block of code demonstrates how to calculate the band gap of diamond using hybrid functionals in a Python environment using the *ab initio* computational simulation package VASP as well as the VASP Wrapper by John Kitchen [27]. This block of code can be run on any computing cluster with VASP.6.X installed.

———————————— BEGIN CODE BLOCK ————————————

```
#OS stuff
import subprocess as sp
import os
from pathlib import Path
import sys
#ase/vasp stuff (calculator object and atoms container)
from ase import Atoms
from ase.io import write
from ase.build import bulk
from vasp import Vasp
from vasp.vasprc import VASPRC
#math tools
import numpy as np

#used in a lot of places for file structure, etc
system_name = "diamond"
system_path = system_name + "/unitcellHybridFunctionals/"

#Create and go to the correct directory
# create the system directory as needed
Path(system_path).mkdir(parents=True, exist_ok=True)
prevDir = os.getcwd()
os.chdir(system_path)

# create the 'images' directory as needed
Path('images').mkdir(exist_ok=True)

#Node stuff
VASPRC['queue.q'] = 'batch'
queue = VASPRC['queue.q']
VASPRC['queue.nodes'] = 24
VASPRC['queue.ppn'] = 4

#system constants
a = 3.5717948718  # lattice const
ecutwfc = 750     # energy cutoff
```

```
    N_RELAX = 2       # shells of relaxation

    formula = 'C'

    #where is the pristine supercell
    ref_pot_full = "pristine/LOCPOT"

    #lattice structure for 1x1x1 cell
    lattice = [[a, 0.0, 0.0],  # work with cubic cell
    [0.0, a, 0.0],
    [0.0, 0.0, a]]

    basis = [[0.0, 0.0, 0.0],
    [0.5, 0.5, 0.0],
    [0.0, 0.5, 0.5],
    [0.5, 0.0, 0.5],
    [0.25, 0.25, 0.25],
    [0.75, 0.75, 0.25],
    [0.25, 0.75, 0.75],
    [0.75, 0.25, 0.75]]



    ############################################################
    #Hybrid Functionals band structure calculation
    ############################################################
    #Creates the pristine unit cell for you in the background
    latticeConst = a
    pristine = bulk(formula, 'diamond', a=latticeConst)
    write("images/" + system_name + f"_pristine.cube", pristine)

    r = pristine.positions
    X = pristine.symbols

    print("Locations of all the atoms:")
    for j, rj in enumerate(r):
    print('{0}: ({1}, {2}, {3})'\
    .format(X[j], rj[0], rj[1], rj[2]))

    #set up vasp object
    homedir = os.getcwd()
    imgdir = os.path.join(homedir, 'images')
    jobdir = os.path.join(homedir, 'pristine')
    VASPRC['queue.jobname'] = f"pristine_diamond"
    calc = Vasp(jobdir,  # output dir
    xc='PBE',
    kpts=[9, 9, 9],
    encut= ecutwfc,
    ismear=0, sigma=0.01,
    lcharg=True,  # you need the charge density
    lwave=True,   # and wavecar for the restart
    gamma=[0,0,0],
    atoms=pristine)
```

```
    #get the energy of the system (SCF calculation)
    Epristine = calc.potential_energy
    calc.stop_if(Epristine is None)
    os.system('rm pristine*.e*')
    print(f'The total energy is: {Epristine:8.04f} eV')
    print('forces = (eV/ang)\n {0}'\
    .format(pristine.get_forces()))

    # p = bands figure
    npoints, band_energies, p, Egap, Ecbm, Evbm =
    calc.get_bandstructure_v02(
    kpts_path=[('$L$', [0.5, 0.5, 0.5]),
    (r'$\Gamma$', [0.0, 0.0, 0.0]),
    (r'$\Gamma$', [0.0, 0.0, 0.0]),
    ('$X$', [0.0, 0.5, 0.5]),
    ('$X$', [0.0, 0.5, 0.5]),
    ('$U$', [0.25, 0.625, 0.625]),
    ('$K$', [0.375, 0.75, 0.375]),
    (r'$\Gamma$', [0, 0, 0])],
    kpts_nintersections=10, ylim=(None, 50))

    if p is None:
    calc.abort()
    else:
    print(f'###PRISTINE CALCS###\n  '\
    'EPristine: {Epristine}\n  Energy gap: {Egap}\n  '\
    'Conduction band minimum: {Ecbm}\n  '\
    'Valance band maximum: {Evbm}')

    figname = f'{imgdir}/{system_name}-fcc-'\
    'bandstructure_diamond.png'
    p.savefig(figname)
    print(f'\n#+caption: Band structure for '\
    '{system_name} crystal.\n[[./{figname}]]\n')

    # Create a band structure plot using sumo-bandplot
    banddir = os.path.join(jobdir, 'bandstructure')
    os.chdir(banddir)
    sumo_cmdlist = ['sumo-bandplot', '-d', imgdir]
    sp.Popen(sumo_cmdlist)

    figname = os.path.join(imgdir, 'band.pdf')
    figname2 = os.path.join(imgdir, f'{system_name}'\
    '-bands-DFT-HSE.pdf')
    if os.path.exists(figname):
    os.rename(figname, figname2)
    print(f'band structure plot:\n {figname2}')

    #go back to the folder where we started
    os.chdir(prevDir)

```

END CODE BLOCK

# APPENDIX B

## Solver for $V_C$ Defect Complex in Diamond

This block of code demonstrates how to calculate the formation energy for the $V_C$ defect complex in diamond for charged states in the range [-2, 3]. The code is in a Python environment and uses the *ab initio* computational simulation package VASP as well as the VASP Wrapper by John Kitchen [27]. This block of code can be run on any computing cluster with VASP.6.X installed.

———————————— BEGIN CODE BLOCK ————————————

```
#OS stuff
import subprocess as sp
import os
from pathlib import Path
import sys
#ase/vasp stuff (calculator object and atoms container)
from ase import Atoms
from ase.io import write
from ase.build import bulk
from vasp import Vasp
from vasp.vasprc import VASPRC
#math tools
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
#tools to deal with calculations
import supercell_defect_relaxation as sdr
import AnalysisTools as ant
from vasp import defectTools as dTools

#used in a lot of places for file structure, etc
system_name = "diamond"
N = 2             # NxNxN supercell
system_path = system_name + "/supercell/n_" + str(N)

#Create and go to the correct directory
# create the system directory as needed
Path(system_path).mkdir(parents=True, exist_ok=True)
prevDir = os.getcwd()
os.chdir(system_path)

# create the 'images' directory as needed
Path('images').mkdir(exist_ok=True)

```

```
    #Node stuff
    VASPRC['queue.q'] = 'batch'
    queue = VASPRC['queue.q']
    VASPRC['queue.nodes'] = 8
    VASPRC['queue.ppn'] = 8

    #system constants
    a = 3.5717948718  # lattice const
    ecutwfc = 750     # energy cutoff
    #(defined earlier for folder structure)
    #N = X            # NxNxN supercell
    nRelax = 2        # shells of relaxation
    q_min = -3        # charge min
    q_max = 3         # charge max
    eps =  5.7        # dielectric constant of diamond
    avg = 2           # averaging length, in Bohr

    #[RECALCULATED BELOW]
    #Eg = 4.141        # [eV] ESTIMATED bandgap of diamond
    mu_C = -250.80891 # [eV] Chemical potential of carbon
    mu_N = -383.9205  # [eV] Chemical potential of nitrogen
    #[RECALCULATED BELOW]
    #EVBM = 9.664      # [eV], valence band maximum
    plot_points=100   # points in a formation energy plot

    #Creates a plot that uses all lines instead of
    #  just the bottom of the plot
    PLOT_ALL_FORMATION_LINES = True

    #define the list of defects
    add_C = [-1]#-1, -2
    #add_N = [0, 1, 1]

    #define the list of charges to try for each defect
    q_arr = list(range(q_min, q_max + 1))

    #chemical formula
    formula = "C8"

    #where is the pristine supercell
    ref_pot_full = "pristine/LOCPOT"

    #lattice structure for 1x1x1 cell
    lattice = [[a, 0.0, 0.0],  # work with cubic cell
    [0.0, a, 0.0],
    [0.0, 0.0, a]]
    basis = [[0.0, 0.0, 0.0],
    [0.5, 0.5, 0.0],
    [0.0, 0.5, 0.5],
    [0.5, 0.0, 0.5],
    [0.25, 0.25, 0.25],
    [0.75, 0.75, 0.25],
    [0.25, 0.75, 0.75],
    [0.75, 0.25, 0.75]]
```

```
    '''
    Step 1: Standard self-consistent run
    See: [1] https://www.vasp.at/wiki/index.php/Si_bandstructure#
Standard_self-consistent_.28SC.29_run
    '''
    #Creates the unit cell for you in the background
    atoms = Atoms(symbols=formula,
    scaled_positions=basis,
    cell=lattice,
    pbc=(1, 1, 1))
    ######################PRISTINE Calculations
########################
    #make the pristine supercell
    pristine = sdr.makeSupercellPristine_rep(atoms, N, N, N)

    #set up vasp object
    homedir = os.getcwd()
    jobdir = os.path.join(homedir, 'pristine')
    calc = Vasp(jobdir,  # output dir
    xc='PBE',
    kpts=[9, 9, 9],
    encut= ecutwfc,
    ismear=0, sigma=0.01,
    lcharg=True,  # you need the charge density
    lwave=True,   # and wavecar for the restart
    lvtot=True,
    gamma=[0,0,0],
    atoms=pristine)

    #get the energy of the system (SCF calculation)
    Epristine = calc.potential_energy
    calc.stop_if(Epristine is None)
    _, _, p, Egap, Ecbm, Evbm =
    calc.get_bandstructure(
    kpts_path=[('$L$', [0.5, 0.5, 0.5]),
    (r'$\Gamma$', [0.0, 0.0, 0.0]),
    (r'$\Gamma$', [0.0, 0.0, 0.0]),
    ('$X$', [0.0, 0.5, 0.5]),
    ('$X$', [0.0, 0.5, 0.5]),
    ('$U$', [0.25, 0.625, 0.625]),
    ('$K$', [0.375, 0.75, 0.375]),
    (r'$\Gamma$', [0, 0, 0])],
    kpts_nintersections=10)

    if p is None:
    calc.abort()
    else:
    print(f'###PRISTINE CALCS###\n  '\
    'EPristine: {Epristine}\n  '\
    'Energy gap: {Egap}\n  '\
    'Conduction band minimum: {Ecbm}\n  '\
    'Valance band maximum: {Evbm}')
```

```
141
142     #####Start defected calculation
143     #make the defected supercell
144     #type 1 is a single atom vacancy
145     defected = sdr.add_defect(pristine.copy(), nRelax, 1, 'center')
146     #calculate the number of electrons for this defect
147     NELECT_defect = dTools.calcNElect(defected)
148
149     #draw atoms object
150     write("images/" + system_name + f"_defect_vacancy_{N}x{N}x{N}.png"
    ,
151     defected, show_unit_cell=2);
152
153     #initialize storage arrays
154     num_q_state = len( q_arr )
155     EDIFF = np.zeros( num_q_state ) # [eV] uncorrected energy
    difference
156     EDIFFCORR = np.zeros( num_q_state )  # [eV] corrected energy
    difference
157
158     #Run all calculations before trying to retrieve them
159     for i in range(0, 2):
160     # Iterate over the charge values
161     for q_idx, q in enumerate(q_arr):
162     #set up vasp object
163     homedir = os.getcwd()
164     imgdir = os.path.join(homedir, 'images')
165     defectFolder = 'vacancy_q' + str(q) + '_nRelax' + str(nRelax)
166     jobdir = os.path.join(homedir, defectFolder)
167     #*-1 because nelect is number of electrons (-1 charge each)
168     tot_num_electrons = NELECT_defect + -1*q
169     calc = Vasp(jobdir,  # output dir
170     xc='PBE',
171     kpts=[9, 9, 9],
172     encut= ecutwfc,
173     ismear=0, sigma=0.01,
174     lcharg=False,  # you need the charge density
175     lwave=False,   # and wavecar for the restart
176     #output LOCPOT file (for sxdefectalign)
177     lvtot=True,
178     ibrion=1,      # ionic relaxation
179     nelmin=4,
180     nelect=(tot_num_electrons),
181     gamma=[0,0,0],
182     atoms=defected)
183     #get the energy of the system (SCF calculation)
184     Edef = calc.potential_energy
185     print("Edef = ", Edef)
186     if i == 1:
187     calc.stop_if(Edef is None)
188     #make sure sxdefectalign is running in the
189     #  right subdirectory
190     os.chdir(defectFolder)
191     print("Running defect alignment in defect folder "\
```

```
192       + defectFolder)
193
194       EnergyCorrection =
195       ant.defectAlignment("../" + ref_pot_full, "LOCPOT",
196       N, q, ecutwfc, eps, avg, a,
197       location='center', VASP_RUN=True)
198
199       print("Energy correction for nelect "\
200       + str(tot_num_electrons)\
201       + ": " + str(EnergyCorrection))
202       Ediff = Edef - Epristine
203       Ediff_corr = Edef - Epristine + EnergyCorrection
204
205       EDIFF[q_idx] = Ediff
206       EDIFFCORR[q_idx] = Ediff_corr
207       #go back home
208       os.chdir(homedir)
209       ##end calculation loop
210       #clean up from jobs run
211       os.system('rm DefaultJob.e*')
212
213
214       #Now, the for-loop over charge states is complete. Let us
    calculate the
215       # formation energy for the defect.
216       print("EDIFF: ", EDIFF)
217       print("EDIFFCORR: ", EDIFFCORR)
218       print("Calculating formation energy for diamond vacancy")
219       print("Raw energy differences: ", EDIFF)
220
221       mu = [Epristine/((N**3)*8.0),]
222       n_atoms_added = [add_C[0],]
223
224       EFermi, EFormMin, EForm =
225       ant.formationEnergy(
226       Egap,          #ESTIMATED band gap of host material [eV]
227       plot_points,   #points in a formation energy plot
228       q_arr,         #list of charge states
229       EDIFFCORR,     #corrected energy difference [eV]
230       Evbm,          #valence band max (from own calcs) [eV]
231       mu,            #list of chem. pot.s of all elements involved
232       # by species
233       n_atoms_added) #list of number of all elements involved
234       # by species
235
236       #Plot formation energy!!!
237       plt.clf()#clear anything previously using the plot
238
239       if PLOT_ALL_FORMATION_LINES:
240       for j, qval in enumerate(q_arr):
241       temp_line = plt.plot(EFermi[0,:], EForm[j,:],
242       Label="$q = {0}$"\.format(qval))
243       idx = round((0.25)*plot_points)
244       plt.text(EFermi[0, idx], EForm[0, idx],
```

```
        '$V_C$',
        verticalalignment='top',
        horizontalalignment='left',
        color=temp_line[0].get_color())

        plt.grid(True)
        plt.legend()
        plt.xlim((0, None))
        plt.ylim((0, None))
        plt.ylabel('E (eV)')
        plt.xlabel('$\epsilon_F - \epsilon_{VBM}$ (eV)')
        plt.savefig(f'images/FormationEnergyPlot_N{N}_nl{nRelax}'\
        '_d{avg}_all.png')

        plt.clf()
        temp_line = plt.plot(EFermi[0,:], EFormMin[0,:])
        idx = round((0.25)*plot_points)
        plt.text(EFermi[0, idx], EFormMin[0, idx],
        '$V_C$',
        verticalalignment='top',
        horizontalalignment='left',
        color=temp_line[0].get_color())

        plt.grid(True)
        plt.xlim((0, None))
        plt.ylim((0, None))
        plt.ylabel('E (eV)')
        plt.xlabel('$\epsilon_F - \epsilon_{VBM}$ (eV)')
        plt.savefig(f'images/FormationEnergyPlot_N{N}_nl{nRelax}_d{avg}_'\
        'minTrace.png')

        #go back to the folder where we started
        os.chdir(prevDir)

```

END CODE BLOCK

# APPENDIX C

## Compound Solver for $V_C$, $N_C$ and NV Defect Complexes in Diamond

This block of code demonstrates how to calculate the formation energy for the $V_C$ defect complex in diamond for charged states in the range [-2, 3]. This code uses the hybrid functionals-derived band gap from Appendix A. The code is in a Python environment and uses the *ab initio* computational simulation package VASP as well as the VASP Wrapper by John Kitchen [27]. This block of code can be run on any computing cluster with VASP.6.X installed.

———————————— BEGIN CODE BLOCK ————————————

```
    #OS stuff
    import subprocess as sp
    import os
    from pathlib import Path
    import sys
    #ase/vasp stuff (calculator object and atoms container)
    from ase import Atoms
    from ase.io import write
    from ase.build import bulk
    from vasp import Vasp
    from vasp.vasprc import VASPRC
    #math tools
    import numpy as np
    import matplotlib.pyplot as plt
    from mpl_toolkits.mplot3d import Axes3D
    #tools to deal with calculations
    import supercell_defect_relaxation as sdr
    import AnalysisTools as ant
    from vasp import defectTools as dTools

    #used in a lot of places for file structure, etc
    system_name = "diamond"
    N = 2             # NxNxN supercell
    system_path = system_name + "/supercellHybridFunctionals/n_" + str
(N)

    #Create and go to the correct directory
    # create the system directory as needed
    Path(system_path).mkdir(parents=True, exist_ok=True)
    prevDir = os.getcwd()
    os.chdir(system_path)
    # create the 'images' directory as needed
```

```
32      Path('images').mkdir(exist_ok=True)
33
34      #Node stuff
35      VASPRC['queue.q'] = 'batch'
36      queue = VASPRC['queue.q']
37      VASPRC['queue.nodes'] = 16
38      VASPRC['queue.ppn'] = 4
39
40      #system constants
41      a = 3.5717948718  # lattice const
42      ecutwfc = 750     # energy cutoff
43      #(defined earlier for folder structure)
44      #N = X            # NxNxN supercell
45      N_RELAX = 2       # shells of relaxation
46      q_min = -3        # charge min
47      q_max = 3         # charge max
48      eps =  5.7        # dielectric constant of diamond
49      avg = 2           # averaging length, in Bohr
50
51      #obtained from my Hybrid Functionals calculations
52      EgapHybrid = 5.22599  # [eV] ESTIMATED bandgap of diamond
53      #[RECALCULATED BELOW]
54      #mu_C = -250.80891    # [eV] Chemical potential of carbon
55      mu_N = -8.3234        # [eV] Chemical potential of nitrogen
56      #obtained from my Hybrid Functionals calculations
57      EvbmHybrid = 9.13     # [eV], valence band maximum
58      plot_points=100       # points in a formation energy plot
59
60      #Creates a formation energy plot that uses all lines
61      # instead of just the bottom of the plot
62      PLOT_ALL_FORMATION_LINES = True
63
64      ##Defect constants -- doing this with an array of objects instead
    of
65      ##  parallel arrays for clarity
66      # Defect codes
67      #    1 --> vacancy
68      #    2 --> substitution
69      #    3 --> vacancy and adjacent substitution
70      # shortName    = name that we can put on files, no spaces
71      # fullName     = full name for the ouptut
72      # latexSym     = usually for graphs
73      # substitution = symbol of substituted element (only for defect
    code 2)
74      # neighborSub  = symbol of substituted element in neighbor (only
    for
75      #   defect code 3)
76      class DefectContainer:
77      def __init__(self, shortName, fullName, latexSym, addedC, addedN,
78      defectCode, substitution, neighborSub):
79      self.shortName = shortName
80      self.fullName  = fullName
81      self.latexSym  = latexSym
82      self.addedC    = addedC
```

```
        self.addedN    = addedN
        self.defectCode = defectCode
        self.substitution = substitution
        self.neighborSub  = neighborSub

        defectContainers = [DefectContainer(
        shortName = 'C_vac',
        fullName  = 'carbon vacancy',
        latexSym  = '$V_C$',
        addedC    = -1,
        addedN    = 0,
        defectCode = 1, #vacancy
        substitution = None,
        neighborSub  = None
        ), DefectContainer(
        shortName = 'N_sub',
        fullName  = 'nitrogen substitution',
        latexSym  = '$N_C$',
        addedC    = -1,
        addedN    = 1,
        defectCode = 2, #substitution
        substitution = 'N',
        neighborSub  = None
        ), DefectContainer(
        shortName = 'NV',
        fullName  = 'nitrogen vacancy',
        latexSym  = '$NV$',
        addedC    = -2,
        addedN    = 1,
        defectCode = 3, #vacancy and adjacent substitution
        substitution = None,
        neighborSub  = 'N'
        )]
        DEFECT_LOC = 'center'

        #define the list of charges to try for each defect
        q_arr = list(range(q_min, q_max + 1))

        #chemical formula
        formula = "C8"

        #where is the pristine supercell
        ref_pot_full = "pristine/LOCPOT"

        #lattice structure for 1x1x1 cell
        lattice = [[a, 0.0, 0.0],  # work with cubic cell
        [0.0, a, 0.0],
        [0.0, 0.0, a]]

        basis = [[0.0, 0.0, 0.0],
        [0.5, 0.5, 0.0],
        [0.0, 0.5, 0.5],
        [0.5, 0.0, 0.5],
        [0.25, 0.25, 0.25],
```

```
    [0.75, 0.75, 0.25],
    [0.25, 0.75, 0.75],
    [0.75, 0.25, 0.75]]



    #'''
    #Step 1: Standard self-consistent run
    #   See: [1] https://www.vasp.at/wiki/index.php/Si_bandstructure#Standard_self-consistent_.28SC.29_run
    #'''
    #Creates the unit cell for you in the background
    atoms = Atoms(symbols=formula,
    scaled_positions=basis,
    cell=lattice,
    pbc=(1, 1, 1))

    ######################PRISTINE Calculations#########################
    #make the pristine supercell
    pristine = sdr.makeSupercellPristine_rep(atoms, N, N, N)
    write("images/" + system_name + f"_pristine_{N}x{N}x{N}.cube", pristine)

    #set up vasp object
    homedir = os.getcwd()
    jobdir = os.path.join(homedir, 'pristine')
    VASPRC['queue.jobname'] = f"pristine_{N}x{N}x{N}"
    calc = Vasp(jobdir,  # output dir
    xc='PBE',
    kpts=[9, 9, 9],
    encut= ecutwfc,
    ismear=0, sigma=0.01,
    lcharg=True,  # you need the charge density
    lwave=True,   # and wavecar for the restart
    lvtot=True,
    gamma=[0,0,0],
    atoms=pristine)

    #get the energy of the system (SCF calculation)
    Epristine = calc.potential_energy
    calc.stop_if(Epristine is None)
    _, _, p, Egap, Ecbm, Evbm =
    calc.get_bandstructure(
    kpts_path=[('$L$', [0.5, 0.5, 0.5]),
    (r'$\Gamma$', [0.0, 0.0, 0.0]),
    (r'$\Gamma$', [0.0, 0.0, 0.0]),
    ('$X$', [0.0, 0.5, 0.5]),
    ('$X$', [0.0, 0.5, 0.5]),
    ('$U$', [0.25, 0.625, 0.625]),
    ('$K$', [0.375, 0.75, 0.375]),
    (r'$\Gamma$', [0, 0, 0])],
    kpts_nintersections=10)

```

```
    if p is None:
    calc.abort()
    else:
    print(f'###PRISTINE SCF CALCS###\n  '\
    'EPristine: {Epristine}\n  '\
    'Energy gap: {Egap}\n  '\
    'Conduction band minimum: {Ecbm}\n  '\
    'Valance band maximum: {Evbm}')
    os.system('rm pristine_*.e*')

    #Precaution: since I'm using the hybrid functional values for
    #  Egap and Evbm
    Egap = Ecbm = Evbm = None

    ######################DEFECTED Calculations
##########################
    #initialize storage arrays
    numChargeStates = len(q_arr)
    numDefects      = len(defectContainers)
    for i in defectContainers:
    i.EDiff     = np.zeros(numChargeStates)
    i.EDiffCorr = np.zeros(numChargeStates)

    #initialize chemical potential array
    #carbon chemical potential is the
    #  pristine energy / number of carbon atoms
    mu_C = Epristine/((N**3)*8.0)
    mu = [mu_C, mu_N] #chemical potentials
    print("Using carbon chemical potential of " + str(mu[0])\
    + " and a nitrogen chemical potential of " + str(mu[1]))

    #Run all calculations (i=0) before trying to retrieve them and
    # use them (i=1)
    for i in range(0, 2):
    # Iterate over defects
    for dfObj in defectContainers:
    print(f"###RUNNING DEFECT CALCS FOR {dfObj.shortName}, i={i} "\
    "(0 = calc run, 1 = post processing)##")
    #make the defected supercell
    #type 1 is a single atom vacancy
    defected = sdr.add_defect(pristine.copy(),
    N_RELAX,
    dfObj.defectCode,
    DEFECT_LOC,
    dfObj.substitution,
    substitutionNeighbor=dfObj.neighborSub)
    print(f"################################################"\
    "N_RELAX={N_RELAX}, "\
    "dfObj.defectCode={dfObj.defectCode}, "\
    "DEFECT_LOC={DEFECT_LOC}")

    #calculate the number of electrons for this defect
    NELECT_defect = dTools.calcNElect(defected)

```

```
        #draw atoms object
        write("images/" + system_name
        + f"_defect_{dfObj.shortName}_{N}x{N}x{N}.png",
        defected, show_unit_cell=2)
        write("images/" + system_name
        + f"_defect_{dfObj.shortName}_{N}x{N}x{N}.cube", defected)

        # Iterate over the charge values
        for q_idx, q in enumerate(q_arr):
        print(f"#####Charge {q}#####")
        #set up vasp object
        #homedir = os.getcwd()#[already calculated]
        imgdir = os.path.join(homedir, 'images')
        defectFolder = f'{dfObj.shortName}/q{q}_nRelax{N_RELAX}'
        jobdir = os.path.join(homedir, defectFolder)
        #*-1 because nelect is number of electrons (-1 charge each)
        tot_num_electrons = NELECT_defect + -1*q
        VASPRC['queue.jobname'] =
        f"{dfObj.shortName}_q{q}_{N}x{N}x{N}"
        calc = Vasp(
        jobdir,  # output dir
        xc='PBE',
        kpts=[9, 9, 9],
        encut= ecutwfc,
        ismear=0, sigma=0.01,
        lcharg=False,  # you need the charge density
        lwave=False,   # and wavecar for the restart
        lvtot=True,    # output LOCPOT file (for sxdefectalign)
        ibrion=1,      # ionic relaxation
        nelmin=4,      # used with ionic relaxation
        nelect=(tot_num_electrons),
        gamma=[0,0,0],
        atoms=defected)
        #get the energy of the system (SCF calculation)
        Edef = calc.potential_energy
        print("Edef = ", Edef)
        if i == 1:
        calc.stop_if(Edef is None)
        #make sure sxdefectalign is running in
        # the right subdirectory
        os.chdir(defectFolder)

        print("Running defect alignment in defect folder "
        + defectFolder)

        EnergyCorrection = ant.defectAlignment(
        homedir + "/" + ref_pot_full, "LOCPOT",
        N, q, ecutwfc, eps, avg, a,
        location=DEFECT_LOC, VASP_RUN=True)

        print("Energy correction for nelect "
        + str(tot_num_electrons) + ": "
        + str(EnergyCorrection))
        dfObj.EDiff[q_idx] = Edef - Epristine
```

```
295       dfObj.EDiffCorr[q_idx] =
296       Edef - Epristine + EnergyCorrection
297
298       #go back home
299       os.chdir(homedir)
300       #ENDIF defect alignment
301       #ENDFOR charge loop
302
303       #############DEFECTED Post ProcessingCalculations###############
304       #if we're in the postprocessing stage
305       if i == 1:
306       #Now, the for-loop over charge states is complete.
307       # Let us calculate the formation energy for the defect.
308       print("Calculating formation energy for " + dfObj.fullName)
309       print("Raw energy differences: ", dfObj.EDiff[:])
310
311       #number of added/removed host and defect atoms
312       n_atoms_added = [dfObj.addedC, dfObj.addedN]
313       EFermi, EFormMin, EForm =
314       ant.formationEnergy(
315       #ESTIMATED band gap of host material [eV]
316       EgapHybrid,
317       #points in a formation energy plot
318       plot_points,
319       #list of charge states
320       q_arr,
321       #corrected energy difference [eV]
322       dfObj.EDiffCorr,
323       #valence band max (from own calcs) [eV]
324       EvbmHybrid,
325       #list of chem. potentials of all elements
326       # involved by species
327       mu,
328       #list of number of all elements
329       # involved by species
330       n_atoms_added)
331
332       #Store for convenience and later use
333       dfObj.EFermi   = EFermi
334       dfObj.EFormMin = EFormMin
335       dfObj.EForm    = EForm
336       ##end calculation loop
337       #clean up from jobs run
338       os.system('rm *_*x*x*.e*')
339
340
341
342       ######################Formation Energy Plots
    !#########################
343       plt.clf()#clear anything previously using the plot
344
345       #If we want a plot of all the formation energy lines,
346       # not just the bottom trace of the whole plot
347       if PLOT_ALL_FORMATION_LINES:
```

```
        k = 0 #moves the latex label down the line on the plot

        #for each defect
        for dfObj in defectContainers:
        plt.clf()#clear anything previously using the plot

        #for each charge state
        for j, qval in enumerate(q_arr):
        temp_line = plt.plot(dfObj.EFermi[0,:],
        dfObj.EForm[j,:],
        Label="$q = {0}$".format(qval))
        idx = round((0.25 + k*0.05)*plot_points)
        plt.text(dfObj.EFermi[0, idx],
        dfObj.EForm[0, idx],
        dfObj.latexSym,
        verticalalignment='top',
        horizontalalignment='left',
        color=temp_line[0].get_color())

        #set plot settings
        plt.grid(True)
        plt.legend()
        plt.xlim((0, None))
        plt.ylim((None, None))
        plt.ylabel('E (eV)')
        plt.xlabel('$\epsilon_F - \epsilon_{VBM}$ (eV)')
        plt.savefig(f'images/FormationEnergyPlot_{dfObj.shortName}_'\
        'N{N}_nRelax{N_RELAX}_d{avg}_all.png')

        #move label left to right
        k += 1

        #Plot the actual formation energy plot (min trace of graph)
        plt.clf()#clear anything previously using the plot
        k = 0 #moves the latex label down the line on the plot
        for dfObj in defectContainers:
        temp_line = plt.plot(dfObj.EFermi[0,:], dfObj.EFormMin[0,:])
        idx = round((0.25 + k*0.05)*plot_points)
        plt.text(dfObj.EFermi[0, idx], dfObj.EFormMin[0, idx],
        dfObj.latexSym,
        verticalalignment='top',
        horizontalalignment='left',
        color=temp_line[0].get_color())

        #move label left to right
        k += 1

        plt.grid(True)
        plt.xlim((0, None))
        plt.ylim((0, None))
        plt.ylabel('E (eV)')
        plt.xlabel('$\epsilon_F - \epsilon_{VBM}$ (eV)')
        plt.savefig(f'images/FormationEnergyPlot_N{N}_'\
        'nRelax{N_RELAX}_d{avg}_minTrace.png')
```

```
402
403         #go back to the folder where we started
404         os.chdir(prevDir)
405
```

END CODE BLOCK

# BIBLIOGRAPHY

[1] H. Haffner, C. Roos, and R. Blatt, "Quantum computing with trapped ions," vol. 469, no. 4, pp. 155–203. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0370157308003463

[2] C. C. McGeoch, "Adiabatic quantum computation and quantum annealing: Theory and practice," vol. 5, no. 2, pp. 1–93. [Online]. Available: http://www.morganclaypool.com/doi/abs/10.2200/S00585ED1V01Y201407QMC008

[3] A. P. M. Place, L. V. H. Rodgers, P. Mundada, B. M. Smitham, M. Fitzpatrick, Z. Leng, A. Premkumar, J. Bryon, A. Vrajitoarea, S. Sussman, G. Cheng, T. Madhavan, H. K. Babla, X. H. Le, Y. Gang, B. Jäck, A. Gyenis, N. Yao, R. J. Cava, N. P. de Leon, and A. A. Houck, "New material platform for superconducting transmon qubits with coherence times exceeding 0.3 milliseconds," vol. 12, no. 1, p. 1779. [Online]. Available: http://www.nature.com/articles/s41467-021-22030-5

[4] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," vol. 140, no. 4, pp. A1133–A1138. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.140.A1133

[5] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," vol. 136, no. 3, pp. B864–B871. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.136.B864

[6] J. R. Weber, W. F. Koehl, J. B. Varley, A. Janotti, B. B. Buckley, C. G. Van de Walle, and D. D. Awschalom, "Quantum computing with defects," vol. 107, no. 19, pp. 8513–8518. [Online]. Available: http://www.pnas.org/content/107/19/8513.abstract

[7] C. G. Van de Walle and J. Neugebauer, "First-principles calculations for defects and impurities: Applications to III-nitrides," vol. 95, no. 8, pp. 3851–3879. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.1682673

[8] L. S. dos Santos, W. G. Schmidt, and E. Rauls, "Group-VII point defects in ZnSe," vol. 84, no. 11, pp. 115–201, publisher: American Physical Society. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.84.115201

[9] . O. Soykal, P. Dev, and S. E. Economou, "Silicon vacancy center in 4 h -SiC: Electronic structure and spin-photon interfaces," vol. 93, no. 8, p. 081207. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.93.081207

[10] R. Nagy, M. Widmann, M. Niethammer, D. B. Dasari, I. Gerhardt, n. O. Soykal, M. Radulaski, T. Ohshima, J. Vučković, N. T. Son, I. G. Ivanov, S. E. Economou, C. Bonato, S.-Y. Lee, and J. Wrachtrup, "Quantum properties of dichroic silicon vacancies in silicon carbide," vol. 9, no. 3, p. 034022. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevApplied.9.034022

[11] M. T. Man and H. S. Lee, "Interband transition and confinement of charge carriers in CdS and CdS/CdSe quantum dots," vol. 24, no. 5, pp. 167–171. [Online]. Available: http://koreascience.or.kr/journal/view.jsp?kj=E1VSCA&py=2015&vnc=v24n5&sp=167

[12] C. Freysoldt, B. Grabowski, T. Hickel, J. Neugebauer, G. Kresse, A. Janotti, and C. G. Van de Walle, "First-principles calculations for point defects in solids," vol. 86, no. 1, pp. 253–305. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.86.253

[13] S. C. Erwin, L. Zu, M. I. Haftel, A. L. Efros, T. A. Kennedy, and D. J. Norris, "Doping semiconductor nanocrystals," vol. 436, no. 7047, pp. 91–94. [Online]. Available: http://www.nature.com/articles/nature03832

[14] P. Reddy, M. P. Hoffmann, F. Kaess, Z. Bryan, I. Bryan, M. Bobea, A. Klump, J. Tweedie, R. Kirste, S. Mita, M. Gerhold, R. Collazo, and Z. Sitar, "Point defect reduction in wide bandgap semiconductors by defect quasi fermi level control," vol. 120, no. 18, p. 185704. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.4967397

[15] S. Clark, "Complex structures in tetrahedral semiconductors." [Online]. Available: https://www.semanticscholar.org/paper/Complex-structures-in-tetrahedrally-bonded-Clark/c783b6f0f71c4559c9294fc30885a22c56d6ba1e

[16] J. Hundley, "Multi-scale modeling of metal-composite interfaces in titanium-graphite fiber metal laminates part i: Molecular scale - scientific figure on ResearchGate." [Online]. Available: https://www.researchgate.net/figure/Titanium-dioxide-unit-cell-and-supercell-configurations-used-for-molecular-simulation_fig5_254112492

[17] S.-E. Ungersböck, "Advanced modeling of strained CMOS technology: Crystallographic unit cell (unit cube) of the diamond structure (image)." [Online]. Available: https://www.iue.tuwien.ac.at/phd/ungersboeck/diss.html

[18] A. Kuronen, "Potential models for diamond and zincblende structures." [Online]. Available: www.acclab.helsinki.fi/~aakurone/atomistiset/lecturenotes/lecture09_2up.pdf

[19] "Cadmium sulfide." [Online]. Available: https://chem.libretexts.org/@go/page/35889

[20] M. K. Y. Chan and G. Ceder, "Efficient band gap prediction for solids," vol. 105, no. 19, p. 196403. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.105.196403

[21] X. Zheng, A. J. Cohen, P. Mori-Sánchez, X. Hu, and W. Yang, "Improving band gap prediction in density functional theory from molecules to solids," vol. 107, no. 2, p. 026403. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.107.026403

[22] H. Lambert and F. Giustino, "*Ab initio* sternheimer-GW method for quasiparticle calculations using plane waves," vol. 88, no. 7, p. 075117. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.88.075117

[23] J. Heyd, G. E. Scuseria, and M. Ernzerhof, "Hybrid functionals based on a screened coulomb potential," vol. 118, no. 18, pp. 8207–8215. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.1564060

[24] B. Santra, "Density-functional theory exchange-correlation functionals for hydrogen bonds in water," publisher: Technische Universität Berlin. [Online]. Available: https://depositonce.tu-berlin.de/handle/11303/2959

[25] C. Freysoldt, "Sxdefectalign." [Online]. Available: https://sxrepo.mpie.de/attachments/download/36/sxdefectalign-manual.pdf

[26] J. Hafner, "*Ab-initio* simulations of materials using VASP: Density-functional theory and beyond," vol. 29, no. 13, pp. 2044–2078. [Online]. Available: http://doi.wiley.com/10.1002/jcc.21057

[27] J. Kitchin, "VASP wrapper." [Online]. Available: https://github.com/jkitchin/vasp

[28] E. Dickey and E. Blair, "VASP ASE kodiak." [Online]. Available: https://github.com/enriquepacis/vasp-ase-kodiak

[29] D. Broberg, B. Medasani, N. E. R. Zimmermann, G. Yu, A. Canning, M. Haranczyk, M. Asta, and G. Hautier, "PyCDT: A python toolkit for modeling point defects in semiconductors and insulators," vol. 226, pp. 165–179. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010465518300079

[30] M. S. Shell, P. G. Debenedetti, and A. Z. Panagiotopoulos, "An improved monte carlo method for direct calculation of the density of states," vol. 119, no. 18, pp. 9406–9411. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.1615966

[31] E. Ekuma, L. Franklin, G. Zhao, J. Wang, and D. Bagayoko, "Ab-initio local density approximation description of the electronic properties of zinc blende cadmium sulfide (zb-CdS)," vol. 406, no. 8, pp. 1477–1480. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921452611000895

[32] A. J. Garza and G. E. Scuseria, "Predicting band gaps with hybrid density functionals," vol. 7, no. 20, pp. 4165–4170. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.jpclett.6b01807