EMPIRICAL PSEUDOPOTENTIAL METHOD FOR MODERN ELECTRON SIMULATION

by

ADAM LEE STRICKLAND

B.S., University of Colorado Denver, 2007

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Electrical Engineering

2014

This thesis for the Master of Science degree by

Adam Lee Strickland

has been approved for the Electrical Engineering Program by

Hamid Z. Fardi, Chair

Miloje Radenkovic

Jae-Do Park

April 30, 2014

Strickland, Adam Lee (M.S. Electrical Engineering)

Empirical Pseudopotential Method for Modern Electron Simulation

Thesis directed by Professor Hamid Z. Fardi

ABSTRACT

The topic of this thesis is to develop a framework for finding critical values that can be used in simulating modern semiconductor device structures. Many current simulation tools use effective mass approximations to model the propagation of electrons through a device. Because modern devices take advantage of different materials, geometries and sizes, a band structure calculation becomes necessary for representing effects such as strained materials and hot carriers. While highly accurate, calculation-intense methods exist, a band structure may need to be recalculated on-the-fly creating the need for a fast, simple algorithm that can be tailored for the situation at hand.

Fortunately, there is a very well established method for calculating band structure based on physical properties of the material. Historically, these properties were measured, but modern techniques involve using algorithms to converge to the proper values. This allows the simulation of devices with interesting properties while maintaining relatively simple models.

The form and content of this abstract are approved. I recommend its publication.

Approved: Hamid Z. Fardi

CONTENTS

Chapter

1 Introduction 1

1.1 Device Simulation 2

1.2 A Breif History of Pseudopotentials and Band Structure Calculation 3

2 Empirical Pseudopotential Method 5

2.1 EPM Proof 5

2.2 Method for Band Structure Calculation 6

2.3 Band Structure of Common Semiconductors 8

2.4 Finding Pseudopotentials 8

3 Applications 17

3.1 Alloy Materials 18

3.2 Genetic Algorithms 19

3.3 Future Work 19

Bibliography 22

Appendix

A Acronyms 26

B Code Listing and Examples 27

B.1 EPM Band Structure in C++ 27

B.2 Gnuplot Example 32

TABLES

Table

2.1 Form factors of Si, Ge, Sn, GaP and GaAs at 300K from [11] 8

3.1 Form factors of SiGe Alloy for different concentrations at 300K. 18

FIGURES

Figure

1.1 A graphical representation of pseudopotential simplification 4

2.1 Si band structure with form factors from [11] 9

2.2 Ge band structure with form factors from [11] 9

2.3 Sn band structure with form factors from [11] 10

2.4 GaP band structure with form factors from [11] 10

2.5 GaAs band structure with form factors from [11] 11

2.6 AlSb band structure with form factors from [11]. 11

2.7 InP band structure with form factors from [11] 12

2.8 GaSb band structure with form factors from [11] 12

2.9 InAs band structure with form factors from [11]. 13

2.10 InSb band structure with form factors from [11]. 13

2.11 ZnS band structure with form factors from [11] 14

2.12 ZnSe band structure with form factors from [11]. 14

2.13 ZeTe band structure with form factors from [11]. 15

2.14 CdTe band structure with form factors from [11] 15

3.1 The Band Structure for Different Concentrations of Silicon-Germanium Alloy . . 21

1. Introduction

Knowing the band structure of a material is important for relating its optical and electronic properties. As new and exotic materials are studied, there is a greater need for energy band calculations. Special physical properties can be studied without building devices. Band structure calculations can be used to engineer custom band structures to tailor optical or electrical devices.

Though calculation of band structure is valuable in its own right, in this paper the focus will be on its importance with regard to semiconductor simulation. In order to understand, design and model new devices it is imperative to know the limitations and applications of different methods. For calculating band structure there are two basic approaches: ab initio and empirical methods. Ab initio, or "from the beginning" methods involve calculation of band structure by use of first-principles without using measured data. Empirical methods take advantage of experimental data to give more accurate band structure representation. Generally, ab initio methods are calculation intensive but give better insight on how the structure arrives. Both methods have their place in calculating and engineering transport properties in semiconductor devices.

Examples of ab initio methods are the Hartree-Fock[44] and Density Functional Theory (DFT)[39]. Hartree-Fock method is Based on Linear Combination of Atomic Orbitals (LCAO) and uses the muffin-tin atom orbital as the method. Several other methods based on LCAO can be employed. The DFT method projects the system of interacting particles into a system of non-interacting particles. Ab initio methods are useful for looking into properties of new materials, such as the currently popular graphene material. QuantumWise[1] is an example of a program that uses ab initio methods for simulations. It will be shown that empirical methods can also be used for exciting materials without being as bulky as first principle based methods.

Some common empirical methods are k·p and EPM. k·p takes advantage of effective mass theory in its calculations. EPM is the topic of this paper and its applications and limits will be

discussed further in the following sections.

1.1 Device Simulation

Semiconductor device simulation can be grouped into three basic categories: Classical, Semi-classical and Quantum. The manufacturing process for semiconductor devices requires that logic and electrical performances be simulated long before they go into production. Most simulators use classical techniques with parameters belted on to account for special situations. The classical approach is still used because it enables complex devices to be simulated quickly. Semi-classical approaches have evolved as a way of engineering, studying and even revealing parameters for classical simulations in new materials. In all of these methods the Poisson equation is used to describe the electrostatic part of the system,

$$\epsilon\delta\varphi = q(n - p - C) \tag{1.1}$$

where the space charge density, $\rho = q(n - p - C)$ consists of the charge of an electron q multiplied by the difference of the number of electron (n), hole (p) carriers and the concentration of charges C. Each method takes "moments" of the Boltzmann equation and simplifies them to self-consistently solve the system. The Drift-Diffusion model is the simplest and it uses current continuity to solve both static devices and slow changes in current. The model focuses on two mechanisms, namely drift of charge carriers due to an external electric field and diffusion caused by a gradient in charge carrier concentration.

The major simplification of Drift-Diffusion is that it assumes that lattice temperature and charge carrier temperature is equal. However, this is almost never the case. The Hydrodynamic model is somewhat more complex but accounts for the constant change in temperature inherent in all devices.

Semi-classical simulations involve calculating the density of states of the system (and therefore the band structure) and applying it the moments of the Boltzmann Transport Equation. For whichever assumptions and simplifications that are made the equations will be solve self consistently

with the Poisson equation.

Finally, The Wigner function approach removes all ballistic, classical properties of the system by replacing the Poisson equation with the Wigner-Boltzmann equation.

1.2 A Breif History of Pseudopotentials and Band Structure Calculation

Since core electrons in the inner shells of atoms are strongly bound, they do not play a significant part in the electronic and optical properties. The Philips Kleinman cancellation theorem[37] creates a smooth wave function to represent the core energies and potentials by taking advantage of crystal symmetry, and allows for simple calculation of wave energies. The previous method proposed by Herring in 1940 (and still of importance when core electrons must be taken into account) involved orthogalization of each plane wave creating a linear combination of core orbitals[28]. This method was successful in improving accuracy for specific applications, but was admittedly laborious. Spherical symmetry of the core was lost due to signalizing a single plane wave to the core functions, and this limited its use to metallic structures. In 1953, Slater effectively solved the symmetry issue by using Bloch waves to represent the periodicity of a crystal. The final step to what we call EPM was to smooth the contribution of the core atoms by using an effective potential. The secondary benefit of doing this is that molecules can also be modeled in the system. [37] discusses the possibility of modifying the repulsive potential to reproduce the free-atom term values. This ironically hints at the powerful adaptability of the model.

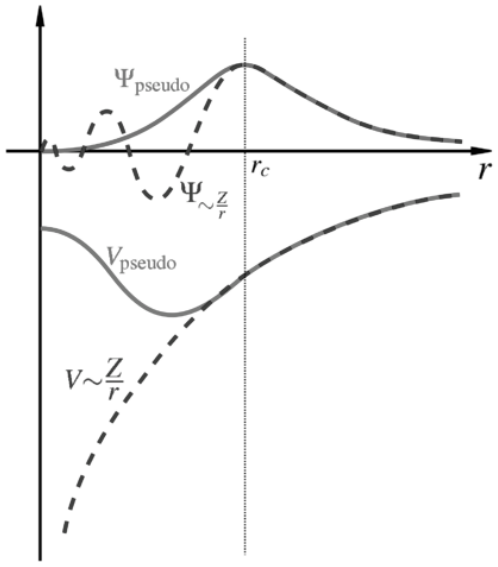


Figure 1.1: A graphical representation of pseudopotential simplification

2. Empirical Pseudopotential Method

2.1 EPM Proof

Herring's Orthogonalized Plane Wave (OPW) method is the starting point for EPM. By assuming s or p atomic symmetry Ψ_k is constructed to be orthogonal to the core states,

$$\Psi_k = \varphi_k + \sum_t a_{k,t}\phi_{k,t} \tag{2.1}$$

where $a_{k,t}$ are the orthogal coefficients and $\phi_{k,t}$ are the core wave functions. We seek a solution that gives φ_k as a smooth function. Using the Schrodinger equation,

$$H\Psi_k = E\Psi_k \tag{2.2}$$

and substituting (2.1) into (2.2) we get,

$$H\varphi_k + \sum_t a_{k,t}(E - E_t)\phi_{k,t} = E\varphi_k \tag{2.3}$$

Introducing

$$V_r = \sum_t a_{k,t}(E - E_t)\phi_{k,t}/\varphi_k \tag{2.4}$$

where V_r represents the short-range repulsive potential; we have the desired form to find the "smooth" function,

$$(H + V_r)\varphi_k = E\varphi_k \tag{2.5}$$

By adding the Hamiltonian operator,

$$H = \frac{p^2}{2m} + V_c \tag{2.6}$$

where V_c represents the core potential. Inserting (2.6) into (2.5), we get the eigenvalue problem,

$$(\frac{p^2}{2m} + V_c + V_r)\varphi_k = E\varphi_k \tag{2.7}$$

It can be seen from (2.7) that the core and the short-range potentials can be added together to equal a pseudopotential $V_p = V_c + V_r$, which is known as the Phillips-Kleinman cancellation theorem. The eigenvalue energy of (2.7) is the actual energy of the crystal wave Φ_k. and φ_k can be thought of as the pseudo-wave function.

The actual contributions of each orbital can be added up through LCAO to create V_p, but historically EPM used potentials measured empirically. Modern methods use other known properties to converge to useful pseudopotentials for different applications or materials.

2.2 Method for Band Structure Calculation

Solving the Schrodinger equation is all that is needed to find the band structure of a particular material. Unfortunately, this is a nontrivial exercise. We will focus on discreet points along the Brillouin Zone (BZ). Starting with the Time Independent Schrodinger Equation (TISE),

$$[-\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r})]\mathbf{\Psi}(\mathbf{k},\mathbf{r}) = \mathbf{E}(\mathbf{k})\mathbf{\Psi}(\mathbf{k},\mathbf{r}) \tag{2.8}$$

Both the wave function and the potential must be descritized. Since the wave function is periodic, it can be written in Bloch form and as a series,

$$\mathbf{\Psi}(\mathbf{k},\mathbf{r}) = e^{i\mathbf{kr}}u_{\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{kr}}\sum_{\mathbf{h}} A_h e^{i\mathbf{K_h r}} \tag{2.9}$$

where $\mathbf{k}$ is the crystal wave vector, $\mathbf{r}$ is the position and $u_{\mathbf{k}}(\mathbf{r})$ is a periodic function with the same periodicity as the crystal. Similarly the potential is periodic,

$$V(\mathbf{r}) = \sum_{\mathbf{m}} V_{\mathbf{m}} e^{i\mathbf{K_m r}} \tag{2.10}$$

By substituting (2.9) and (2.10) into (2.8) we arrive at a nearly discrete function,

$$\frac{\hbar^2}{2m}\sum_{\mathbf{h}}|\mathbf{k}+\mathbf{K_h}|^2 A_{\mathbf{h}} e^{i(\mathbf{k}+\mathbf{K_h})\mathbf{r}} + \sum_{\mathbf{m}}\sum_{\mathbf{h}} V_m A_{\mathbf{h}} e^{i(\mathbf{k}+\mathbf{K_h}+\mathbf{K_m})\mathbf{r}} = E(\mathbf{k})\sum_{\mathbf{h}} A_{\mathbf{h}} e^{i(\mathbf{k}+\mathbf{K_h})\mathbf{r}} \tag{2.11}$$

To take advantage of orthogonality (2.11) is multiplied by $e^{i(\mathbf{k}+\mathbf{K_h})\mathbf{r}}$ and integrated to give,

$$\frac{\hbar^2}{2m}\sum_{\mathbf{h}}|\mathbf{k}+\mathbf{K_h}|^2 A_{\mathbf{h}}\delta_{h,l} + \sum_{\mathbf{m}}\sum_{\mathbf{h}} V_m A_{\mathbf{h}}\delta_{m,l-h} = E(\mathbf{k})\sum_{\mathbf{h}} A_{\mathbf{h}}\delta_{h,l} \tag{2.12}$$

where $\delta_{h,l}$ are Kronecker delta functions. In other words,

$$\delta_{h,l} = \begin{cases} 1 & \text{if } h = l \\ 0 & \text{if } h \neq l \end{cases} \tag{2.13}$$

This can be simplified further since $h = l$ only happens once for the first term of the left-hand side, h times for the second term and once for the right-hand side,

$$\frac{\hbar^2}{2m}|\mathbf{k} + \mathbf{K_h}|^2 A_h + \sum_h V_m A_h = E(\mathbf{k}) A_h \tag{2.14}$$

(2.14) is an eigenvalue problem. All that is needed to find $E(\mathbf{k})$ is to construct the Hamiltonian and solve for the eigen values,

$$H_{i,j} = \frac{\hbar^2}{2m}|\mathbf{k} + \mathbf{K_h}|^2 \delta_{i,j} + V \tag{2.15}$$

where,

$$V = V_S \cos(K_m \cdot \tau) + i V_A \sin(K_m \cdot \tau) \tag{2.16}$$

and,

$$\tau = a(\frac{1}{8}, \frac{1}{8}, \frac{1}{8}) \tag{2.17}$$

(2.16) accounts for two atoms per lattice point where the maximum offset is τ. V_S are form factors and are measured experimentally. The actual code starts by creating two tables used during the actual calculation process. The first table contains all of the k values. The second table is the precalculated bottom half of the Hamiltonian matrix. The eigen solver library[24] only needs the lower half if the matrix is Hermitian. Since these values do not change for different k, they can be precalculated. Creating these tables reduces the execution time and simplifies the creation of the Hamiltonian matrix in the main execution loop.

Once these tables are created, the main execution loop is started. The diagonals for a given k are calculated and added to the the template H matrix. After the Hamiltonian is built, the eigen library finds the eigenvalues for a given k. It should be noted the the number of points used for k determines the resolution for the Brillouin Zone, while the number of K points determines the accuracy for the Energies of Eigen Values.

Table 2.1: Form factors of Si, Ge, Sn, GaP and GaAs at 300K from [11]

	a(Å)	V_{3S}	V_{8S}	V_{11S}	V_{3A}	V_{4A}	V_{11A}
Si	5.43	-0.21	0.04	0.08	0	0	0
Ge	5.56	-0.23	0.01	0.06	0	0	0
Sn	6.49	-0.20	0	0.04	0	0	0
GaP	5.44	-0.22	0.03	0.07	0.12	0.07	0.02
GaAs	5.64	-0.23	0.01	0.06	0.07	0.05	0.01
AlSb	6.13	-0.21	0.02	0.06	0.06	0.04	0.02
InP	5.86	-0.23	0.01	0.06	0.07	0.05	0.01
GaSb	6.12	-0.22	0.00	0.05	0.06	0.05	0.01
InAs	6.04	-0.22	0.00	0.05	0.08	0.05	0.03
IsSb	6.04	-0.20	0.00	0.04	0.06	0.05	0.01
ZnS	5.41	-0.22	0.03	0.07	0.24	0.14	0.04
ZnSe	5.65	-0.23	0.01	0.06	0.18	0.12	0.03
ZnTe	6.07	-0.22	0.00	0.05	0.13	0.10	0.01
CdTe	6.41	-0.20	0.00	0.04	0.15	0.09	0.04

2.3 Band Structure of Common Semiconductors

A calculation replicating the work of [11] was made to verify the validity of the code. Figures 2.1 through 2.14 show the results of program.

The average run time for the program was approximately 1 minute and 10 seconds. This includes calculating the eigen energies for 70, 124 $\times$ 124 Hamiltonian matrices or about 10 seconds per lattice point. For materials such as Si or Ge where V_A values are 0 the calculation reduces to about 1 second per lattice point. For both cases 98% of these times were spent in the eigen library calculating energies.

2.4 Finding Pseudopotentials

Ab initio techniques to find band structures can be used, but this can be both time and memory intensive. It is sometimes desirable to calculate band structure inside of a simulation for certain applications. Examples of this include studying hetero-structures with different doping gradients [13], Strained Latices [26] or studying the effects of surface roughness [22] in a device.

EPM is a powerful method for calculating band structure and studying the effects, and there

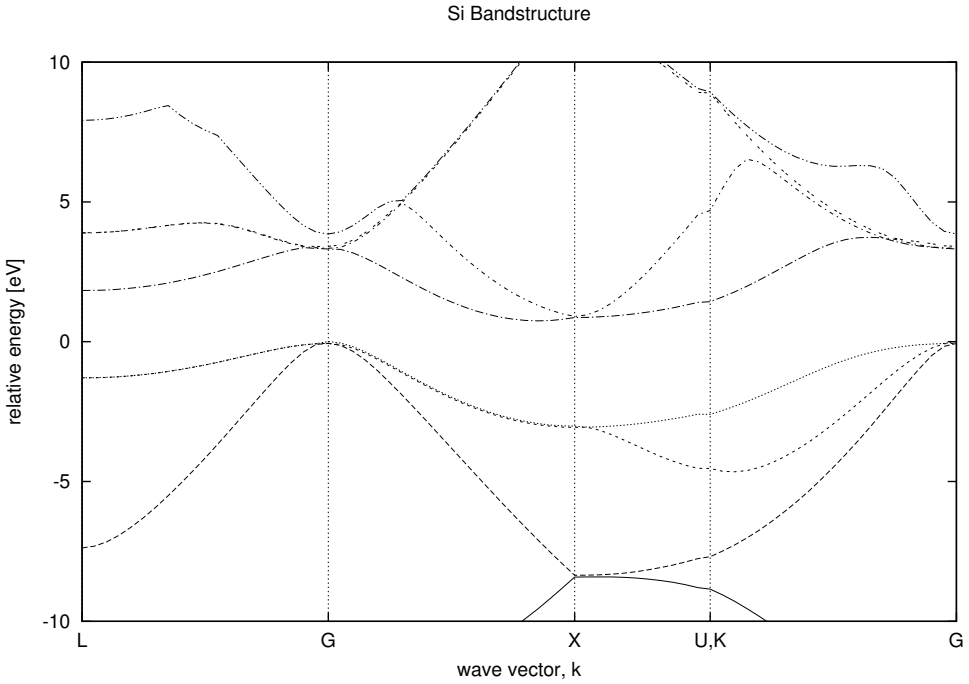


Figure 2.1: Si band structure with form factors from [11]

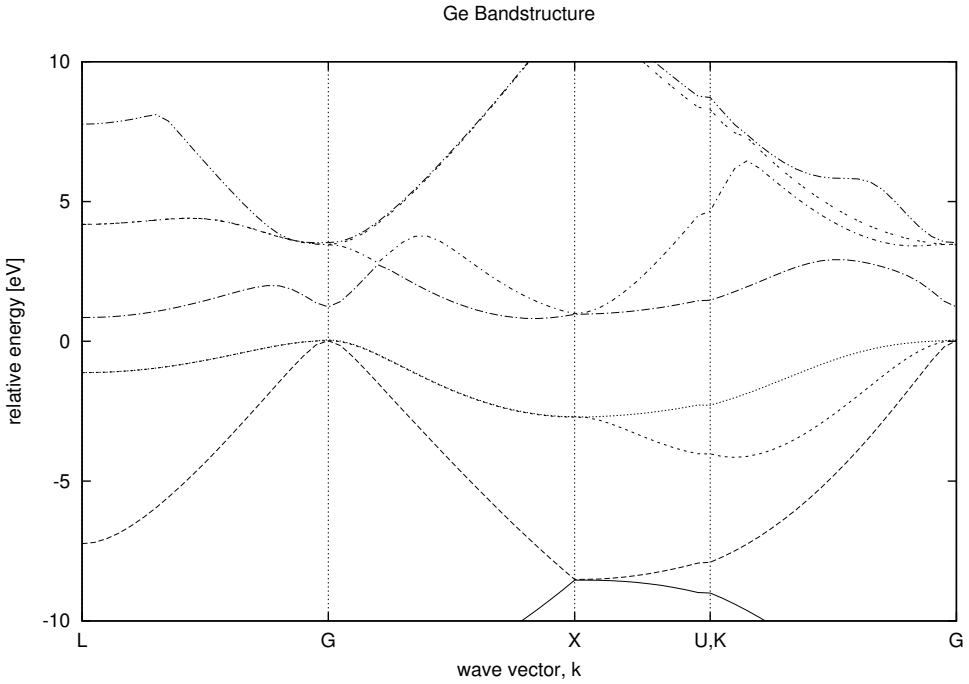


Figure 2.2: Ge band structure with form factors from [11]

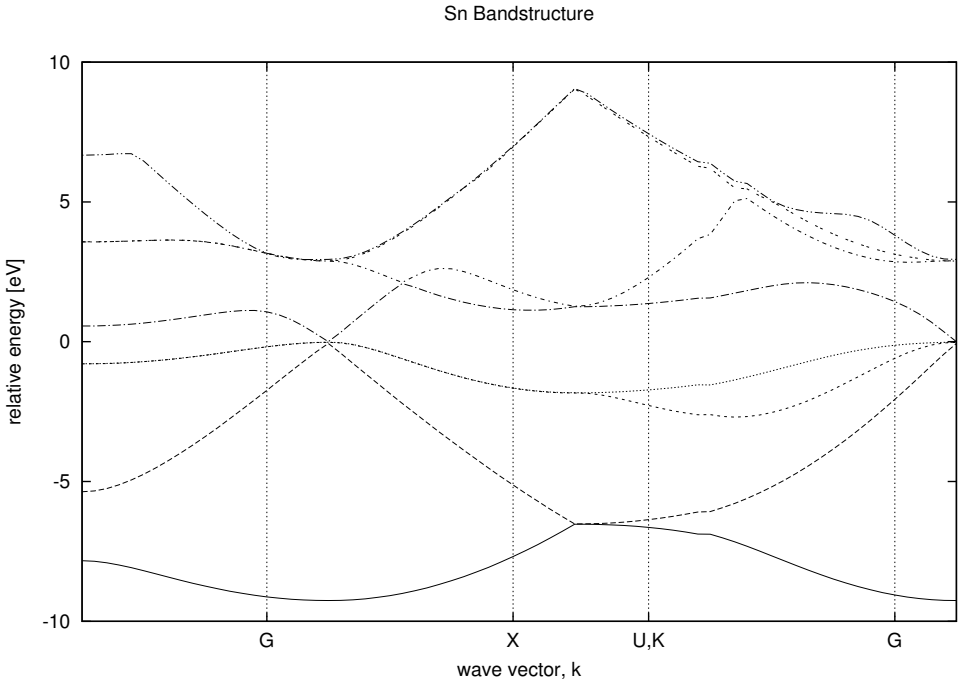


Figure 2.3: Sn band structure with form factors from [11]

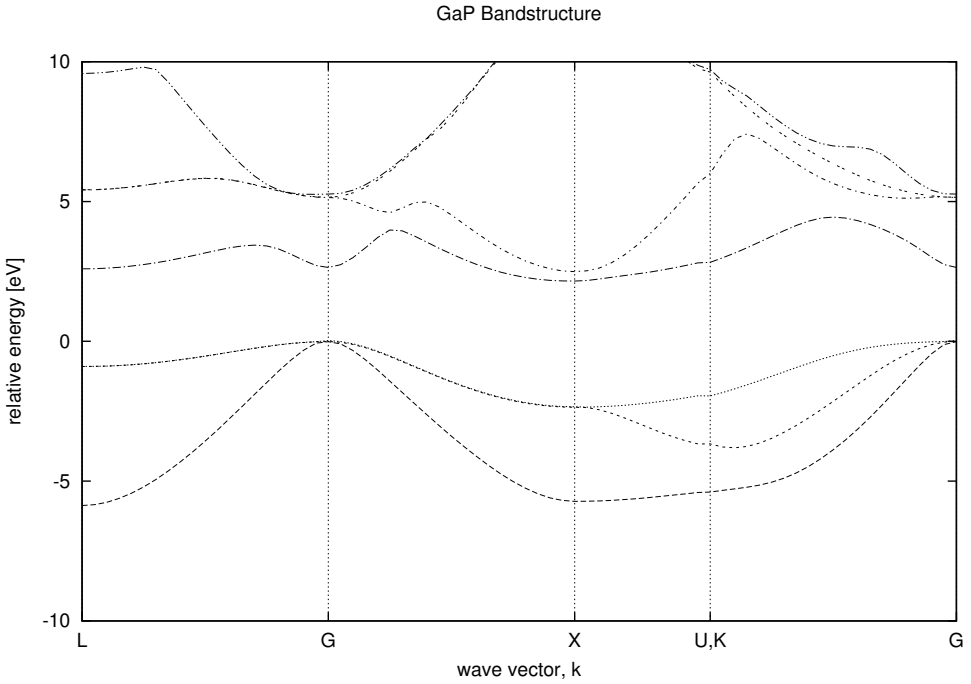


Figure 2.4: GaP band structure with form factors from [11]

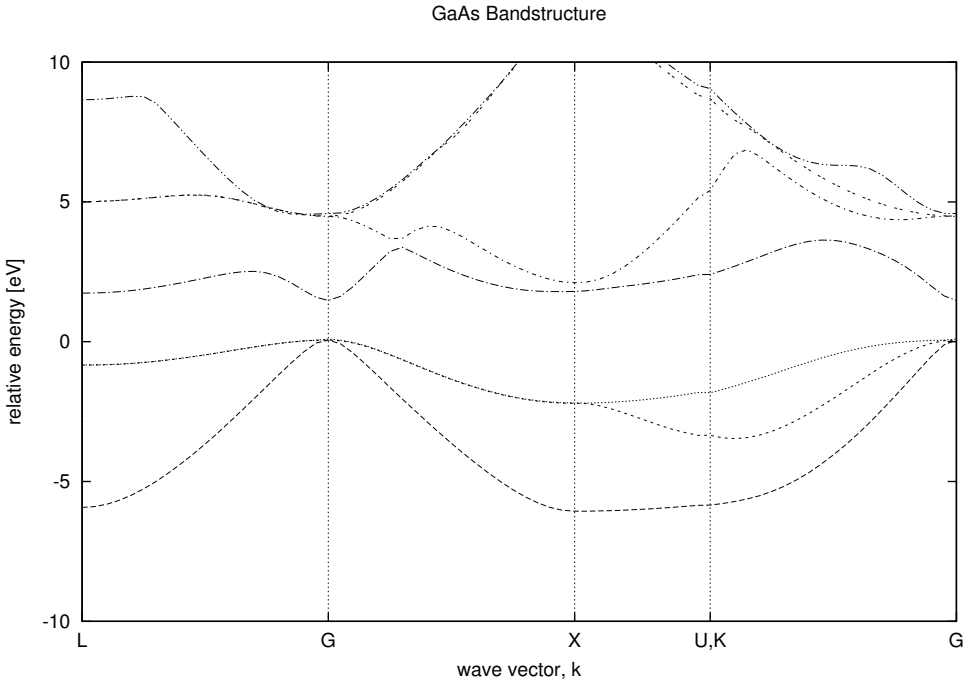


Figure 2.5: GaAs band structure with form factors from [11]

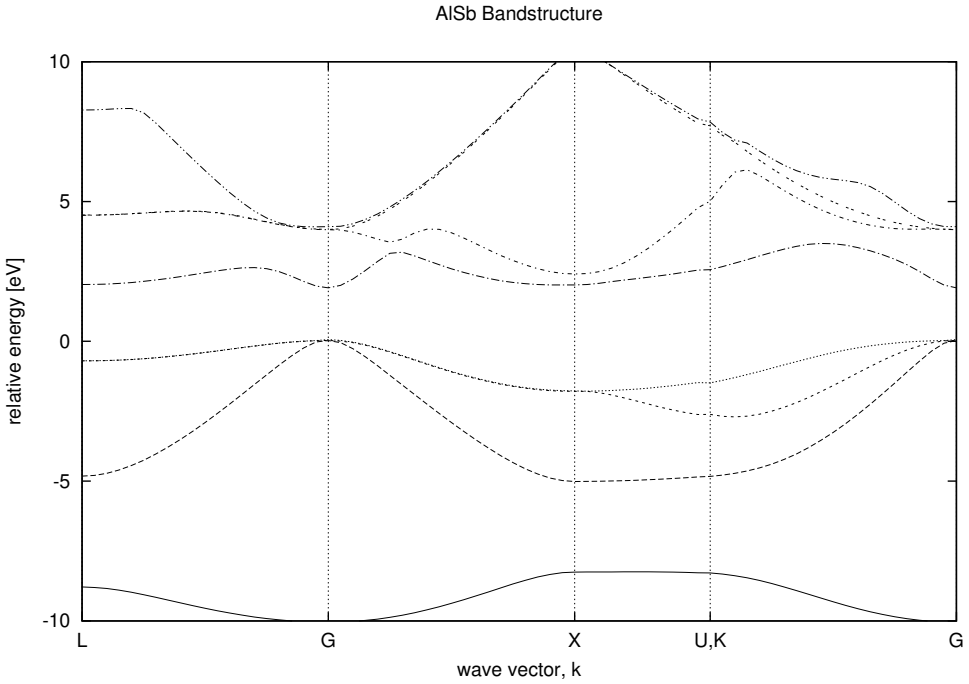


Figure 2.6: AlSb band structure with form factors from [11]

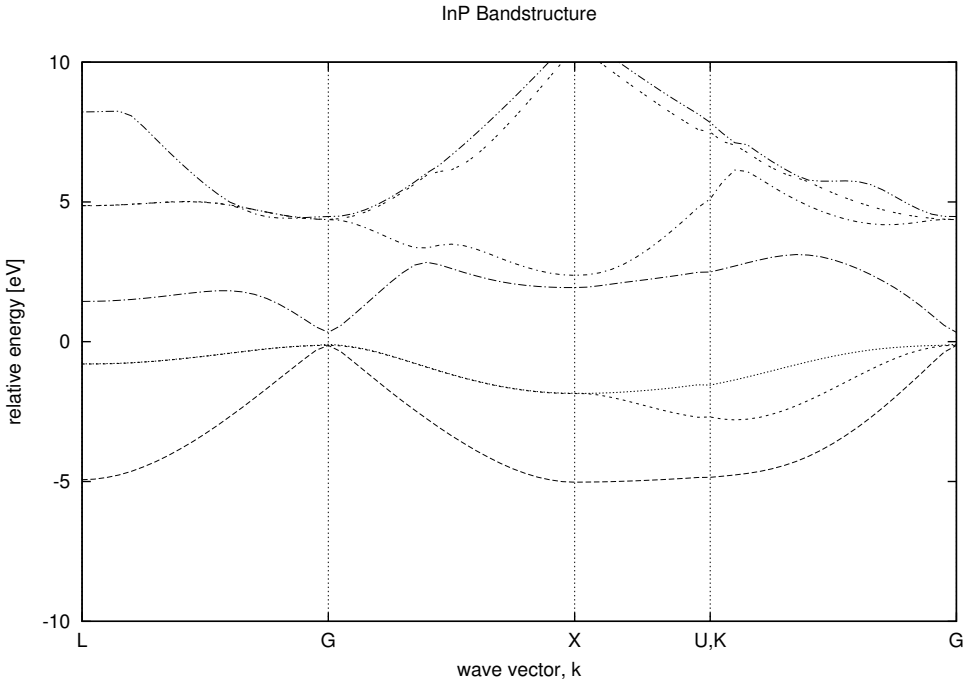


Figure 2.7: InP band structure with form factors from [11]

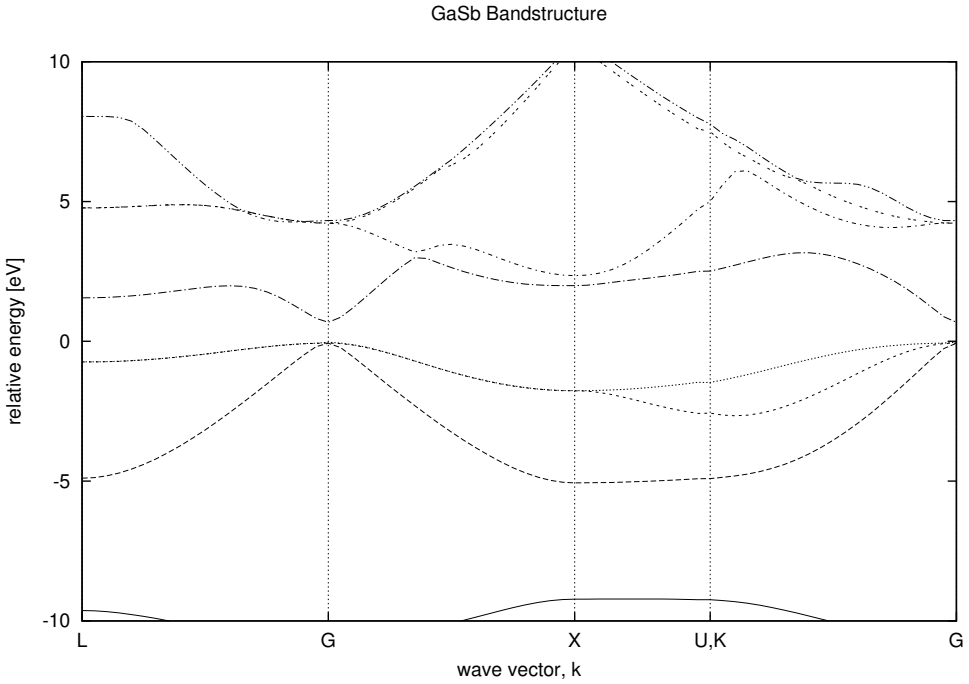


Figure 2.8: GaSb band structure with form factors from [11]

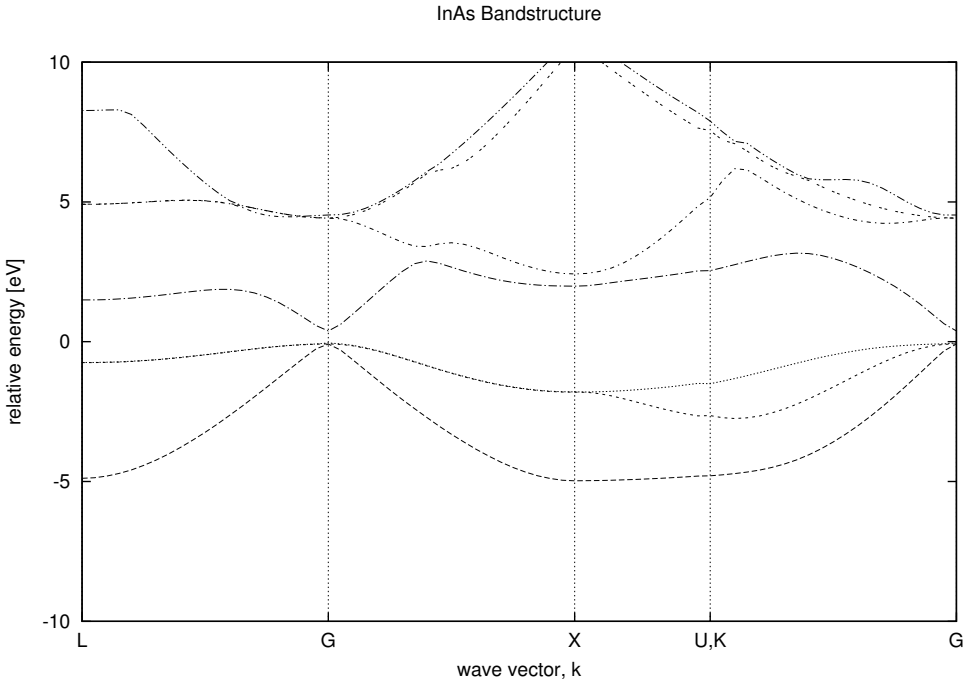


Figure 2.9: InAs band structure with form factors from [11]

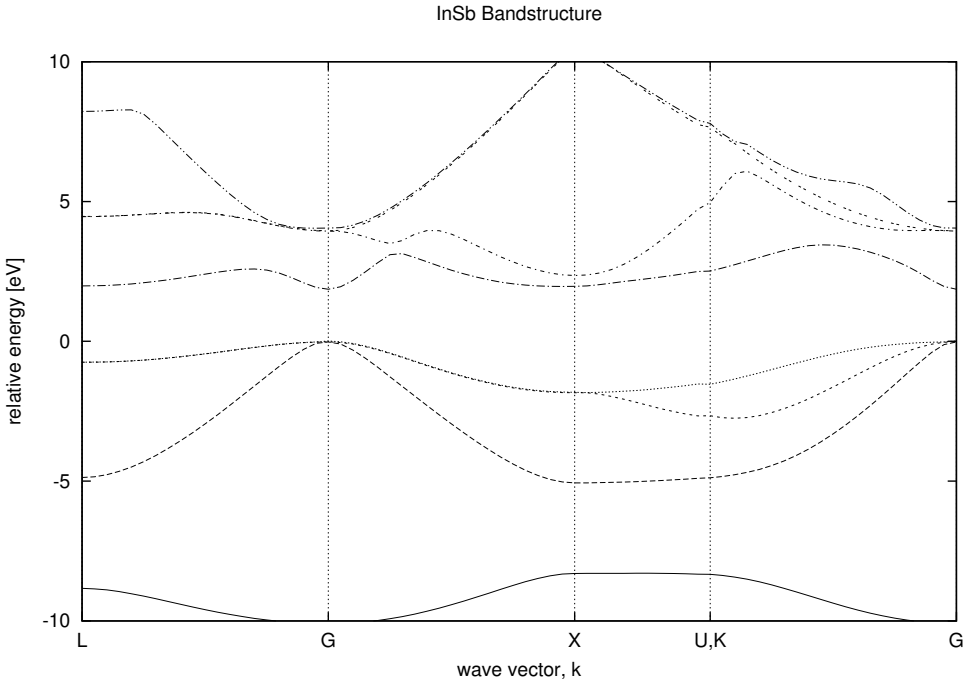


Figure 2.10: InSb band structure with form factors from [11]

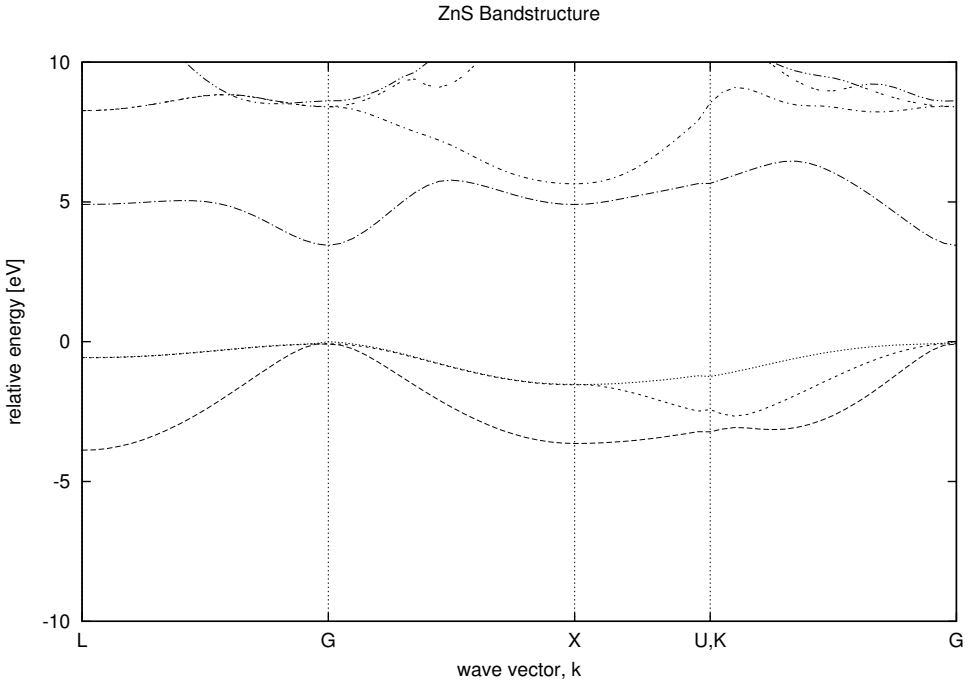


Figure 2.11: ZnS band structure with form factors from [11]

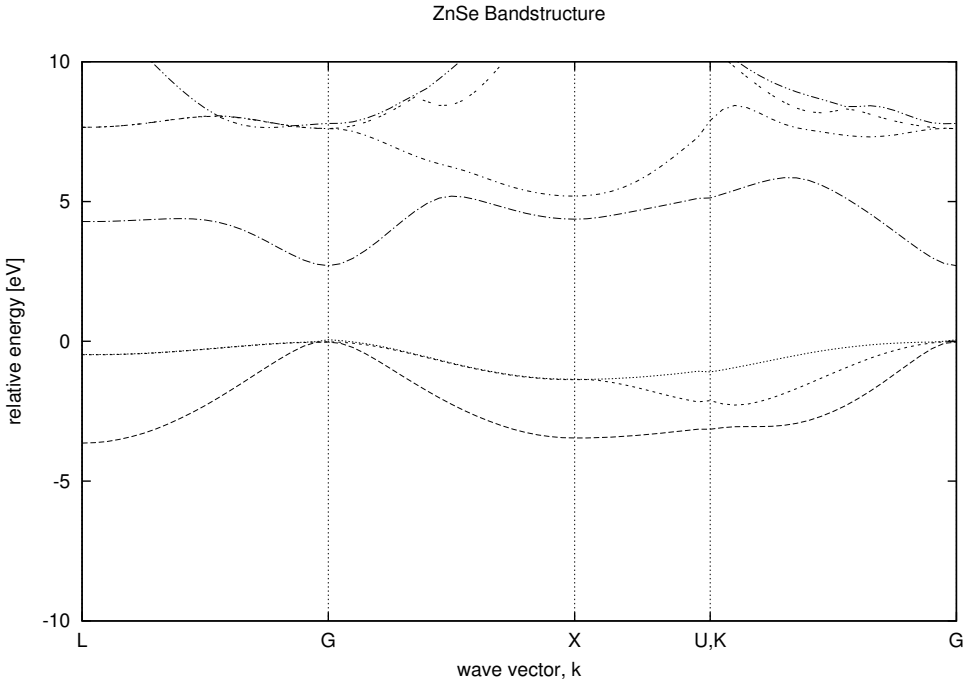


Figure 2.12: ZnSe band structure with form factors from [11]

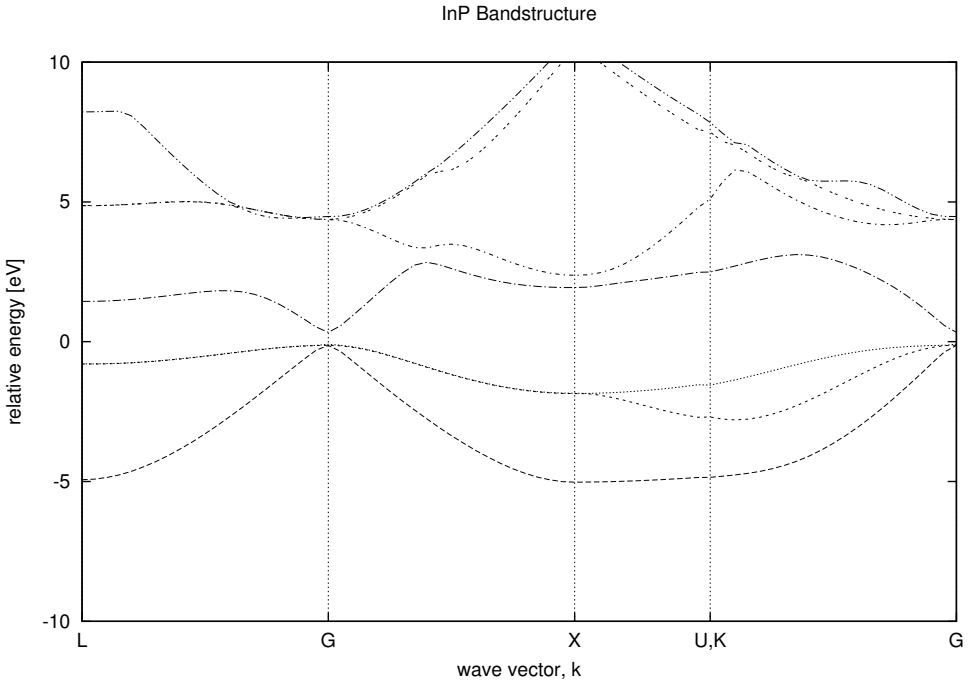


Figure 2.13: ZeTe band structure with form factors from [11]

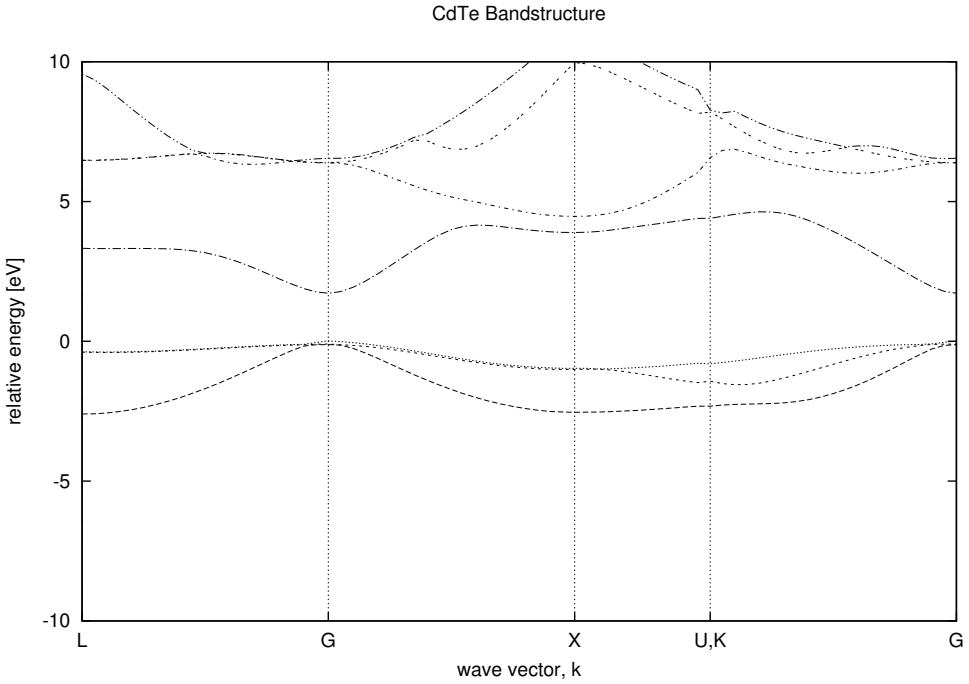


Figure 2.14: CdTe band structure with form factors from [11]

are several modern approaches to finding form factors for different applications. First-principles calculations such as LCAO can be used to find pseudopotentials. Curve fitting and Genetic algorithms can be employed to find fine grain values of band structure where data have already be acquired [34].

3. Applications

Utilizing exciting new materials such as graphene and device structures such as Silicon On Insulator (SOI) necessitates being being able predict their operation as actual devices. Ab initio approaches give insight into engineering effective geometries, but they are limited to structures in the range of thousands of atoms[20]. For any kind of large scale integration these methods fall short. EPM offers a solution for devices $\gtrsim 10^5$ atoms. There are many techniques for finding pseudopotentials that will work for different systems. Most of these methods stretch the definition of an "empirical" method. Ab initio calculations can be used to find pseudopotentials. Another procedure that will be shown below is to make a simple hybrid representation of the atomic structure in the system. Finally, stochastic methods for finding pseudopotentials have become increasingly popular. Indeed this is no longer empirical, but shows the versatility of EPM for modeling modern materials. Remarkably, pseudopotentials for armchair nanoribbons of graphene have been used to calculate band structure[31] and simulate the properties of devices.

There are situations where it becomes desirable to do band calculations inside device simulations where the band structure can change as a result of the operation of the device. This is a good application for modeling using first principles to discover pseudopotentials that could work for using EPM in the actual device.

Once band structure is known (either by ab initio or EPM) it can be mapped into effective mass and/or mobility to lessen the computational burden in actual device simulation. Work has been carried out to calculate mobility in strained materials by finding the dilation deformation potentials from unstrained materials, and then applying the strained band structure[23] to recalculate the mobility. Another method is to use Full Band Monte Carlo (FBMC) to calculate the mobility for specific common geometries[45].

3.1 Alloy Materials

First principles techniques are able to make calculations based on specific atom locations. This is of great use in studying boundary deformations. For some alloy materials there is a known concentration, and there may be a known strain. For example SiGe with highly controlled concentrations can be epitaxially grown through the use of ultrahigh-vacuum chemical vapor deposition[33]. There may even be a gradient concentration in the material. Because these materials are quasi-homogeneous virtual crystal approximation (VCA) allows the properties of two atoms to be used to make a hybrid atom[26]. For this method the strained lattice and the form factors are a function of the concentration percentage. The strained lattice can be represented as:

$$a'(x) = a_1(1-x) + a_2x \tag{3.1}$$

Similarly the form factors can be calculated as:

$$V'(x) = V_1(1-x) + V_2x \tag{3.2}$$

Table 3.1: Form factors of SiGe Alloy for different concentrations at 300K

x	a(Å)	V_{3S}	V_{8S}	V_{11S}	V_{3A}	V_{4A}	V_{11A}
0.00	5.4300	-0.210	0.0400	0.080	0	0	0
0.25	5.4625	-0.215	0.0325	0.075	0	0	0
0.50	5.4950	-0.220	0.0250	0.070	0	0	0
0.75	5.5275	-0.225	0.0175	0.065	0	0	0
1.00	5.5600	-0.230	0.0100	0.060	0	0	0

Table 3.1 shows the parameters of SiGe alloy at differing concentrations, and figure 3.1 shows progression of how the band structure changes. The method for calculating the band structure is the same as above. It can be seen that the top band in figure 3.1(a) starts to pull down in 3.1(b), and by 3.1(c) at a concentration of $x = 50\%$ that same band is below the normal conduction band of Silicon. At $x = 75\%$ the alloy becomes Germanium-like as the direct band gap begins to takeover.

3.2 Genetic Algorithms

Genetic algorithms have become highly utilized in many aspects of semiconductor devices. They have been used to find device parameters[19], in band structure engineering[30], in direct band gap discovery[14], in reconstruction of Scanning Tunneling Microscope (STM) imagery[10] and even in optimization of manufacturing devices[50]. They have been used to because of their ability to hone into special systems by evolving from similar parent states. A detailed overview of genetic algorithms is given in [12]. Here is the abbreviated description of a genetic algorithm:

(1) **Start.** An initial population is made. This can be either a random population or a population that already has desired traits.

(2) **Breeding.** Breeding takes place in three steps:

   (a) **Selection.** Parents are selected at random and weighted by their fitness and Copies.

   (b) **Crossover.** Children exchange chromosomes.

   (c) **Mutation.** Each chromosome has the chance to mutate by some probability.

(3) **Repeat.** Breed until a desired trait is optimized to the selected quantity or by a predetermined number of generations.

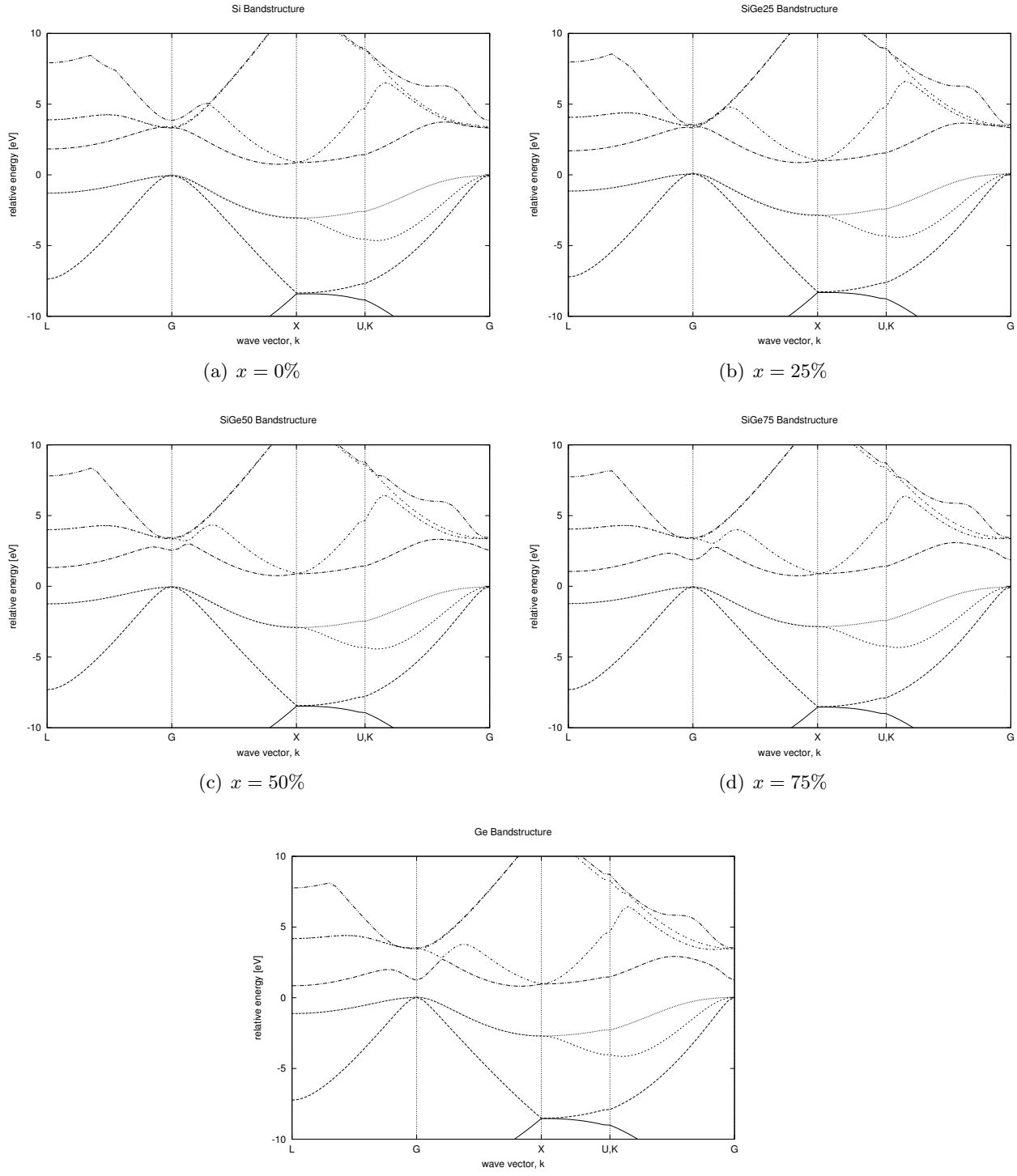
There are many variations on this system. One of its main advantages is that many generations can be done in parallel.

Recently, genetic algorithms applied to EPM has been used to calculate the band structure for 4H-SiC [34].

3.3 Future Work

Future work will be focused on applying the band structures to actual device simulation. A framework for this purpose needs to be developed. Spin-orbit coupling is necessary for effectively simulating holes in many materials such a germanium and is an easy addition to the algorithm.

A density of states calculation is also necessary. Using a genetic algorithm find different devices' properties and simulate operation is the end goal.



(a) $x = 0\%$

(b) $x = 25\%$

(c) $x = 50\%$

(d) $x = 75\%$

(e) $x = 100\%$

Figure 3.1: The Band Structure for Different Concentrations of Silicon-Germanium Alloy

BIBLIOGRAPHY

[1] Quantumwise atomistix toolkit (atk).

[2] Wanda Andreoni and R. Car. Similarity of (ga, al, as) alloys and ultrathin heterostructures: Electronic properties from the empirical pseudopotential method. Phys. Rev. B, 21:3334–3344, Apr 1980.

[3] Gabriel Bester. Electronic excitations in nanostructures: an empirical pseudopotential based approach. Journal of Physics: Condensed Matter, 21(2):023202, 2009.

[4] S. Bloom and T.K. Bergstresser. Band structure of î±-sn, insb and cdte including spin-orbit effects. Solid State Communications, 6(7):465 – 467, 1968.

[5] P Bowlan, E Martinez-Moreno, K Reimann, M Woerner, and T Elsaesser. Terahertz radiative coupling and damping in multilayer graphene. New Journal of Physics, 16(1):013027, 2014.

[6] Mads Brandbyge, José-Luis Mozos, Pablo Ordejón, Jeremy Taylor, and Kurt Stokbro. Density-functional method for nonequilibrium electron transport. Phys. Rev. B, 65:165401, Mar 2002.

[7] J. Chelikowsky, D. J. Chadi, and Marvin L. Cohen. Calculated valence-band densities of states and photoemission spectra of diamond and zinc-blende semiconductors. Phys. Rev. B, 8:2786–2794, Sep 1973.

[8] James R. Chelikowsky and Marvin L. Cohen. Electronic structure of silicon. Phys. Rev. B, 10:5095–5107, Dec 1974.

[9] James R. Chelikowsky and Marvin L. Cohen. Nonlocal pseudopotential calculations for the electronic structure of eleven diamond and zinc-blende semiconductors. Phys. Rev. B, 14:556–582, Jul 1976.

[10] F.C. Chuang, C.V. Ciobanu, V.B. Shenoy, C.Z. Wang, and K.M. Ho. Finding the reconstructions of semiconductor surfaces via a genetic algorithm. Surface Science, 573(2):L375 – L381, 2004.

[11] Marvin L. Cohen and T. K. Bergstresser. Band structures and pseudopotential form factors for fourteen semiconductors of the diamond and zinc-blende structures. Phys. Rev., 141:789–796, Jan 1966.

[12] E.A.B. Cole. Overview of device modelling. In Mathematical and Numerical Modelling of Heterostructure Semiconductor Devices: From Theory to Programming, pages 3–20. Springer London, 2009.

[13] Richard A. Coles. Theory of the Electronic States of Semiconductor Heterostructures. PhD thesis, Durham University, March 1999.

[14] Mayeul d'Avezac, Jun-Wei Luo, Thomas Chanier, and Alex Zunger. Genetic-algorithm discovery of a direct-gap and optically allowed superstructure from indirect-gap si and ge semiconductors. Phys. Rev. Lett., 108:027401, Jan 2012.

[15] A. De and Craig E. Pryor. Predicted band structures of iii-v semiconductors in the wurtzite phase. Phys. Rev. B, 81:155210, Apr 2010.

[16] Stephen M. Goodnick Dragica Vasileska, editor. Nano-Electronic Devices: Semiclassical and Quantum Transport Modeling. Springer Science+Business Media, 2011.

[17] N. D. Drummond, V. Zólyomi, and V. I. Fal'ko. Electrically tunable band gap in silicene. Phys. Rev. B, 85:075423, Feb 2012.

[18] Aniello Esposito. Band Structure Effects and Quantum Transport. PhD thesis, ETH ZURICH, 2010.

[19] Fang Feng-bo and Wu Tao. Genetic algorithm and semiconductor device model parameter extraction. In Genetic and Evolutionary Computing, 2009. WGEC '09. 3rd International Conference on, pages 97–100, Oct 2009.

[20] Massimo V Fischetti, Jiseok Kim, Sudarshan Narayanan, Zhun-Yong Ong, Catherine Sachs, David K Ferry, and Shela J Aboud. Pseudopotential-based studies of electron transport in graphene and graphene nanoribbons. Journal of Physics: Condensed Matter, 25(47):473202, 2013.

[21] Massimo V. Fischetti and Steven E. Laux. Monte carlo analysis of electron transport in small semiconductor devices including band-structure and space-charge effects. Phys. Rev. B, 38:9721–9745, Nov 1988.

[22] Massimo V. Fischetti and Sudarshan Narayanan. An empirical pseudopotential approach to surface and line-edge roughness scattering in nanostructures: Application to si thin films and nanowires and to graphene nanoribbons. Journal of Applied Physics, 110(8):–, 2011.

[23] M. V. Fischettia and S. E. Laux. Band structure, deformation potentials, and carrier mobility in strained Si, Ge, and SiGe alloys, volume 80. J. Appl. Phys., 1996.

[24] et al Gel Guennebaud, Benoit Jacob. Eigen v3, 2010.

[25] Salvador Gonzalez. Empirical pseudopotential method for the band structure calculation of strauned-slilicon germanium materials. Master's thesis, Arizona State University, 2001.

[26] Salvador Gonzalez, Dragica Vasileska, and AlexanderA. Demkov. Empirical pseudopotential method for the band structure calculation of strained-silicon germanium materials. Journal of Computational Electronics, 1(1-2):179–183, 2002.

[27] Suyog Gupta, Blanka Magyari-KÃpe, Yoshio Nishi, and Krishna C. Saraswat. Achieving direct band gap in germanium through integration of sn alloying and external strain. Journal of Applied Physics, 113(7), 2013.

[28] Conyers Herring. A new method for calculating wave functions in crystals. Phys. Rev., 57:1169–1177, Jun 1940.

[29] Jiseok Kim and Massimo V. Fischetti. Empirical pseudopotential calculations of the band structure and ballistic conductance of strained [001], [110], and [111] silicon nanowires. Journal of Applied Physics, 110(3):–, 2011.

[30] Kwiseon Kim, Peter A. Graf, and Wesley B. Jones. A genetic algorithm based inverse band structure method for semiconductor alloys. Journal of Computational Physics, 208(2):735 – 760, 2005.

[31] Yoshiyuki Kurokawa, Shintaro Nomura, Tadashi Takemori, and Yoshinobu Aoyagi. Large-scale calculation of optical dielectric functions of diamond nanocrystallites. Phys. Rev. B, 61:12616–12619, May 2000.

[32] Rita Magri. Pseudopotentials for band structure calculations. TMCSIII, January 2012.

[33] Bernard S. Meyerson, Kevin J. Uram, and Francoise K. LeGoues. Cooperative growth phenomena in silicon/germanium lowtemperature epitaxy. Applied Physics Letters, 53(25):2555–2557, 1988.

[34] G. Ng, D. Vasileska, and D.K. Schroder. Empirical pseudopotential band structure parameters of 4h-sic using a genetic algorithm fitting routine. Superlattices and Microstructures, 49(1):109 – 115, 2011.

[35] M. Ali Omar. Elementry Solid State Physis. Addison-Wesley Publishing Company, Inck, 1993.

[36] R. H. Parmenter. Symmetry properties of the energy bands of the zinc blende structure. Phys. Rev., 100:573–579, Oct 1955.

[37] James C. Phillips and Leonard Kleinman. New method for calculating wave functions in crystals and molecules. Phys. Rev., 116:287–294, Oct 1959.

[38] Jeff Racine. gnuplot 4.0: A portable interactive plotting utility. Journal of Applied Econometrics, 21:133–141, January/February 2006.

[39] I. N. Remediakis and Efthimios Kaxiras. Band-structure calculations for semiconductors within generalized-density-functional theory. Phys. Rev. B, 59:5536–5543, Feb 1999.

[40] Martin M. Rieger and P. Vogl. Electronic-band parameters in strained substrates. Phys. Rev. B, 48:14276–14287, Nov 1993.

[41] Mark Silver. Application of the Pseudopotential Method to the Theory of Serniconductors. PhD thesis, University of Surrey, 1991.

[42] J. C. Slater. An augmented plane wave method for the periodic potential problem. Phys. Rev., 92:603–608, Nov 1953.

[43] D. L. Smith and C. Mailhiot. Theory of semiconductor superlattice electronic structure. Rev. Mod. Phys., 62:173–234, Jan 1990.

[44] A. Svane. Hartree-fock band-structure calculations with the linear muffin-tin-orbital method: Application to c, si, ge, and ¡i¿¡/i¿-sn. Phys. Rev. B, 35:5496–5502, Apr 1987.

[45] Enzo Ungersboeck, Siddhartha Dhar, Gerhard Karlowatz, Viktor Sverdlov, Hans Kosina, and Siegfried Selberherr. The Effect of General Strain on the Band Structure and Electron Mobility of Silicon, volume 54. IEEE Transactions on Electron Devices, Sep 2007.

[46] David Vanderbilt. Theory of pseudopotentials. Technical report, July 2006.

[47] Dragica Vasileska. Empirical pseudopotential method.

[48] Dragica Vasileska. Tutorial for semi-empirical band-structure calculation.

[49] P. Vogl, Harold P. Hjalmarson, and John D. Dow. A semi-empirical tight-binding theory of the electronic structure of semiconductorsâ. Journal of Physics and Chemistry of Solids, 44(5):365 – 378, 1983.

[50] Cheng-Shuo Wang and Reha Uzsoy. A genetic algorithm to minimize maximum lateness on a batch processing machine. Computers and Operations Research, 29(12):1621 – 1640, 2002.

[51] Y.W. Wen, H.J. Liu, L. Pan, X.J. Tan, H.Y. Lv, J. Shi, and X.F. Tang. Reducing the thermal conductivity of silicon by nanostructure patterning. Applied Physics A, 110(1):93–98, 2013.

A. Acronyms

BZ Brillouin Zone 6

DFT Density Functional Theory 1

EPM Empirical Pseudopotential Method

FBMC Full Band Monte Carlo 17

LCAO Linear Combination of Atomic Orbitals 1

OPW Orthogonalized Plane Wave 5

SOI Silicon On Insulator 17

STM Scanning Tunneling Microscope 19

TISE Time Independent Schrodinger Equation 6

VCA virtual crystal approximation 18

B. Code Listing and Examples

It is necessary to have the Eigen Library[24] to compile and run the simulator. To compile (using the gnu compiler) and run a simulation type (from a UNIX shell):

```
g++ -o epm epm.cpp
./epm < Si.epm > Si.dat
```

The Si.epm file is a parameter file used to deliver the lattice constant, form factors and the lattice traversal info to the program. It is a very roughly parsed into the simulator.

resources/Si.epm

```
LC 5.431e-10 ; lattice constant
V  3 -0.21 0 ; Vs3 Va3
V  8  0.04 0
V  11 0.08 0
L 20 G 20 X 10 U 0 K 20 G ; crystal points to traverse
```

"LC" is a keyword to tell the program that the lattice constant follows after some amount of white space. "V" tells it that a form factor follows. If L, G, X, U, K or G starts a line, the parser the will build the k vector and will calculate the number of points specified inside the Brillouin zone. If 0 is used, the program will still calculate one point from the reciprocal lattice.

B.1 EPM Band Structure in C++

resources/epm.cpp

```
#include <stdio.h>
#include <math.h>
#include <iostream>
#include "Eigen/Dense"
```

```
5
using namespace std;
using namespace Eigen;

// constants
#define m0     9.109e-31  // free electron mass
#define q      1.602e-19  // carge of an electron
#define hbar   1.054e-34  // Plank's constant
#define hbareV 6.581e-16
#define pi     3.1415926535898
15
#define nN 124

double dot(double *A, double *B) {
    return A[0]*B[0] + A[1]*B[1] + A[2]*B[2];
20 }

int main(int argc, char *argv[]) {

   double T[3] = { 0.125, 0.125, 0.125 };      // structure factor
25 int Nn = 124;
   double k[Nn][3];                            // k vectors
   double K[nN*2+1][3];                        // lookup table for K vectors
   double H[nN][nN];                           // Hamaltonian
   complex <double> V;                                  // psudopotential
30 double i,j,l[3];                            // generic counters
   int    dd,d,e,f,g,h;
   double *p;
   typedef Matrix<double, 124, 124> Matrix124d;
   typedef Matrix<complex<double>, 124, 124> Matrix124cd;
35 Matrix124cd HH;

   char line[256];
   char *tok;
   double lc;                // lattice constant
40 double C;                 // constant used for building H
   int    ppc = 0;           // to keep track of the number of pseudopotentials
   int    pp[20];            // 20 pseudopotentials max
```

```
double Vs[20];
double Va[20];

// get parameters: lattice constant, pseudopotentials and lattice traversal
Nn = 0;
while(cin.getline(line,256)) {
   tok = strtok(line,"␣");
   if(!tok) break;

   // get lattice constant
   if(!strcmp(tok,"LC")) {
      lc = atof(strtok(NULL,"␣"));
      C  = hbar*hbar*2*pi*pi/(lc*lc*m0*q);
   }

   // get pseudopotential values
   if(!strcmp(tok,"V") && ppc<20) {
      pp[ppc]   = atof(strtok(NULL,"␣"));
      Vs[ppc]   = atof(strtok(NULL,"␣"));
      Va[ppc++] = atof(strtok(NULL,"␣"));
   }

   // get and generate k matrix
   if(!strcmp(tok,"L")) { k[0][0]= .5; k[0][1]= .5; k[0][2]= .5; Nn=1; }
   if(!strcmp(tok,"G")) { k[0][0]=  0; k[0][1]=  0; k[0][2]=  0; Nn=1; }
   if(!strcmp(tok,"X")) { k[0][0]=  1; k[0][1]=  0; k[0][2]=  0; Nn=1; }
   if(!strcmp(tok,"W")) { k[0][0]=  1; k[0][1]= .5; k[0][2]=  0; Nn=1; }
   if(!strcmp(tok,"K")) { k[0][0]=.75; k[0][1]=.75; k[0][2]=  0; Nn=1; }
   if(!strcmp(tok,"U")) { k[0][0]=  1; k[0][1]=.25; k[0][2]=.25; Nn=1; }
   if(Nn) {
      Nn=0;
      tok = strtok(NULL,"␣");
      while(isdigit(*tok)) {
         h = atoi(tok);
         tok = strtok(NULL,"␣");
         if(h==0) h=1;
         if(!strcmp(tok,"L")) { l[0]= (.5-k[Nn][0])/h; l[1]= (.5-k[Nn][1])/h; l[2]=  (.5-
             k[Nn][2])/h; }
```

```
            if(!strcmp(tok,"G")) { l[0]=  (0-k[Nn][0])/h; l[1]=  (0-k[Nn][1])/h; l[2]=   (0-
                k[Nn][2])/h; }
            if(!strcmp(tok,"X")) { l[0]=  (1-k[Nn][0])/h; l[1]=  (0-k[Nn][1])/h; l[2]=   (0-
                k[Nn][2])/h; }
            if(!strcmp(tok,"W")) { l[0]=  (1-k[Nn][0])/h; l[1]= (.5-k[Nn][1])/h; l[2]=   (0-
                k[Nn][2])/h; }
            if(!strcmp(tok,"K")) { l[0]=(.75-k[Nn][0])/h; l[1]=(.75-k[Nn][1])/h; l[2]=   (0-
                k[Nn][2])/h; }
            if(!strcmp(tok,"U")) { l[0]=  (1-k[Nn][0])/h; l[1]=(.25-k[Nn][1])/h; l[2]= (.25-
                k[Nn][2])/h; }
            if(-1e-15<l[0] && l[0]<1e-15) l[0] = 0;
            if(-1e-15<l[1] && l[1]<1e-15) l[1] = 0;
            if(-1e-15<l[2] && l[2]<1e-15) l[2] = 0;
            while(h-- && (Nn<124)) {
               Nn++;
               k[Nn][0] = k[Nn-1][0] + l[0];
               k[Nn][1] = k[Nn-1][1] + l[1];
               k[Nn][2] = k[Nn-1][2] + l[2];
               if(-1e-15<k[Nn][0] && k[Nn][0]<1e-15) k[Nn][0] = 0;
               if(-1e-15<k[Nn][1] && k[Nn][1]<1e-15) k[Nn][1] = 0;
               if(-1e-15<k[Nn][2] && k[Nn][2]<1e-15) k[Nn][2] = 0;
            }
            tok = strtok(NULL,"␣");
         }
         Nn++;
      }
   }

   // verify pseudopotentials
   //f=ppc; while(f--) cout << pp[f] << " " << Vs[f] << " " << Va[f] << endl;
   // verify k vector generation
//   for(h=0; h<Nn; h++) printf("%i %g %g %g\n", h, k[h][0], k[h][1], k[h][2]);
//   cout << endl;

   // build a look up table for building K vector
   // caution! this section must be modified if nN is modified
   // needs to be updated for variable nN
   f=-5;
```

```
    g=0;
    h=1;
    p=&K[0][0];
    while(1) {
       *p++ = -f+g+h;
       *p++ =  f-g+h;
       *p++ =  f+g-h;

       if(f==5 && g==0 && h==-1) break;
       if(++h>2) {
          h=-2;
          if(++g>2) {
             g=-2;
             f++;
          }
       }
    }
    // verify K vector generation
    //printf("\n"); for(h=0; h<249; h++) printf("%i %g %g %g\n", h-124, K[h][0], K[h][1], K[h
        ][2]);

    // generate Hamiltonian with out diagonals
    for(f=0; f<nN ; f++) for(g=0; g<f; g++) {
       // calculate pseudopotential
       h = f+124-g;

       // search pseudopotential table
       V = 0;
       d = ppc;
       e = (int)dot(&K[h][0],&K[h][0]);
       while(d--) if(e == pp[d]) {
//             V = Vs[d]*13.6059;
             V.real(Vs[d]*13.6059);
             V.imag(Va[d]*13.6059);
             break;
          }
//      V = V*cos(2*pi*dot(&K[h][0],&T[0]));
       V.real(real(V)*cos(2*pi*dot(&K[h][0],&T[0])));
```

```
150         V.imag(imag(V)*sin(2*pi*dot(&K[h][0],&T[0])));
            HH(f,g) = V;
        }

        // create diagonals for each k space and find energies!
155     for(f=0; f<Nn; f++) {
            // calculate diagonals
            for(g=0; g<nN; g++) {
                h = g + 62;
                l[0]    = k[f][0] + K[h][0];
160             l[1]    = k[f][1] + K[h][1];
                l[2]    = k[f][2] + K[h][2];
    //            H[g][g] = C*dot(&l[0],&l[0]);
                HH(g,g) = C*dot(&l[0],&l[0]);
            }
165         SelfAdjointEigenSolver<Matrix124cd> eigensolver(HH);
            cout << eigensolver.eigenvalues().transpose() << endl;
        }

        // varify full Hamiltonian
170     //for(f=0; f<nN; f++) for(g=0; g<f+1; g++) cout << " " << f << " " << g <<  " " << HH.
            block(f,g,1,1) << " " << endl;
    }
```

B.2 Gnuplot Example

[38] was used to generate the plots in this paper. Which can be done by:

```
gnuplot Si.gp
```

resources/Si.gp

```
set key off
set title "Si␣Bandstructure\n"
set xlabel "wave␣vector,␣k"
4 set ylabel "relative␣energy␣[eV]"
```

```
unset x2tics
set grid x
set yr [-10:10]
set xtics ("L" 0 , "G" 20, "X" 40, "U,K" 51, "G" 71)
plot   "Si.dat" using ($1-10.54) with lines
replot "Si.dat" using ($2-10.54) with lines
replot "Si.dat" using ($3-10.54) with lines
replot "Si.dat" using ($4-10.54) with lines
replot "Si.dat" using ($5-10.54) with lines
replot "Si.dat" using ($6-10.54) with lines
replot "Si.dat" using ($7-10.54) with lines
replot "Si.dat" using ($8-10.54) with lines
pause -1
set term postscript eps enhanced
set output "Si.eps"

plot "Si.dat" using ($1-10.54) with lines, "Si.dat" using ($2-10.54) with lines, "Si.dat"
    using ($3-10.54) with lines, "Si.dat" using ($4-10.54) with lines, "Si.dat" using ($5
    -10.54) with lines, "Si.dat" using ($6-10.54) with lines, "Si.dat" using ($7-10.54) with
     lines, "Si.dat" using ($8-10.54) with lines
```