Ab initio calculations of phonon properties and spectra in condensed matter

Shauna M. Story

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2015

Reading Committee:

John J. Rehr, Chair

David H. Cobden

Andreas Karch

Program Authorized to Offer Degree:
UW Physics

University of Washington

Abstract

Ab initio calculations of phonon properties and spectra in condensed matter

Shauna M. Story

Chair of the Supervisory Committee:
Chair John J. Rehr
Physics

Phonons, the quantization of atomic vibrations, are important in studying many solid state properties, ranging from Raman, infrared, and neutron scattering to thermal expansion, specific heat, and heat conductivity to electrical resistivity and superconductivity. Generally, modeling the interatomic forces and vibrational modes of a given system require costly computer simulations, though once calculated, they provide the means to a wide variety of phonon properties. Our goal is to enable easy access to these phonon properties and to do this, we have developed a framework for easily automating the workflows involved in interfacing a phonon mode calculation with the analysis tools for determining such physical properties. This was originally implemented with the AI2PS (ab initio to phonon spectra) tool, meant solely for the calculation of vibrational properties. It has since greatly expanded in scope and capabilities to a general scientific workflow tool called CORVUS, which was started with the eventual goal of collecting all our various scientific workflow efforts—phonon properties, optical properties, and so on—into a single hub. We present here both the evolution of AI2PS into the CORVUS project and the phonon properties simulated, including Debye–Waller factors, phonon contributions the electron self–energy and spectral function, vibrational free energy, thermal expansion, and heat capacity.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

DFT: Density Functional Theory

DFPT: Density Functional Perturbation Theory

DOS: Density Of States

DM: Dynamical Matrix

DW: Debye-Waller

GGA: Generalized Gradient Approximation

LDA: Local Density Approximation

MA: Migdal Approximation

MS: Multiple–scattering

MSRD: Mean Square Relative Displacement

PDOS: Phonon (or Vibrational) Density of States

QP: Quasiparticle

RC: Retarded Cumulant

XAS: X-ray Absorption Spectra

XAFS: X-ray Absorption Fine Structure

XC: Exchange–correlation

VDOS: Vibrational (or Phonon) Density of States

ACKNOWLEDGMENTS

I would like to thank Professor John Rehr for the opportunity to work on a project that blends two of my greatest interests—physics and computer science—and for all his guidance throughout my doctorate. I would like to thank my colleagues Fernando Vila, Joshua Kas, and Kevin Jorissen for their indispensable help with my work. I would also like to thank Egor Klevak, Andrew Lee, Scott Hayashi, Siri Vimolchalao, Erik Lentz, Spencer Williams, Mitch Jones, Seyda Ipek, and John Fuini for their comradery and friendship. Finally, I am eternally grateful for the support of Catherine Provost throughout these five years.

DEDICATION

To Erreca Story

Chapter 1
INTRODUCTION

The driving force of theoretical physics is developing models that represent observed phenomena as accurately as possible. Generally, theorists will start with simplified models and gradually relax assumptions to incorporate more physical effects and improve agreement with observed phenomena. With the ubiquity of computers, this is now realized in the form of increasingly sophisticated computational software used to test the performance of our models numerically. Our overarching goal is to create scientific software tools that provide easy access to the oftentimes complex physical simulation of condensed matter systems.

1.1 Background

Three years after the discovery of the electron by J. J. Thomson in 1897, [1] Paul Drude introduced his model of electronic transport in metals. [2, 3] The Drude model entails a free electron gas moving in a stationary background of positive ions, with the classical collisions between them being the source of resistance. This simple kinetic model was relatively successful in explaining electronic and thermal conductivity in metals, the Hall effect, and the Wiedemann-Franz law (relates electrical conductivity and thermal conductivity), but failed significantly for electronic heat capacities.

Realizing the importance of the Pauli exclusion principle for the electron gas, Arnold Sommerfeld expanded on the Drude model by replacing the classical Maxwell-Boltzmann velocity distribution with the quantum Fermi-Dirac distribution. [4] The Drude-Sommerfeld model, typically referred to as the free electron model, improved agreement with experiment on a number of physical properties, including specific heat, thermal conductivity, and

thermopower. Some notable shortcomings were the failure of the Wiedemann–Franz law at moderate temperatures, the T^3 dependence of specific heat at low temperatures, and what determines the number of conduction electrons.

Crystalline materials were observed in 1912 by Max von Laue [5] to produce intense peaks in x–ray scattering experiments, leading William Lawrence Bragg and William Henry Bragg to propose in 1913 [6] that crystals were composed of parallel planes of ions forming a periodic lattice. In 1928, Felix Bloch encapsulated the effect of a static periodic ionic potential using Bloch waves. [7] The static lattice model brings improvements to properties like electronic band structure and chemical bonding, but still cannot accurately predict properties like low-temperature scaling of specific heat, thermal expansion, melting, electric and thermal conductance, and sound propagation.

At this point, it becomes imperative to include the effects of the motion of the ions. The lattice motion can be broken down into a combination of collective modes of vibration. These vibrational modes are called phonons, and they can lead to interesting effects in materials that we will present here.

1.2 Goal of Dissertation

This thesis aims to model various phonon effects in solid state materials, while building a simulation framework that can accommodate generically any scientific workflow. There are numerous current materials science software packages capable of calculating the phonon modes of a system, as well as numerous physical properties that can be simulated from these phonon modes. However, the process of obtaining the latter from the former often involves connecting multiple software packages. As such, our goal to model various phonon-related properties has necessitated the additional goal of creating a physical simulation workflow system that eases the process for our users. Both aspects will be presented in this thesis.

We will start by defining some common conventions and notation used throughout. In the next chapter, we will present a brief overview of the theory of phonons and related properties simulated in this work. In chapter 3, we present the AI2PS tool for calculating

phonon properties by connecting density functional theory (DFT) software to the DMDW analysis tool. Chapter 4 presents the previously published results for calculating phonon contributions to the electron spectral function using AI2PS, which demonstrates the use of phonon properties such as the phonon density of states and Eliashberg function. In chapter 5, we present the Corvus software package, the updated and expanded incarnation of the AI2PS tool, developed for general physical simulations that depend on the coordinated execution of multiple software packages, which is necessary for simulating many phonon properties. Chapter 6 details our use of the vibrational free energy in modeling the thermal expansion—a complex calculation workflow recently implemented in CORVUS that now allows us to include expansion effects in calculated properties—as well as heat capacity at constant volume and pressure. Concluding remarks are given in chapter 7, followed by appendices with more detailed information on using the software packages described here.

1.3 Conventions and Notation

Throughout this dissertation, Hartree atomic units ($\hbar = e = m_e = k_B = 1$) can be assumed unless stated otherwise.

Chapter 2

PHONON THEORY

Lattice vibrations can be thought of as oscillations of the atoms about the equilibrium static lattice positions. The vibrational effects can be determined from the equation of motion, which is encapsulated in the quantity called the dynamical matrix, from which various macroscopic physical properties can be modeled. This chapter will give a brief overview of the theoretical background for the phonon properties simulated in this work.

2.1 Dynamical Matrix and Vibrational Density of States

Starting with a monoatomic lattice, for notational simplicity, we consider an atom located at a Bravais lattice site $\mathbf{R}$. In the static approximation, the atom is fixed at $\mathbf{R}$ at all time t. We relax this assumption, and allow the atom to oscillate about the lattice site with displacement $\mathbf{u}$, such that its instantaneous position is given by

$$\mathbf{r}(\mathbf{R}) = \mathbf{R} + \mathbf{u}(\mathbf{R}), \tag{2.1}$$

where $\mathbf{u}$ is small compared to the interatomic spacing and the mean position of the ion is still located at $\mathbf{R}$. Given the pair potential ϕ between two atoms, the total potential energy U is just the sum of ϕ for all pairs in the crystal

$$U = \frac{1}{2}\sum_{\mathbf{R}\mathbf{R}'} \phi(\mathbf{r}(\mathbf{R}) - \mathbf{r}(\mathbf{R}')) = \frac{1}{2}\sum_{\mathbf{R}\mathbf{R}'} \phi(\mathbf{R} - \mathbf{R}' + \mathbf{u}(\mathbf{R}) - \mathbf{u}(\mathbf{R}')). \tag{2.2}$$

If the $\mathbf{u}(\mathbf{R})$ are small, we can use the three-dimensional Taylor expansion of U about its equilibrium value

$$U = \frac{N}{2}\sum \phi\left(\mathbf{R}\right) + \frac{1}{2}\sum_{\mathbf{R}\mathbf{R}'}\left(\mathbf{u}\left(\mathbf{R}\right) - \mathbf{u}\left(\mathbf{R}'\right)\right)\cdot\nabla\phi\left(\mathbf{R}-\mathbf{R}'\right) + \frac{1}{4}\sum_{\mathbf{R}\mathbf{R}'}\left[\left(\mathbf{u}\left(\mathbf{R}\right) - \mathbf{u}\left(\mathbf{R}'\right)\right)\cdot\nabla\right]^2\phi\left(\mathbf{R}-\mathbf{R}'\right) + O\left(u^3\right). \quad (2.3)$$

The equilibrium potential energy, the zeroth-order term, is independent of the dynamics of the atoms and can therefore be neglected in many dynamical problems that do not rely on the absolute energy of the crystal or its equilibrium size. Additionally, the force exerted on the atom at $\mathbf{R}$ by all other atoms in the lattice, when all atoms are located at their equilibrium position, is given by

$$\sum_{\mathbf{R}'}\left[-\nabla\phi\left(\mathbf{R}-\mathbf{R}'\right)\right], \quad (2.4)$$

and must therefore vanish for the equilibrium configuration. This expression is, modulo a negative sign, the coefficient of the linear term in Eq. 2.3, meaning the first nonvanishing correction to the equilibrium potential energy is the quadratic term, i.e., a harmonic approximation. The harmonic term is then given by

$$\begin{aligned} U_{\text{harm}} &= \frac{1}{4}\sum_{\mathbf{R}\mathbf{R}'}\left[\mathbf{u}\left(\mathbf{R}\right) - \mathbf{u}\left(\mathbf{R}'\right)\right]\boldsymbol{\phi}\left(\mathbf{R}-\mathbf{R}'\right)\left[\mathbf{u}\left(\mathbf{R}\right) - \mathbf{u}\left(\mathbf{R}'\right)\right], \\ \phi_{\mu\nu}\left(\mathbf{r}\right) &= \frac{\partial^2\phi\left(\mathbf{r}\right)}{\partial r_\mu \partial r_\nu}. \end{aligned} \quad (2.5)$$

Higher order corrections are known as anharmonic terms, generally treated as perturbations of the dominant harmonic term, but will be neglected in this work. Finally, the harmonic term can be rewritten in the general form

$$\begin{aligned} U_{\text{harm}} &= \frac{1}{2}\sum_{\mathbf{R}\mathbf{R}'}\mathbf{u}\left(\mathbf{R}\right)\mathbf{D}\left(\mathbf{R}-\mathbf{R}'\right)\mathbf{u}\left(\mathbf{R}'\right), \\ D_{\mu\nu}\left(\mathbf{R}-\mathbf{R}'\right) &= \delta_{\mathbf{R},\mathbf{R}'}\sum_{\mathbf{R}''}\phi_{\mu\nu}\left(\mathbf{R}-\mathbf{R}''\right) - \phi_{\mu\nu}\left(\mathbf{R}-\mathbf{R}'\right). \end{aligned} \quad (2.6)$$

For N ions of mass M, the $(3N)$ equations of motion given by $\mathbf{F} = -\boldsymbol{\nabla} U$ are then

$$M\ddot{\mathbf{u}}\left(\mathbf{R}\right) = -\boldsymbol{\nabla} U_{\text{harm}} = -\sum_{\mathbf{R}'} \mathbf{D}\left(\mathbf{R}-\mathbf{R}'\right)\mathbf{u}\left(\mathbf{R}'\right). \tag{2.7}$$

To obtain the normal modes of the oscillating system, we use plane waves $\mathbf{u}(\mathbf{R},t) = \boldsymbol{\epsilon} e^{i(\mathbf{k}\cdot\mathbf{R}-\omega t)}$ as an ansatz, where $\boldsymbol{\epsilon}$ is the polarization vector. Under Born-von Karman periodic boundary conditions, where $\mathbf{u}(\mathbf{R}+N_i\mathbf{a}_i) = \mathbf{u}(\mathbf{R})$ for each primitive vector $\mathbf{a}_i$ and N_i are large integers such that $N = N_1N_2N_3$, wave vectors are restricted N allowed values of the form

$$\mathbf{k} = \frac{n_1}{N_1}\mathbf{b}_1 + \frac{n_2}{N_2}\mathbf{b}_2 + \frac{n_3}{N_3}\mathbf{b}_3, \tag{2.8}$$

where the reciprocal lattice vectors $\mathbf{b}_i$ satisfy the relation $\mathbf{b}_i \cdot \mathbf{a}_i = 2\pi\delta_{ij}$. Substituting the ansatz solution into the equations of motion yields

$$M\omega^2\boldsymbol{\epsilon} = \mathbf{D}\left(\mathbf{k}\right)\boldsymbol{\epsilon}, \tag{2.9}$$

where

$$\mathbf{D}\left(\mathbf{k}\right) = \sum_{\mathbf{R}} \mathbf{D}\left(\mathbf{R}\right) e^{-i\mathbf{k}\cdot\mathbf{R}} \tag{2.10}$$

is known as the dynamical matrix of force constants. [8,9] The three-dimensional eigenvalue problem gives three solutions for each of the N allowed values of $\mathbf{k}$, yielding $3N$ normal modes that correspond to the phonon frequencies $\omega(\mathbf{k})$ of the system. The real space dynamical matrix can be expressed in terms of second derivatives of the total internal energy U with respect to the $i = x, y, z$ Cartesian displacements $u_i(\mathbf{R})$ of the atom associated with lattice site $\mathbf{R}$

$$D_{ij'} = \frac{1}{\sqrt{M(\mathbf{R})M(\mathbf{R}')}} \frac{\partial^2 U}{\partial u_i(\mathbf{R})\partial u_{j'}(\mathbf{R}')}, \tag{2.11}$$

where U contains implicit thermal dependence of the lattice geometry. Previously, the force constants were roughly modeled by Born–von Karman parameters, dominated by nearest and next–nearest neighbor terms, which were determined through fitting elastic, calorimetric, and spectroscopic data, but this method was limited to the temperatures specified by the experiments. With modern condensed matter simulation software, a system can be modeled with

any given expansion or contraction, which can in turn give us full temperature dependence of the dynamical matrix. Tables 2.1 and 2.2 give example Born–von Karman parameters for a selection of materials and temperatures from experimental fits and theoretical calculations.

From the real space dynamical matrix $\mathbf{D}$, we can calculate the vibrational density of states (VDOS) for a given multiple–scattering (MS) path P from the imaginary part of the lattice dynamical Green's function using the Lanczos algorithm [15–18]

$$\rho_P(\omega) = -\frac{2\omega}{\pi}\mathrm{Im}\left\langle 0 \left| \frac{1}{\omega^2 - \mathbf{D} + i\epsilon} \right| 0 \right\rangle, \tag{2.12}$$

where $|0\rangle$ is an appropriate Lanczos seed vector representing an initial displacement of atoms along the path P and $\mathbf{D}$ is the real space dynamical matrix. Similarly, the VDOS $\rho_a(\omega)$ for atomic distortions can also be calculated using Equation 2.12 with the seed state representing the distortion. These two forms of the VDOS allow us to calculate the Debye–Waller factors for a system, as discussed in the following section.

2.2 Debye–Waller Factors

Debye–Waller factors are used to describe the exponential damping observed in x–ray scattering spectra due to thermal vibrations and configurational disorder. In normalized x-ray absorption fine structure (XAFS) spectra $\chi(k) = (\mu - \mu_0)/\mu_0$, the effect of vibrations is dominated by an average over the oscillatory behavior $\chi_R \propto \sin(2kR + \Phi)$ of the MS paths. Figure 2.1 shows an example of copper x–ray absorption spectra simulated by the FEFF9 code with and without the proper XAFS Debye–Waller factors, clearly demonstrating the importance of their inclusion and therefore the accurate determination of σ^2.

The oscillations can be expressed in terms of the cumulant expansion as follows [14]

$$\begin{aligned} \left\langle e^{i2kr} \right\rangle &\equiv e^{2ikR_0} e^{-W(T)} \\ W(T) &= -\sum_{n=1}^{\infty} \frac{(2ik)^n}{n!} \sigma^{(n)}(T), \end{aligned} \tag{2.13}$$

where R_0 and r represent the equilibrium length without vibrations instantaneous length respectively. The dominating term $W(T) \approx 2k^2\sigma^2(T)$ is related to the mean square relative

displacement (MSRD) of the scattering path R [20]

$$\sigma^{(2)} = \left\langle (r - \bar{r})^2 \right\rangle \equiv \sigma_R^2 (T) , \tag{2.14}$$

which is referred to as the XAFS Debye–Waller factor. Evaluating the average using Bose–Einstein statistics for a normal mode ω_λ and normal coordinate q_λ [16, 17]

$$\omega^2 \langle q_\lambda \rangle^2 = \left\langle n(\omega_\lambda) + \frac{1}{2} \right\rangle \hbar\omega_\lambda = \frac{1}{2} \left[\frac{2}{e^{\beta\hbar\omega_\lambda} - 1} + 1 \right] \hbar\omega_\lambda = \frac{\hbar\omega_\lambda}{2} \coth\left(\frac{\beta\hbar\omega_\lambda}{2} \right) , \tag{2.15}$$

where $\beta \equiv 1/k_B T$. The XAFS Debye–Waller factor is then

$$\sigma_R^2 (T) = \frac{\hbar}{2\mu_R} \int_0^\infty \frac{d\omega}{\omega} \coth\left(\frac{\beta\hbar\omega}{2} \right) \rho_R (\omega) , \tag{2.16}$$

where μ_R is the reduced mass associated with path R. The mean–square displacements $u^2(T)$ for a given atom, referred to as the crystallographic Debye–Waller factors, can likewise be calculated from Equation 2.16 by replacing the VDOS with that for an atomic distortion $\rho_a(\omega)$. [14]

2.3 Electron–phonon Coupling and Superconducting Transition

The Hamiltonian for a many–electron system with bare energies given by ϵ_k^0 coupled linearly to phonons with energies ω_q can be expressed as

$$H = \sum_k \epsilon_k^0 c_k^\dagger c_k + \sum_q \omega_q a_q^\dagger a_q + \sum_{kk'q} V_{kk'}^q \left(a_q + a_q^\dagger \right) c_k^\dagger c_{k'}, \tag{2.17}$$

where $V_{kk'}^q$ represents the electron–phonon potential, and $c_k^\dagger$ (c_k) and $a_q^\dagger$ (a_q) are the electron and phonon creation (annihilation) operators respectively. The electron–phonon matrix elements are given by [21–24]

$$g_{\mathbf{kk'}}^{\mathbf{q}} = \sum_\sigma \langle k | \, \delta^{\mathbf{q}} V \, | \mathbf{k} \rangle . \tag{2.18}$$

The $g_{\mathbf{kk'}}^{\mathbf{q}}$ represent scattering from a point $\mathbf{k}$ to $\mathbf{k'}$ in $\mathbf{k}$–space, but it is often useful to look at the coupling between a state with energy $\epsilon_{\mathbf{k}}$ and all other final states with energy $\epsilon_{\mathbf{k}} \pm \hbar\omega$. This coupling is given by the Eliashberg function

$$\alpha^2 F_{\mathbf{k}} (\omega) = \frac{1}{2\pi N(\epsilon_{\mathrm{F}})} \sum_{\mathbf{qk'}} \left| g_{\mathbf{kk'}}^{\mathbf{q}} \right|^2 \delta (\omega - \omega_q) , \tag{2.19}$$

where ϵ_{F} is the Fermi energy. The inverse moment of the Eliashberg function gives the electron–phonon coupling constant λ [9, 25]

$$\lambda = 2\int_0^{\omega_{\max}} \frac{\alpha^2 F(\omega)}{\omega} d\omega, \tag{2.20}$$

where $\alpha^2 F(\omega) = \sum_{\mathbf{k}} \alpha^2 F_{\mathbf{k}}(\omega)$. The electron–phonon coupling constant λ, Coulomb pseudopotential μ^*, and Debye temperature Θ_{D} can be related in the McMillan relation [9, 21]

$$T_{\mathrm{c}} = \frac{\Theta_{\mathrm{D}}}{1.45} \exp\left[-\frac{1.04\,(1+\lambda)}{\lambda - \mu^*\,(1+0.62\lambda)}\right], \tag{2.21}$$

which allows us to determine bounds on T_{c} using λ and estimates of μ^*.

2.4 Phonon Contributions to Electron Self–Energy and Spectral Function

The electron self–energy describes the energy of an electron resulting from its effect on the surroundings. The interaction of an electron with a single phonon with energies ϵ and ω respectively at temperature T can be characterized by a self–energy in the Einstein model [9, 26, 27]

$$\Sigma^{\mathrm{E}}(\epsilon, \omega, T) = \frac{1}{2}\int_{-\infty}^{\infty} \left[\frac{n(\omega,T) + 1 - f(\omega,T)}{\epsilon - \nu - \omega} + \frac{n(\omega) + f(\omega)}{\epsilon - \nu + \omega}\right] d\nu, \tag{2.22}$$

where f and n denoted the Fermi and Bose distributions respectively. This can then be analytically continued as [26–29]

$$\tilde{\Sigma}^{\mathrm{E}}(z, \omega, T) = -i\pi\left(n(\omega) + \frac{1}{2}\right) + \frac{1}{2}\Psi\left(\frac{1}{2} + i\frac{\omega - z}{2\pi T}\right) - \frac{1}{2}\Psi\left(\frac{1}{2} - i\frac{\omega + z}{2\pi T}\right), \tag{2.23}$$

where Ψ represents the Digamma function. The full electron self–energy is then found by integrating over all phonon modes weighted by the electron–phonon coupling [9]

$$\Sigma_{\mathbf{k}}(\epsilon) = \int 2\Sigma^{\mathrm{E}}(\epsilon, \omega)\; \alpha^2 F(\omega)\,. \tag{2.24}$$

The one–electron Green's function in frequency space is then

$$G_{\mathbf{k}}(\omega) = \frac{1}{\omega - [\epsilon_{\mathbf{k}} - \epsilon_{\mathrm{F}}] - \Sigma_{\mathbf{k}}(\omega)}. \tag{2.25}$$

To be more complete, one would include another self–energy term with $\Sigma_{\mathbf{k}}$ due to electron–electron interactions, though such terms give negligible corrections in most cases and are typically dropped. The spectral function $A_{\mathbf{k}}(\omega)$ for electrons interacting with phonons is then [9, 28]

$$A_{\mathbf{k}}(\omega) = \frac{1}{\pi}\left|\mathrm{Im}G_{\mathbf{k}}(\omega)\right| = \frac{1}{\pi}\frac{\left|\mathrm{Im}\Sigma_{\mathbf{k}}(\omega)\right|}{\left\{\omega - \left[\epsilon_{\mathbf{k}} - \epsilon_{\mathrm{F}}\right] - \mathrm{Re}\Sigma_{\mathbf{k}}(\omega)\right\}^2 + \left[\mathrm{Im}\Sigma_{\mathbf{k}}(\omega)\right]^2}. \tag{2.26}$$

The above approximate forms of $G_{\mathbf{k}}(\omega)$ and $A_{\mathbf{k}}(\omega)$ are called the Migdal approximation. [30, 31] An alternative method using the retarded Green's function [32] is described and contrasted with the above method in the publication presented in Chapter 4.

2.5 Vibrational Free Energy and Thermal Expansion

In a canonical ensemble, a mechanical system of fixed N, V in thermal equilibrium, the Helmholtz free energy F is the minimum free energy of the system and can be expressed in terms of the partition function Z as $F = -k_B T \ln Z$. [33,34] For the vibrational contribution F_{vib}, we first consider the partition function for a single harmonic oscillator of frequency ω_i [20, 35]

$$Z_i = \sum_n^{\infty} e^{-\beta E_n} = \sum_i^{\infty} e^{-\beta\hbar\omega_i\left(n+\frac{1}{2}\right)} = \frac{e^{-\beta\hbar\omega_i/2}}{1 - e^{-\beta\hbar\omega_i}} = \frac{1}{2\sinh\left(\frac{\beta\hbar\omega_i}{2}\right)}, \tag{2.27}$$

which gives a partition function for N lattice atoms as

$$Z = \prod_i^{3N} Z_i = \prod_i^{3N} \frac{1}{2\sinh\left(\frac{\beta\hbar\omega_i}{2}\right)}, \tag{2.28}$$

where $\beta \equiv 1/k_B T$. Substituting this into the expression above for F, gives us the following form for F_{vib} [20, 35]

$$F_{\mathrm{vib}} = k_B T \sum_i^{3N} \ln\left[2\sinh\left(\frac{\beta\hbar\omega_i}{2}\right)\right]. \tag{2.29}$$

The Helmholtz free energy as a function of physical size and temperature of the system can then be determined from the sum of the total internal energy $U(V)$ and the vibrational free energy $F_{\mathrm{vib}}(T, V)$ [20]

$$F(T, V) = U(V) + k_B T \int_0^{\infty} d\omega \ln\left[2\sinh\left(\frac{\beta\hbar\omega}{2}\right)\right] \rho_V(\omega), \tag{2.30}$$

where $\rho_V(\omega)$ is the total VDOS (e.g., sum of all three Cartesian distortions) for the system at volume V. Since F is minimized when the system is in its equilibrium configuration at fixed temperature and volume, we can determine the thermal expansion of the system $V(T)$ by minimizing F with respect to V at a given temperature T. The average pressure is related to F as [34]

$$P = -\left(\frac{\partial F}{\partial V}\right)_T, \tag{2.31}$$

which means we can also calculate the Gibbs free energy $G = F + PV$ after determining $V(T)$ and $F(V,T)$.

2.6 Heat Capacity

The heat capacity at constant volume C_V is given by

$$C_V = \left(\frac{\partial U}{\partial T}\right)_V. \tag{2.32}$$

Condensed matter simulations generally calculate the total internal energy U for a given structure geometry, meaning the simulations are parameterized by size $V(T)$ instead of temperature T. However, as with the vibrational free energy above, the heat capacity can be reexpressed as an integral over the VDOS obtained from the dynamical matrix calculation. For a canonical ensemble of fixed N and V, the total internal energy is [34]

$$U = \frac{\sum_n^\infty E_n e^{-\beta E_n}}{\sum_n^\infty e^{-\beta E_n}} = -\frac{\partial \ln Z}{\partial \beta}. \tag{2.33}$$

Substituting the results from Equation 2.28, we get

$$\ln Z = \ln \prod_i^{3N} \frac{1}{2\sinh\left(\frac{\beta\hbar\omega_i}{2}\right)} = -\sum_i^{3N} \ln\left[2\sinh\left(\frac{\beta\hbar\omega_i}{2}\right)\right], \tag{2.34}$$

which yields the following expression for the internal energy

$$U = \sum_i^{3N} \frac{\hbar\omega_i}{2} \coth\left(\frac{\beta\hbar\omega_i}{2}\right). \tag{2.35}$$

Finally, the temperature derivative gives the heat capacity for constant volume

$$C_V = \frac{\partial U}{\partial \beta}\frac{\partial \beta}{\partial T} = k_B \sum_i^{3N} \left(\frac{\beta\hbar\omega_i}{2}\right)^2 \operatorname{csch}^2\left(\frac{\beta\hbar\omega_i}{2}\right) \tag{2.36}$$

or expressed in integral form

$$C_V\left(T,V\right) = k_B \int_0^\infty d\omega \left(\frac{\beta\hbar\omega_i}{2}\right)^2 \operatorname{csch}^2\left(\frac{\beta\hbar\omega_i}{2}\right) \rho_V\left(\omega\right). \tag{2.37}$$

To calculate the heat capacity at constant pressure

$$C_P = \left(\frac{\partial H}{\partial T}\right)_P, \tag{2.38}$$

where $H = U + PV$ is the enthalpy of the system, we instead calculate the difference $C_P - C_V$ and add the above result for C_V. Taking the thermodynamic identities for internal energy and enthalpy [34]

$$dU = T\,dS - P\,dV - \mu\,dN \tag{2.39}$$

$$dH = T\,dS + V\,dP + \mu\,dN \tag{2.40}$$

at constant volume and pressure respectively (for fixed N), we see

$$\left(\frac{\partial U}{\partial T}\right)_V = T\left(\frac{\partial S}{\partial T}\right)_V = C_V \tag{2.41}$$

$$\left(\frac{\partial H}{\partial T}\right)_P = T\left(\frac{\partial S}{\partial T}\right)_P = C_P. \tag{2.42}$$

Expressing the entropy as a function of temperature and volume

$$S = \left(\frac{\partial S}{\partial T}\right)_V dT + \left(\frac{\partial S}{\partial V}\right)_T dV, \tag{2.43}$$

the volume as a function of temperature and pressure

$$V = \left(\frac{\partial V}{\partial T}\right)_P dT + \left(\frac{\partial V}{\partial P}\right)_T dP, \tag{2.44}$$

and making use of the Maxwell relation [34]

$$\left(\frac{\partial S}{\partial V}\right)_T = \left(\frac{\partial P}{\partial T}\right)_V, \tag{2.45}$$

we have

$$S = \frac{C_V}{T} dT + \left(\frac{\partial P}{\partial T}\right)_V \left[\left(\frac{\partial V}{\partial T}\right)_P dT + \left(\frac{\partial V}{\partial P}\right)_T dP\right], \tag{2.46}$$

which results in

$$T\left(\frac{\partial S}{\partial T}\right)_P = C_P = C_V + T\left(\frac{\partial P}{\partial T}\right)_V \left(\frac{\partial V}{\partial T}\right)_P. \tag{2.47}$$

Mathematically, the partial derivative $(\partial P/\partial T)_V$ can be rewritten [34]

$$\left(\frac{\partial P}{\partial T}\right)_V = -\frac{(\partial V/\partial T)_P}{(\partial V/\partial P)_T}, \tag{2.48}$$

which yields

$$C_P = C_V - T\frac{(\partial V/\partial T)_P^2}{(\partial V/\partial P)_T}. \tag{2.49}$$

This is commonly written in terms of the (volumetric) thermal expansion coefficient α_V

$$\alpha_V = \frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_P \tag{2.50}$$

and the isothermal bulk modulus B_T

$$B_T = -V\left(\frac{\partial P}{\partial V}\right)_T, \tag{2.51}$$

giving the final result [33, 34]

$$C_P - C_V = VT\,\alpha_V^2\,B_T. \tag{2.52}$$

Table 2.1: Born–von Karman parameters D_{ij}^{m} (N/m) up to next–nearest neighbors for select metals calculated at ground state ($T = 0$) using density functional theory (DFT) with local density approximation (LDA) exchange–correlation functional, in addition to fits from neutron inelastic scattering [10–12] and x–ray thermal diffusion scattering experiments at room temperature. [13]

	Neighbor position m	ij	Th. (0 K)	Expt. (RT)	Ref.
Cu	(1,1,0)a/2	xx	14.82	13.124	[10]
		zz	−3.11	−1.503	
		xy	17.66	14.626	
	(2,0,0)a/2	xx	1.18	0.525	
		yy	−0.21	0.265	
Ta	(1,1,1)a/2	xx	19.80	16.983	[13]
		xy	8.88	11.201	
	(2,0,0)a/2	xx	3.88	1.182	
		yy	1.89	1.423	
Nb	(1,1,1)a/2	xx	18.26	14.14(9)	[12]
		xy	5.41	8.84(12)	
	(2,0,0)a/2	xx	15.45	14.16(9)	
		yy	−1.71	−3.64(20)	
V	(1,1,1)a/2	xx	14.42	10.872(380)	[11]
		xy	4.18	7.244(390)	
	(2,0,0)a/2	xx	7.67	6.494(840)	
		yy	−0.63	−2.148(530)	

Table 2.2: Born–von Karman parameters D_{ij}^m (N/m) up to next–nearest neighbors for copper at select temperatures calculated using DFT/LDA and fit from neutron scattering experiments.

Neighbor position m	ij	0 K	49 K		298 K	
		Th.	Th.	Expt.	Th.	Expt.
(1,1,0)a/2	xx	14.82	14.53[a]	13.278[b]	13.87[a]	13.124[b]
	zz	−3.11	−3.17[a]	−1.351[b]	−3.01[a]	−1.503[b]
	xy	17.66	17.12[a]	14.629[b]	16.33[a]	14.626[b]
(2,0,0)a/2	xx	1.18	1.07[a]	−0.041[b]	1.06[a]	0.525[b]
	yy	−0.21	−0.12[a]	−0.198[b]	−0.12[a]	−1.265[b]

[a]Ref. [14] [b]Ref. [10]

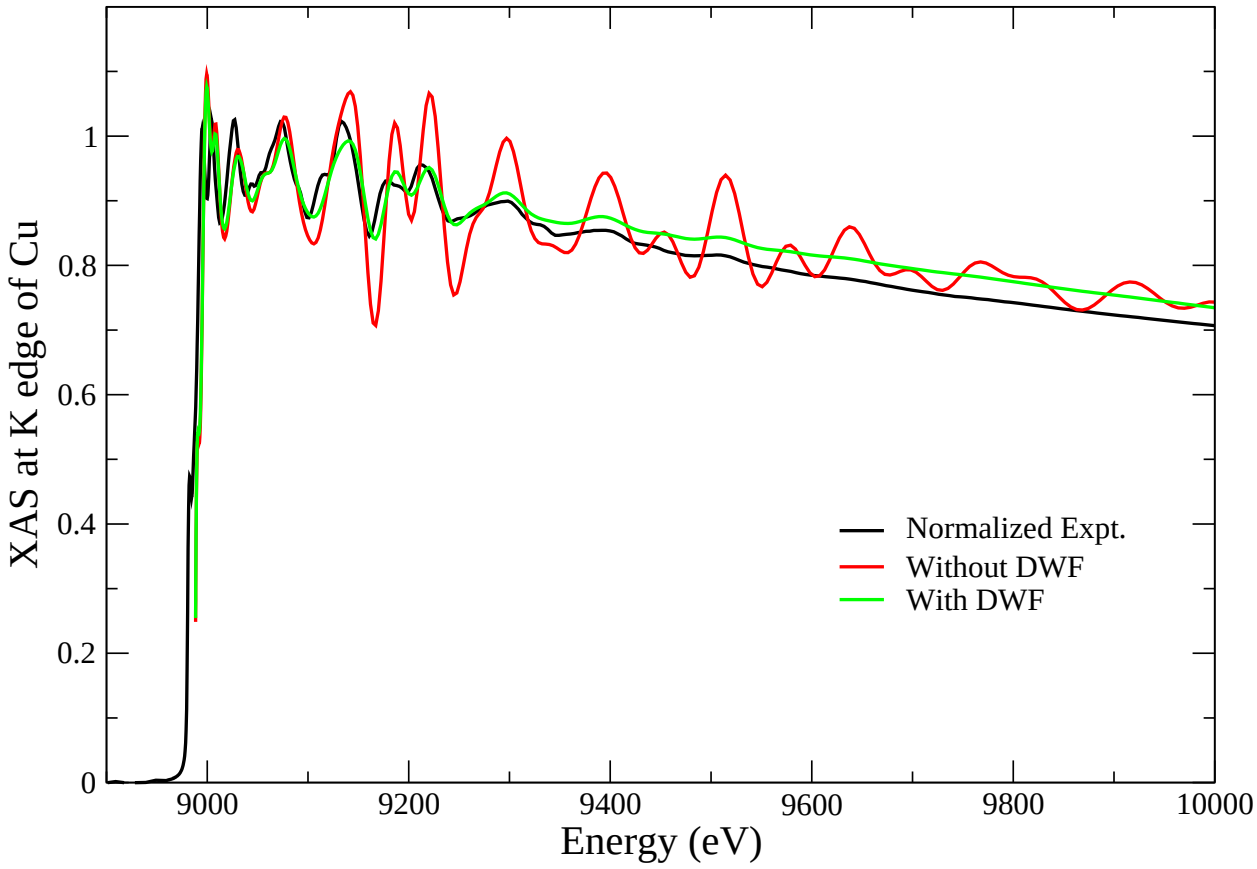

Figure 2.1: Cu x–ray absorption spectra at 298 K simulated with the FEFF9 software package using the correlated Debye model (Θ_D = 315 K) with and without XAFS Debye–Waller factors σ^2, along with experimental spectra for reference. [19] The XAS amplitudes are significantly improved by the inclusion of σ^2.

Chapter 3

AI2PS: AN INTEGRATED WORKFLOW FOR AB INITIO DFT CALCULATIONS OF VIBRATIONAL PROPERTIES

3.1 Introduction

Vibrational properties—such as Debye–Waller factors, phonon self energy, and vibrational free energies—are key to understanding many aspects of materials. Thus, their simulation using *first principles* methods is of particular importance to computational physicists. As shown in Chapter 2, these properties can be calculated from the dynamical matrix, or interatomic forces, and other quantities produced by density functional theory (DFT) calculations, but require additional analysis tools to translate all these quantities ultimately into these phonon properties.

The process of manually coordinating the simulation and analysis calculations is a tedious undertaking for the user, which motivated the creation of an automated tool to run the phonon property workflow. This kind of automated workflow coordination has already been accomplished for UV/Visual and x–ray simulations with our AI2NBSE and OCEAN [36–39] packages. With the addition of the AI2PS (ab initio to phonon spectra) tool, capable of connecting DFT software packages with the DMDW analysis package, we are able to efficiently simulate phonon properties such as XAFS (x–ray absorption fine structure) Debye–Waller factors σ^2, crystallographic Debye–Waller factors u^2, vibrational free energy, and phonon contributions to the electron spectral function and electron self–energy. Here we present the AI2PS tool, along with example calculations of the aforementioned properties.

3.2 AI2PS

3.2.1 Global structure

AI2PS serves as an automated driver for three main components—a DFT engine, an interfacing program, and a vibrational properties engine. The DFT engine is responsible for producing the physical quantities of a given structure. The interfacing program takes that output and translates it into a format compatible with the vibrational properties engine, which calculates the desired vibrational property.

We developed AI2PS to work with the standalone module DMDW, which is also packaged with the FEFF9 code. DMDW is capable of calculating numerous vibrational properties when provided with a properly formatted dynamical matrix and electron–phonon coupling coefficients. AI2PS has been designed to be modular, in that we can connect with various DFT packages by writing an associated interfacing program.

The interfacing has been already been established manually for several DFT software packages—ABINIT,Gaussian, Quantum ESPRESSO, NWChem, VASP. Here, we focus first on the implementation with ABINIT as the DFT engine, as it is often by our group and collaborators, though additional DFT packages can be supported in the future.

3.2.2 ABINIT program

ABINIT is an open source package that computes electron density and derived quantities within DFT using pseudopotentials and a planewave basis for periodic and molecular systems, as well as response functions using DFPT necessary for modeling vibrational properties. [40–44] Their active development and collaboration efforts made for an attractive intial use case for the AI2PS project.

AI2PS requires ABINIT version 7.0 or above. The interfacing program for ABINIT utilizes the interatomic force constants calculated internally. A patch is included with AI2PS to print out this data if using versions 7.0 or 7.2. Starting in version 7.4, ABINIT now includes the `prt_ifc` option in the ANADDB utility, which will do the same when enabled.

The calculation of the interatomic force constants necessary for the modeling of phonon properties can be computationally expensive, and rely heavily on the parallelization available in the ABINIT software. For instance, we found that the calculation of the electron–phonon couplings, essential to the accuracy of the electron–phonon coupling constant λ and therefore estimates of the superconducting transition temperature T_c, required a relatively large k–point grid (`ngkpt`=32×32×32) and planewave energy cutoff (`ecut`=50.0 Ha) for convergence, which took roughly 200 minutes on 160 AMD Opteron 6128 (800 MHz) cores spread across ten nodes of our computer cluster.

3.2.3 DMDW program

DMDW is an open source package for the calculation of vibrational properties, namely XAFS Debye–Waller factors σ^2, crystallographic Debye–Waller factors u^2, vibrational free energy, electron–self energy with phonon contributions, and electron spectral function with phonon contributions, from the dynamical matrix using the Lanczos recursion algorithm, [15–18] providing a better alternative to the correlated Einstein and Debye models. [14] It is released as a standalone package, as well as a module integrated into the FEFF9 code. [45, 46] DMDW is a serial code, though its computations take only a few minutes on any modern processor.

3.2.4 AI2PS Driver

The main AI2PS driver coordinates the calculation workflow. In general, an AI2PS workflow consists of a DFT calculation that generates the interatomic force constants or dynamical matrix, the translation of this output into the input format for DMDW, and finally the execution of DMDW to produce the desired vibrational property. In the case of ABINIT, the generation of the interatomic forces is itself a multi–step calculation using ABINIT and its analysis tool ANADDB, which produces the interatomic force constants and the electron–phonon couplings.

The full process, illustrated in Figure 3.1, starts with parsing the single input file containing relevant physical parameters defining the system. The AI2PS input file consists of

keywords followed by values enclosed in curly braces (see Appendix A for example input files). AI2PS then generates all the input files necessary for each software package involved in the full calculation, ABINIT and DMDW in this case, and stores them in a single subdirectory. The ABINIT calculation consists of running the main `abinit` tool to produce the multitude of derivative database (DDB) and electron–phonon matrix (GKK) files. Each set of these files has a respective merging tool—`mrgddb` and `mrggkk`—that consolidates all this information into input for ABINIT's analysis tool ANADDB, which in turn produces the interatomic force constants in the dynamical matrix, as well as the electron–phonon coupling constants used in calculating the electron self–energy and electron spectral function with phonon contributions. This output is fed into AI2PS's interfacing translator tool, which formats the quantities as needed for use in DMDW. Lastly, DMDW is used to calculate the selected phonon property.

3.3 Applications

This section gives an example calculation of each vibrational property that can be produced using AI2PS with the ABINIT and DMDW software packages. AI2PS input files used for these calculations can be found in Appendix A, as well as more details on how to run AI2PS. Chapter 4 presents published results for the phonon contributions to the electron spectral function and related quantities for Cu, Ta, V, Nb, and Pb, all of which were obtained using AI2PS.

3.3.1 XAFS DWF

X–ray absorption fine structure (XAFS) spectra demonstrate exponential damping with increasing temperature and energy of approximate form $\exp[-2k^2\sigma^2(T)]$, due to displacements of the x–ray multiple–scattering, and is commonly described quantitatively by the XAFS Debye-Waller factors $\sigma^2(T)$. An example for nearest neighbor single scattering in Cu is shown in Figure 3.2, which was generated using the input file found in Appendix A, calculated at both the experimental lattice constant, $a = 6.82$ Bohr, and the lattice constant for the optimized cell, $a = 6.73339$ Bohr. Both calculations used the standard LDA exchange–

correlation functional. One can see how much the choice of lattice constant affects the results, which is due to the anharmonicity of the dynamical matrix. Improved results can be obtained by using temperature–dependence of the lattice constant. [14]

3.3.2 Crystallographic DWF

Crystallographic Debye–Waller factors correspond to the effects of individual atomic displacements on x–ray and neutron scattering spectra, where the exponential damping factor is of approximate form $\exp[-\frac{1}{2}k^2u^2]$ and $u^2(T)$ is the mean–square relative displacement on a given atom. An example for Cu is shown in Figure 3.3, which was generated using the input file found in Appendix A. Again, the u^2 were calculated at both the experimental lattice constant, $a = 6.82$ Bohr, and the lattice constant for the optimized cell, $a = 6.73339$ Bohr, to demonstrate the sensitivity to the lattice constant choice, and therefore the need for more extensive workflow that can first simulate $a(T)$ to account for thermal expansion effects in the dynamical matrix and any subsequently derived phonon properties. Also shown in Figure 3.4 and Figure 3.5 are the u^2 for Ga and As atoms, respectively, in bulk GaAs. GaAs simulations using the LDA function do not have the same issue of systematically underestimating Debye–Waller factors that metals like Cu do. [14, 48]

3.3.3 Electron self-energy and spectral function

AI2PS can also be used to calculation phonon contributions to the electron self–energy Σ and electron spectral function $A(\omega)$ to study phonon excitations. An example for Cu is shown in Figure 3.7, which was generated using the input file found in Appendix A, however, a more in–depth look into the phonon effects with the electron spectral function for Cu, Ta, V, Nb, and Pb can be found in Chapter 4.

3.3.4 Vibrational free energy

DMDW also gives AI2PS access to the vibrational free energy, shown for bulk Cu in Figure 3.8. The vibrational free energy is a quantity of particular interest, because it can be combined with the total internal energy U to get the Helmholtz free energy $F(T, V)$, which can be minimized at all T to determine $V(T)$ (see Chapter 2). Therefore, it is possible to use the vibrational free energy produced by this workflow to determine the thermal expansion of the lattice constant $a(T)$. This in turn means it is possible to include thermal expansion effects in the dynamical matrix and other phonon properties, but requires a more complex calculation workflow structure, essentially a nested workflow of AI2PS workflows!

3.4 Conclusions

We have demonstrated an integrated code, AI2PS, for ab initio calculations of vibrational properties using the dynamical matrix and electron–phonon couplings produced by the DFT package ABINIT. The same machinery can accommodate additional DFT packages in the future with the modular design of AI2PS, which would allow the user to pick the DFT software best suited for the condensed matter system being simulation. The automation of the numerous tasks related to mere technical details of running the calculations (e.g., input generation and coordinated file names) significantly reduces time spent preparing these phonon–property calculations, which means users can focus on exploring the physics instead.

We have also seen that, due to the anharmonicity of the dynamical matrix, i.e., the interatomic force constants, simulation of phonon properties become sensitive to the choice of lattice constant when specifying the system of study. It is therefore imperative to establish the thermal expansion of the system's structure first, before calculating other phonon properties. As alluded to in Section 3.3.4, access to the vibrational free energy gives us the ability to calculate the thermal expansion of the lattice constant $a(T)$, though it is even more complicated to coordinate as it would require an iteration of AI2PS calculations to build the $F(T, V)$—or $F(T, a)$—surface and an analysis tool to determine the $a(T)$ curve

that minimizes it.

Nevertheless, it is within our capabilities to do so, but building the next level workflow tool to accomplish this in an elegant and sustainable way requires us to rethink our construction of scientific workflows. Continuing from the work accomplished with AI2PS, covered in detail in Chapter 4, we transferred the development of vibrational property calculations to the redesigned scientific workflow framework that became CORVUS, which is discussed in Chapter 5. Finally, the sought after thermal expansion automated workflow was successfully implemented, and those results are presented in Chapter 6.

The ABINIT code is a common project of the Université Catholique de Louvain, Corning Incorporated and other contributors (See `http://www.abinit.org`). This work was supported in part by NSF Grant PHY–0835543.

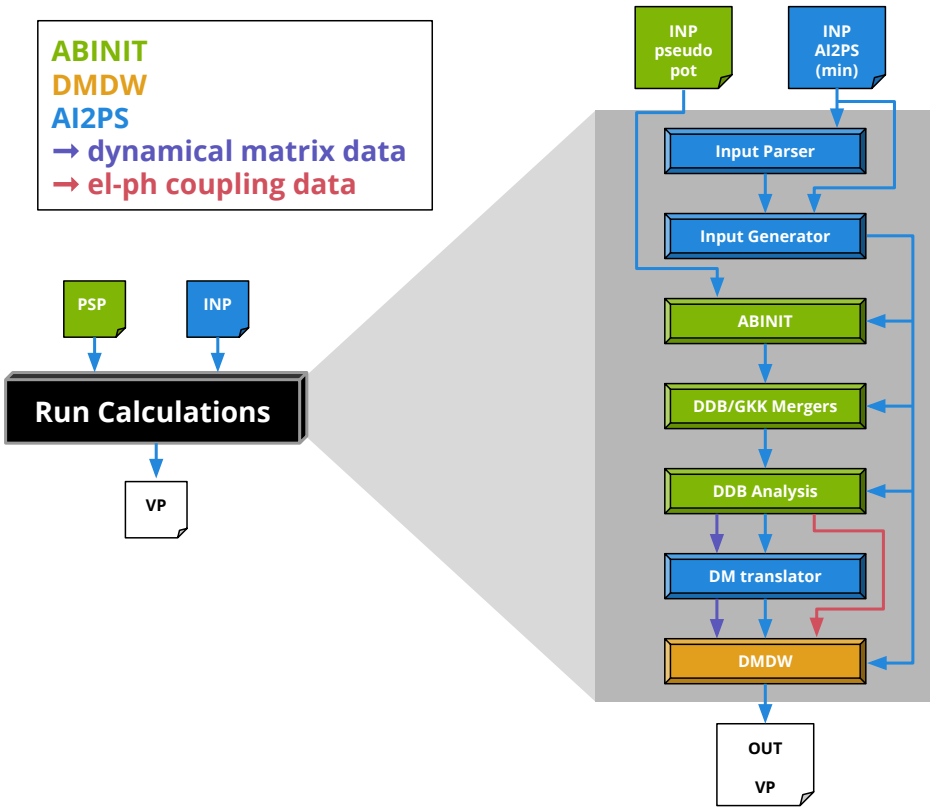

Figure 3.1: An illustration of the workflow necessary for generating vibrational properties using ABINIT as the DFT engine and DMDW as the analysis backend. The automated input generator, fed by one minimal input file, and dynamical matrix translator significantly decrease the workload for the user. The only additional input from the user is the pseudopotential files used by DFT engine. AI2PS calculations are run with a one–line command, which is explained in further detail in Appendix A.

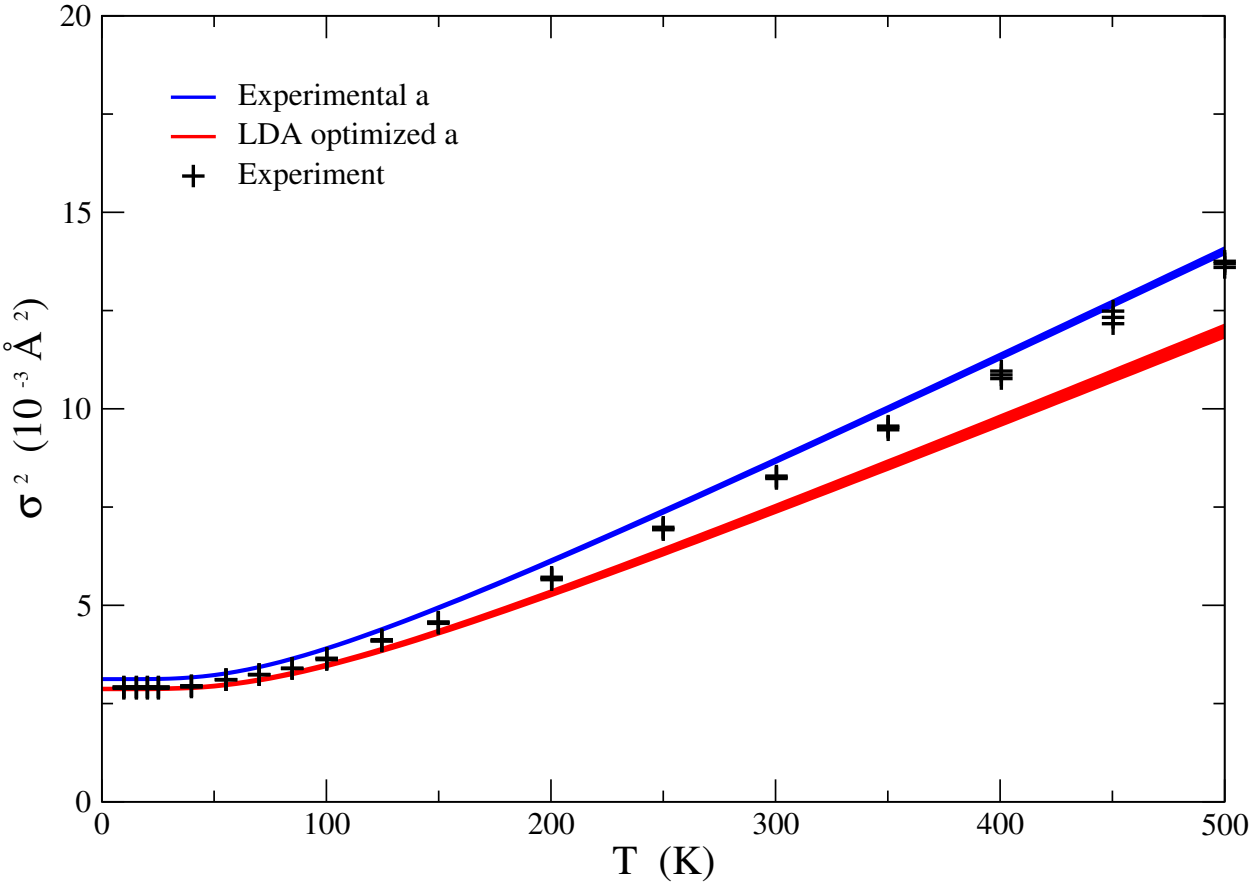

Figure 3.2: Example $\sigma^2(T)$ calculation for bulk Cu nearest neighbors using ABINIT with the LDA function and both the experimental lattice constant $a = 6.82$ Bohr (blue) and the lattice constant for the LDA–optimized cell $a = 6.73339$ Bohr(red) to demonstrate the sensitivity to a. Experimental results, shifted to match the LDA at $T = 0$, are included for comparison. [47]

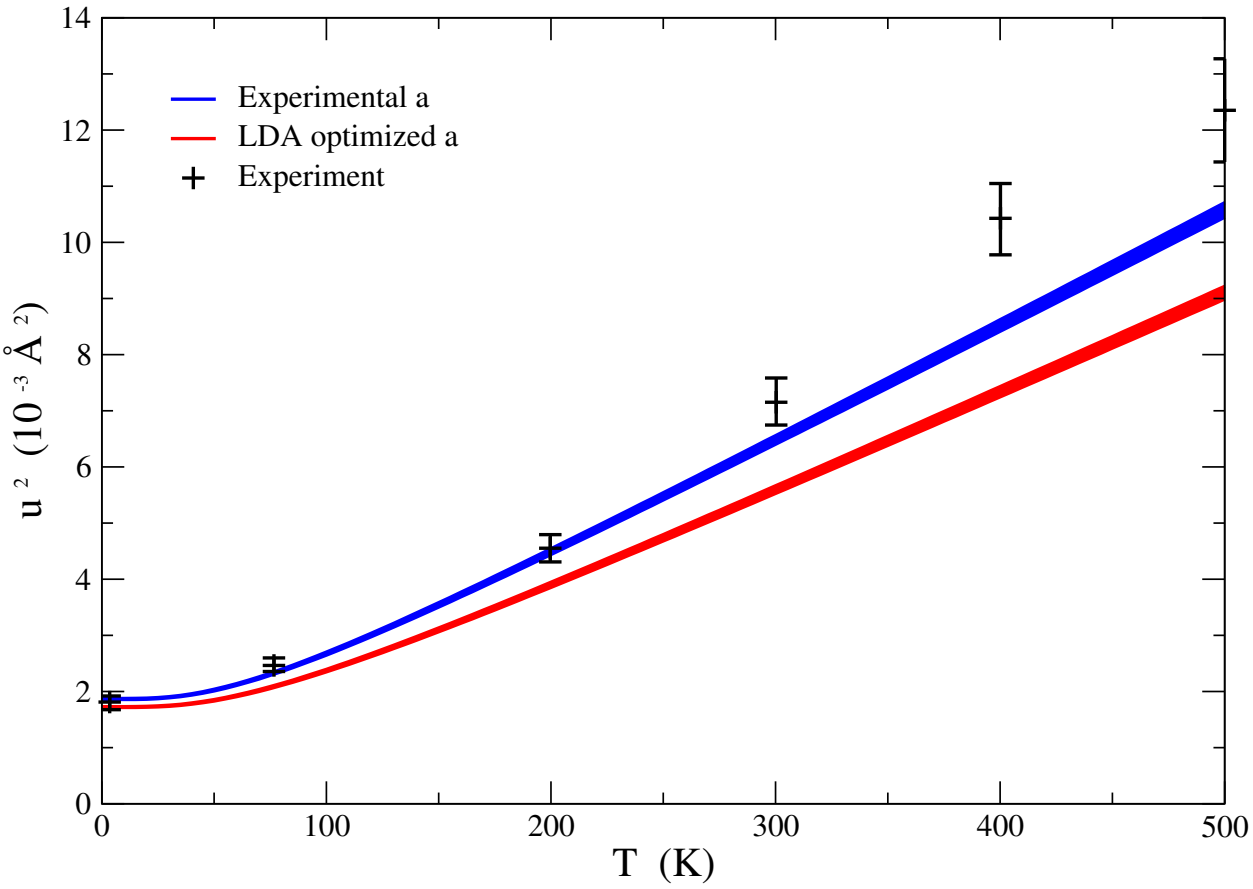

Figure 3.3: Example $u^2(T)$ calculation for bulk Cu using ABINIT with the LDA function and both the experimental lattice constant $a = 6.82$ Bohr (blue) and the lattice constant for the LDA–optimized cell $a = 6.73339$ Bohr(red) to demonstrate the sensitivity to a. Experimental results are included for comparison. [49]

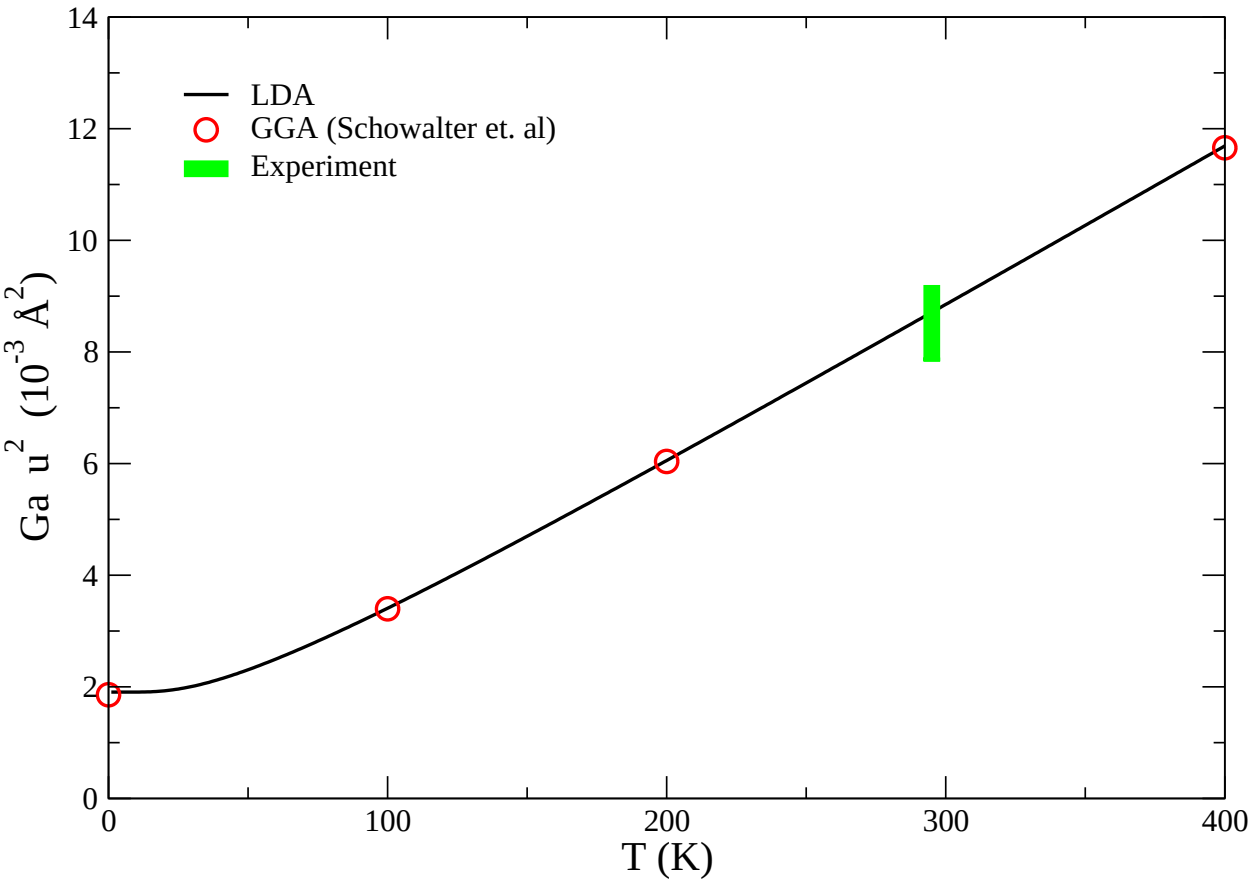

Figure 3.4: Example $u^2(T)$ calculation for Ga in bulk GaAs using ABINIT with the LDA functional. GGA calculations [50] and range of experimental results [51–54] for comparison.

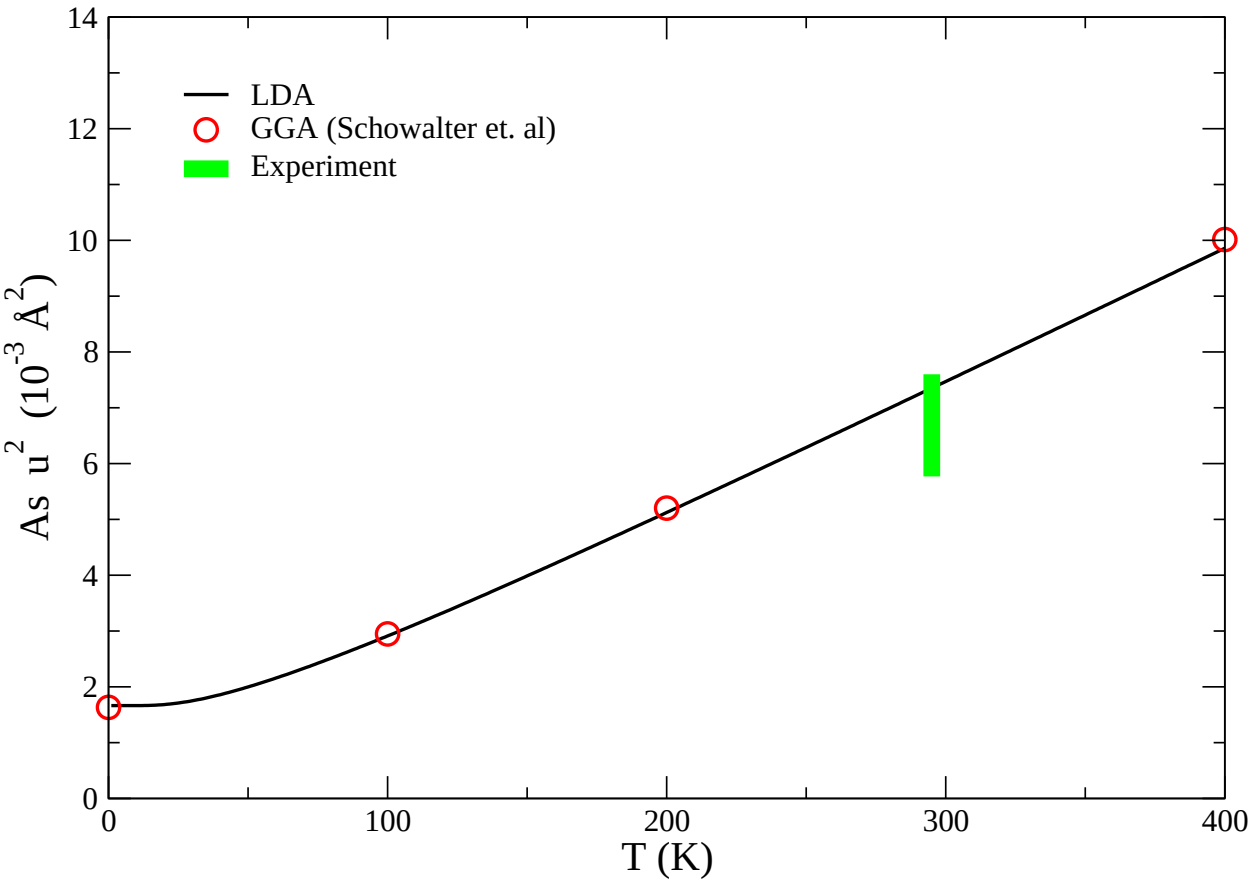

Figure 3.5: Example $u^2(T)$ calculation for As in bulk GaAs using ABINIT with the LDA functional. GGA calculations [50] and range of experimental results [51–54] for comparison.

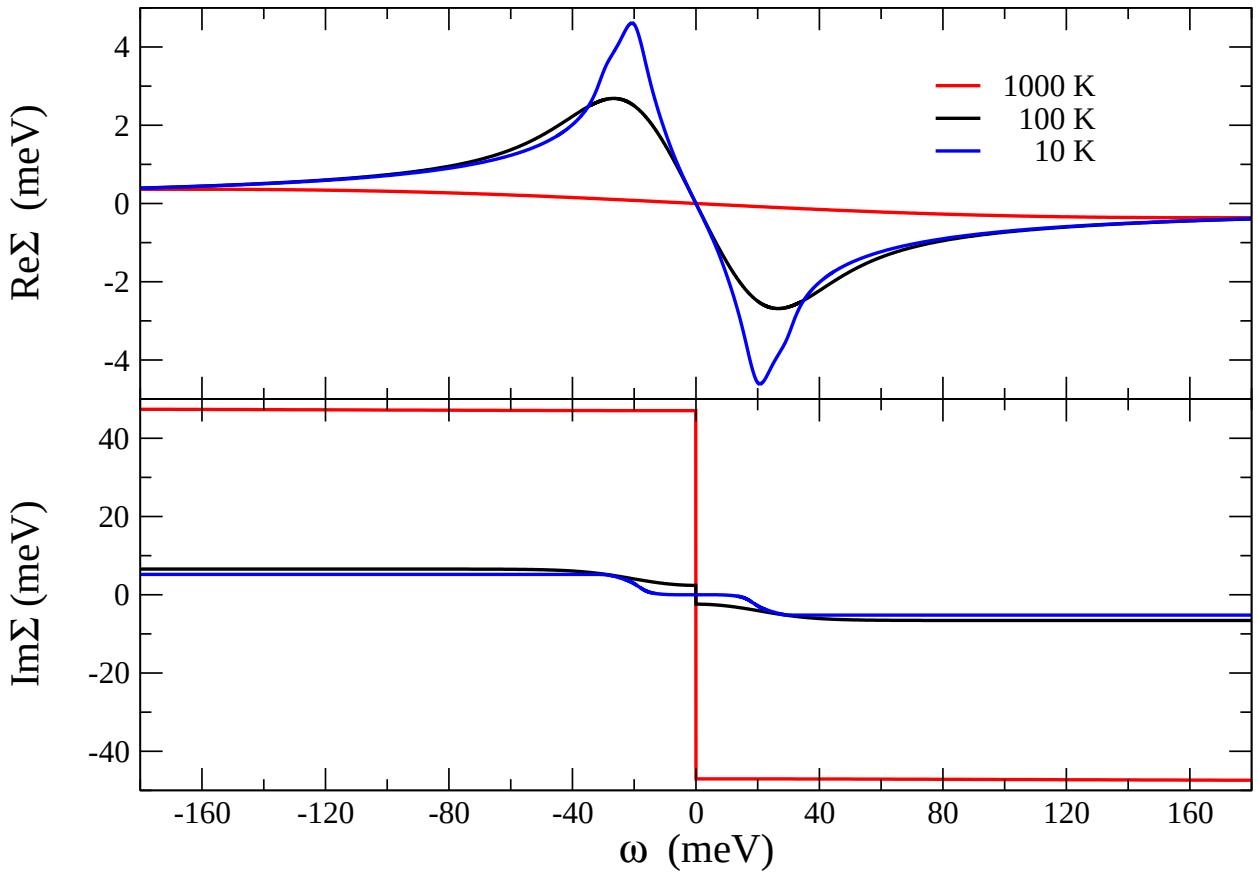

Figure 3.6: Example electron self–energy calculation for bulk Cu using ABINIT with the LDA functional.

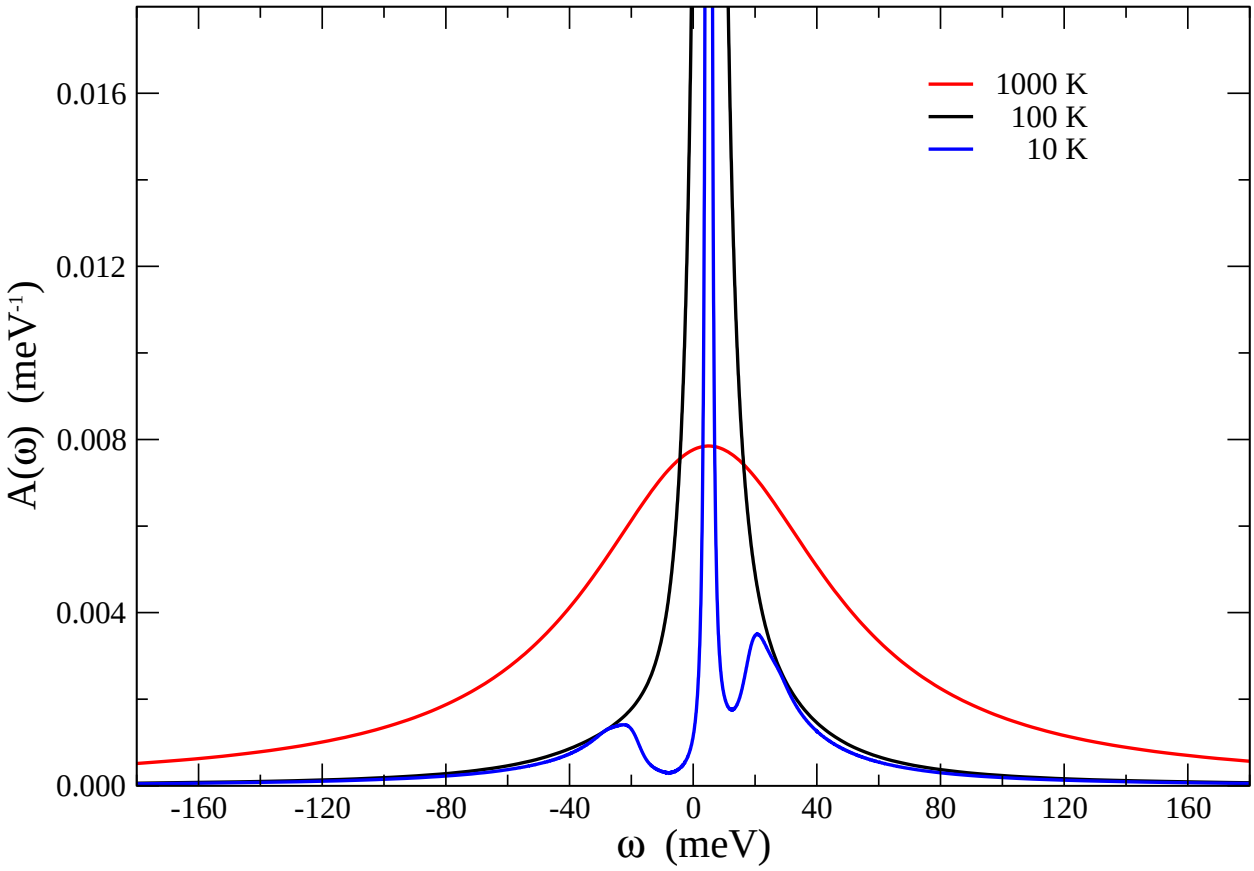

Figure 3.7: Example electron spectral function calculation for bulk Cu using ABINIT with the LDA functional.

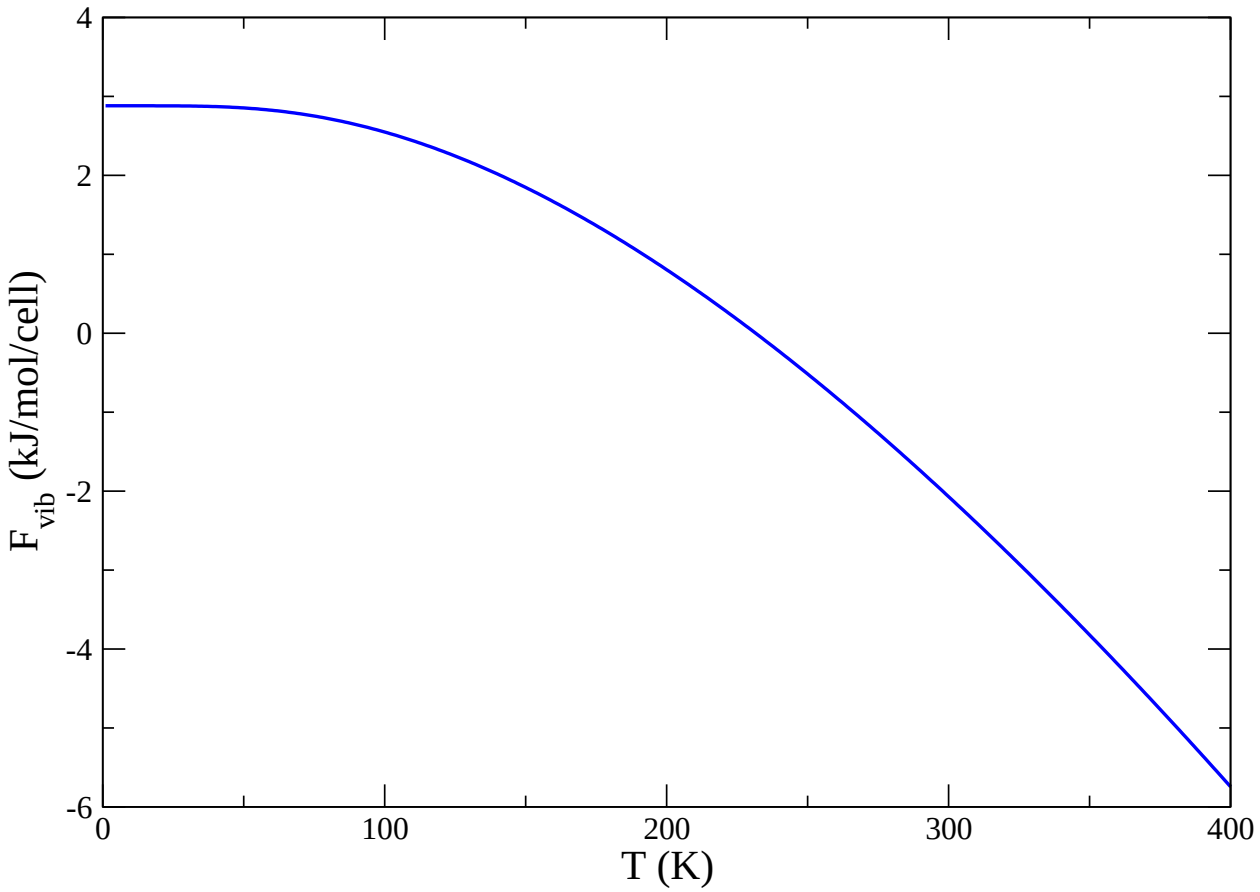

Figure 3.8: Vibrational free energy for bulk Cu using ABINIT with the LDA functional.

Chapter 4

CUMULANT EXPANSION FOR PHONON CONTRIBUTIONS TO THE ELECTRON SPECTRAL FUNCTION

Originally published as Phys. Rev. B ***90****, 195135 [55],*
by S. M. Story, J. J. Kas, F. D. Vila, M. J. Verstraete, and J. J. Rehr.

We describe an approach for calculations of phonon contributions to the electron spectral function, including both quasiparticle properties and satellites. The method is based on a cumulant expansion for the retarded one-electron Green's function and a many-pole model for the electron self-energy. Pole models are also used for the phonon density of states and the Eliashberg functions. Our calculations incorporate *ab initio* dynamical matrices and electron-phonon couplings from the density functional theory. Illustrative results are presented for several elemental metals and for Einstein and Debye models with a range of coupling constants. These are compared with experiment and other theoretical models. Estimates of corrections to Migdal's theorem are obtained by comparing with leading order contributions to the self-energy, and are found to be significant only for large electron-phonon couplings and low temperatures.

4.1 Introduction

To first approximation, electronic and vibrational properties can be treated separately in condensed matter due to the large mass ratio between electrons and ions, e.g., within the Born-Oppenheimer approximation. However, corrections to this approximation, which depend on the strength of the electron-phonon interaction, are of considerable importance both theoretically and experimentally. Here we investigate the effects of electron-phonon inter-

actions on the quasiparticle properties and inelastic losses of electrons due to coupling to phonons. Due to such interactions, the electron energy levels ε_k are not sharply defined, but have finite lifetimes characterized by the electron self-energy Σ, which lead to broadening of the associated spectral function. In addition, phonon-excitations give rise to inelastic losses. Such effects are visible experimentally, e.g., in high resolution ARPES spectra at low temperatures. [56]

In general, the electron spectral function at energies near the Fermi-level is dominated by a sharp quasiparticle peak, but it can also exhibit satellites due to phonon excitations. According to Migdal's theorem, [30] only the leading order electron-phonon interaction contributions to the self-energy are important, due to the large mass ratio between electrons and nuclei. In that case, the electron self-energy can be approximated by the simplest diagram, and vertex corrections can be neglected. We will refer to this diagram as the Migdal approximation (MA). The Migdal approximation (MA) is analogous to the GW approximation of Hedin [31] for electrons coupled to plasmons, where G is the electron Green's function and W the screened Coulomb interaction, by replacing W with the phonon propagator D. This approximation has been investigated in detail [28, 57–60] and extended to finite temperature, e.g., by Allen. [26] The MA leads to a spectral function with a quasiparticle peak and two satellite features originating from single-boson excitations, one on each side of the quasiparticle peak. However, this leading order approximation is generally unsatisfactory as the satellite peaks typically appear at the wrong energies and with the wrong intensities compared to experiment. Moreover, systems of electrons coupled to neutral bosonic excitations typically exhibit multiple satellites. [61,62] Thus, it is of interest to investigate possible corrections to Migdal's theorem, i.e., the effects of higher order terms in an expansion in powers of the electron-phonon coupling. [63] One approach to this end is to investigate contributions to the self-energy from the vertex function Γ, as in the formal identity $\Sigma = iGD\Gamma$. However, direct calculations of Γ have been formidably challenging, and there has been little progress along these lines. An attractive alternative that overcomes some of the above the shortcomings is provided by the cumulant expansion, [32,63,64] which is an exponential rep-

resentation of the electron Green's function in the time domain. The cumulant expansion is exact for the case of a deep core-level coupled to bosons, and generalizations have been developed for valence electrons coupled to plasmons. [65,66] The approach has been applied with considerable success in cases ranging from multiple plasmon satellites in photoemission [67] to dynamical mean field theory. [68] Nevertheless, the conventional approach based on the time-ordered Green's function is only strictly applicable for the hole- *or* particle-branch of the spectral function, depending on whether the state is above or below the Fermi level. This limitation is particularly problematic in systems with particle-hole symmetry, such as electrons coupled to phonons. To overcome this difficulty, we utilize here the recently developed retarded cumulant (RC) approach, which is based on a particle/hole cumulant and a *retarded* Green's function formalism. [32] A further goal of the present work is to develop a practical approach for calculations of phonon contributions to properties of condensed matter.

The remainder of this paper is organized as follows. In Sec. 4.2, we describe the retarded cumulant expansion method and many-pole model self-energy used to calculate phonon contributions to the electron spectral function. Sec. 4.3 gives details on how this method is implemented computationally with our workflow tool AI2PS. Finally, our results are presented in Sec. 4.4, and Sec. 4.5 contains a summary and conclusions.

4.2 Theory and Methodology

In this section, we briefly summarize the MA and RC methods used in this work for calculations of the electron spectral function in systems linearly coupled to phonons. As usual, the Hamiltonian for the electron-phonon system can be represented as

$$H = \sum_k \varepsilon_k^0 c_k^\dagger c_k + \sum_q \omega_q a_q^\dagger a_q + \sum_{kk'q} V_{kk'}^q (a_q + a_q^\dagger) c_k^\dagger c_{k'}, \tag{4.1}$$

where k denotes the electron levels and q the phonon modes with bare energies ε_k^0 and ω_q respectively, $V_{kk'}^q$ are the electron-phonon matrix elements, and c_k ($c_k^\dagger$) and a_q ($a_q^\dagger$) are the electron and phonon destruction (creation) operators. In this paper, we use atomic units $e = \hbar = m = 1$ and $k_B = 0.08617$ meV/K. At low temperatures, the electrons are nearly

degenerate with Fermi energy ε_F and $\omega_q \ll \varepsilon_F \ll \omega_p$, where ω_p is the dominant plasmon excitation energy (~ 10 eV) and ε_F is taken relative to the bottom of the band. Thus, for our purposes here, the density of electron states near ε_F is replaced by a constant, which we assume is non-vanishing. The generalization to insulators or molecular systems with discrete spectra near ε_F is straightforward, but will not be treated here.

4.2.1 MA spectral function

Briefly, the Migdal approximation for the self-energy is given by $\Sigma = iGD$, where G is the one-electron Green's function and D is the phonon propagator. Within MA, the usual strategy is to calculate the spectral function $A_k(\omega)$ from the imaginary part of the one-electron Green's function in frequency space, [28]

$$\begin{aligned} G_k(\omega) &= \frac{1}{\omega - \varepsilon_k^0 - \Sigma_k(\omega)} \\ A_k(\omega) &= \frac{1}{\pi} |\mathrm{Im}\, G_k(\omega)| \\ &= \frac{1}{\pi} \frac{|\mathrm{Im}\, \Sigma_k(\omega)|}{|\omega - \varepsilon_k^0 - \mathrm{Re}\, \Sigma_k(\omega)|^2 + |\mathrm{Im}\, \Sigma_k(\omega)|^2}. \end{aligned} \tag{4.2}$$

The spectral function is comprised of two main features—a dominant quasiparticle peak at $\omega = \varepsilon_k = \varepsilon_k^0 + \Sigma_k$ with width $\mathrm{Im}\, \Sigma_k$ and phonon satellites at $\omega = \varepsilon_F \pm \omega_q$, consistent with Ref. [28]. Other physical properties such as the quasiparticle lifetime and energy levels can be obtained from the properties of $A_k(\omega)$ and $\Sigma_k(\omega)$.

4.2.2 RC spectral function

As noted in the introduction, the conventional time-ordered cumulant expansion must be generalized to treat cases with particle-hole symmetry, such as phonon excitations in metals. [63] Our treatment is based on the RC formalism which is discussed in detail by Kas et al. [32] For a degenerate Fermi system in the absence of plasmons, the RC representation of

the retarded one-particle Green's function is

$$\begin{aligned} G_k^R(t) &= G_k^{0,R}(t) e^{C_k^R(t)} \\ G_k^{0,R}(t) &= -i\, e^{-i\varepsilon_k^0 t}\theta(t), \end{aligned} \tag{4.3}$$

where $C_k^R(t)$ is the cumulant as described below. Formally, the spectral function is obtained from the Fourier transform of $G_k^R(t)$

$$A_k(\omega) = \mathrm{Im} \int_{-\infty}^{\infty} \frac{dt}{\pi}\, e^{i\omega t} G_k^R(t). \tag{4.4}$$

The retarded particle/hole cumulant $C_k^R(t)$ is then approximated by the second order (in electron-phonon coupling) cumulant diagram [32]

$$\begin{aligned} C_k^R(t) &\approx C_{2,k}^R(t) \\ &= i e^{i\varepsilon_k^0 t} \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} e^{-i\omega t} \left[G_k^{0,R}(\omega)\right]^2 \Sigma_k^R(\omega). \end{aligned} \tag{4.5}$$

This diagram is conveniently evaluated in frequency space [32] and can be expressed in terms of the imaginary part of the G^0D^0 boson excitation spectrum $\beta_k(\omega)$ as

$$C_k^R(t) = \int_{-\infty}^{\infty} d\omega\, \beta_k(\omega)\, \frac{e^{i\omega t} - i\omega t - 1}{\omega^2}, \tag{4.6}$$

where $\beta_k(\omega)$ is obtained from the MA self-energy

$$\beta_k(\omega) = \frac{1}{\pi} \left|\mathrm{Im}\, \Sigma_k\left(\omega + \varepsilon_k^0\right)\right|. \tag{4.7}$$

Consequently, the ingredients in the RC are similar to those in RC spectral function is no more difficult to calculate than the MA spectral function. Since we approximate the non-interacting Green's function with that of a homogeneous electron gas, material-specific band structure will not be reflected in the resulting spectral functions. In contrast to the conventional time-ordered cumulant expansion, which only contains frequencies within the particle- or hole branches, the retarded cumulant in Eq. (4.6) contains *all* frequencies, and explicitly builds in the particle-hole symmetry desired for phonons. Also, due to the behavior of the self-energy $\Sigma_k(\omega)$, (Fig. 4.1), multiple phonon satellites may exist with the cumulant

expansion, as peaks at integral multiples of ω_E on both sides of the Fermi energy ε_F. This is in contrast to the case with plasmons, where the satellites appear at multiples of ω_p from the quasiparticle peak at ε_k.

4.2.3 Many-pole self-energy

Above we gave the formulae relating the RC and MA spectral functions to the MA phonon self-energy. We now focus on a self-energy model appropriate for phonons. Here we have adapted the finite-temperature Einstein model for phonons, [9, 26, 27] where the self-energy is represented as a sum over Einstein modes. For a single mode with Einstein frequency ω', the MA self-energy at finite temperature T (with unit coupling) is given by [26, 27]

$$\Sigma^{\mathrm{E}}(\omega, \omega', T) = -i\pi \left[n(\omega') + \frac{1}{2} \right] + \frac{1}{2}\Psi\left(\frac{1}{2} + i\frac{\omega' - \omega}{2\pi T} \right) - \frac{1}{2}\Psi\left(\frac{1}{2} - i\frac{\omega' + \omega}{2\pi T} \right), \qquad (4.8)$$

where $n(\omega)$ is the Bose-Einstein distribution and $\Psi(z)$ is the digamma function. The electron-phonon coupling constants in the model are represented in terms of the Eliashberg function $\alpha^2 F_k$. The self-energy is then [9, 26]

$$\Sigma_k(\omega, T) = \int d\omega' \, 2\Sigma^{\mathrm{E}}(\omega, \omega', T) \, \alpha^2 F_k(\omega') \,. \qquad (4.9)$$

We emphasize that the form of the self-energy in Eq. (4.9) is strictly appropriate only for cases where the band width of electron states near the Fermi energy is large compared to characteristic phonon energies ω, and will not work for sharp band features. This is the case for valence states in metals and in many semi-metals, semiconductors, and insulators, but becomes questionable in the case of small molecules and core level states. Thus in the present work, we focus only on a selection of metallic systems covering a range of electron-phonon couplings. As an example, Fig. 4.1 shows the real and imaginary parts of the self-energy calculated using Eq. (4.9) when coupling to a single Einstein mode, i.e., an Einstein model for the phonon spectrum in Cu. We also note that the approximation in Eq. (4.8) neglects Debye-Waller corrections to the self-energy discussed for example, by Allen and Heine and others [69–71]. This correction is found to give temperature dependent shifts in the quasi-particle energies and a broadening of the energy bands. On the other hand, this term gives no

contribution to the excitation spectrum and hence the structure of the spectral function [69] which is the main topic of this work.

Generally, the Eliashberg function depends on both k and k' through the electron-phonon matrix elements $g^q_{kk'}$. [21–24] However, since the phonon contributions to the spectral function involve energies very close to ε_F, it is sufficient for our purposes here to use the Eliashberg function averaged over the Fermi surface

$$\begin{aligned} \alpha^2 F(\omega) &= \frac{1}{2\pi N(\varepsilon_F)} \sum_q \sum_{\varepsilon_k, \varepsilon_{k'} \approx \varepsilon_F} |g^q_{kk'}|^2 \, \delta(\omega - \omega_q) \\ g^q_{kk'} &= 2 \sum \langle \psi_{k'} | \, \delta V^q \, | \psi_k \rangle \, , \end{aligned} \tag{4.10}$$

where $N(\varepsilon_F)$ is the bare density of states at the Fermi level, δV is the change in potential due to a change in the nuclear parameters, and the sum over spin states for non-magnetic systems considered here gives a factor of 2. Typically, $\alpha^2 F(\omega)$ is rather similar to the total phonon density of states (PDOS) $F(\omega)$ of the system (See Fig. 4.2), allowing us to express it as a smooth coupling $\alpha^2(\omega)$ multiplying the PDOS. Moreover, it is convenient to use a many-pole model for the PDOS, [28] analogous to the plasmon-pole self-energy model of Hedin and Lundqvist. [72–75] The integration over the phonon frequencies ω' in Eq. (4.9) can then be replaced by a discrete sum over a sufficiently large number of poles without significant loss of accuracy. To this end, an efficient many-pole Lanczos representation of the PDOS has been developed, [14]

$$F^{\mathrm{MP}}(\omega) = \sum_i F_i \delta(\omega - \omega_i). \tag{4.11}$$

Thus, a many-pole representation of the Eliashberg function $\alpha^2 F^{\mathrm{MP}}$ can be constructed similarly,

$$\alpha^2 F^{\mathrm{MP}}(\omega) = \sum_i \alpha^2(\omega) \, F_i \delta(\omega - \omega_i) \, , \tag{4.12}$$

where

$$\alpha^2(\omega) = \alpha^2 F(\omega) / F(\omega) \tag{4.13}$$

A 16-pole representation of the copper Eliashberg function is shown in Fig. 4.2. Finally, an effective or mean electron-phonon coupling constant λ can be defined, which is related to the first inverse frequency moment of the Eliashberg function [25]

$$\lambda = 2\int_0^\infty \frac{d\omega}{\omega}\alpha^2F(\omega) \approx 2\sum_i \frac{\alpha_i^2 F_i}{\omega_i}. \tag{4.14}$$

This quantity provides a dimensionless characterization of the strength of electron-phonon coupling in a given material.

4.3 Implementation

The calculations of phonon properties presented here were carried out using AI2PS (*ab initio* DFT to Phonon Spectra), [46] a workflow tool we have developed that links density functional theory electronic structure codes, ABINIT in this case, [42, 43] to the vibrational properties module of real-space Green's function code FEFF9. [45] AI2PS can be used to calculate phonon properties such as Debye-Waller factors in x-ray spectra. The modular interface automatically coordinates the desired workflow. Briefly, for our purposes here, AI2PS uses ABINIT to generate a set of real-space symmetry-inequivalent blocks of the lattice dynamical matrix (DM), which are used to calculate the many-pole PDOS $F^{\mathrm{MP}}(\omega)$. [14] The code ABINIT also yields both $F(\omega)$ and $\alpha^2F(\omega)$, which are used to calculate the couplings $\alpha^2(\omega)$ using Eq. (4.13). Since Eq. (4.9) is restricted to energies near the Fermi level, this presently excludes any k-dependent features in the spectral functions presented in the current study. The ABINIT calculations used Troullier-Martins LDA pseudopotentials provided by ABINIT, [76] and an energy cutoff of 50 Hartrees; for convergence of $\alpha^2F(\omega)$, a 32×32×32 Monkhorst-Pack k-point grid was found to be necessary. For the metallic systems discussed here, the occupation numbers were smeared with the Methfessel and Paxton scheme [77] with a broadening parameter of 0.025. To simplify the numerical calculations by removing self-energy shifts, the RC and MA spectral function expressions were expressed in terms of the quasiparticle energy $\varepsilon_k = \varepsilon_k^0 + \mathrm{Re}\Sigma_k(\varepsilon_k^0)$ as opposed to ε_k^0 (see Eqs. (4.2), (4.3),(4.7)).

4.4 Results and Discussion

In this section, we present illustrative results for several elemental metals and for Einstein and Debye models with a range of electron-phonon couplings over a range of temperatures and energies for both the RC and MA methods.

4.4.1 Einstein model

As a first example, we consider the Einstein model self-energy Σ^{E}, i.e., using the single-pole (zeroth-order Lanczos) approximation for the Eliashberg function,

$$\alpha^2 F(\omega) \to \alpha_E^2 \delta(\omega - \omega_E), \tag{4.15}$$

where ω_E is the Einstein frequency and $\alpha_E^2 = \alpha^2(\omega_E)$. For realistic systems, the value of ω_E is taken to be the centroid of the calculated PDOS. As an example, we present results for an Einstein model with $\omega_E = 21.6$ meV (251 K), representative of Cu metal, in Fig. 4.3. Note that phonon satellites in the spectral function are visible only for quasiparticle energies small compared to phonon frequencies $\varepsilon_k - \varepsilon_F < \omega_E$, and very low temperatures (~ 10 K), as seen in the top two panels of Fig. 4.3. For the Einstein model, the mean coupling constant λ is simply $2\alpha_E^2/\omega_E$, so we artificially ramp up the coupling by manually setting the value of α_E^2. Typically, metals have coupling constants λ that range from roughly 0.1 to 1.7, [24] so we will focus on that range. The satellites become larger as λ is increased (third panel), and for $\lambda \approx 1.6$, a weak second phonon satellite becomes apparent at $\omega = \varepsilon_F \pm 2\omega_E$. The relative weakness of the second satellite even at $\lambda = 1.6$ suggests Migdal's theorem is valid to high accuracy for typical metals, apart from corrections close to the Fermi energy at very low temperatures.

4.4.2 Debye model

For comparison, we show similar results using the Debye model PDOS converted to a many-pole form in Fig. 4.4, with quantities expressed in terms of the Debye temperature for copper

$\Theta_D = 315$ K $= 27.1$ meV $= \omega_D$. Overall, the Debye model shows trends quite similar to the Einstein model. However, the phonon satellites are not as sharply peaked, and the satellites at $\varepsilon_F \pm 2\omega_D$ are indiscernible at the same scale for large couplings $\lambda \sim 1.6$. Note that artifacts of the many-pole model can be seen in the spectral functions as small peaks near the Fermi energy (third panel), though these effects are negligible compared to the phonon satellites.

4.4.3 Comparison of RC and MA

We note that the electron spectral function near the Fermi level ε_F is generally nearly symmetrical due to particle-hole symmetry, and is sensitive to phonon correlations beyond MA at strong electron-phonon coupling, as illustrated by the significant deviation of RC from MA seen in Fig. 4.5. Thus, it is useful to compare the RC and MA methods in this limit, especially since the differences characterize corrections to the Migdal approximation due to vertex effects. Fig. 4.5 shows that the two methods differ significantly at strong couplings *and* low temperatures compared to the Debye or Einstein temperature (see Table 4.1 for distribution of spectral weight). The RC method gives larger satellite weights, with a stronger first satellite peak closer to the expected $\varepsilon_F \pm \omega_E$ (see Inset to Fig. 4.5). However, the differences between the two methods diminish as the temperature is increased towards room temperature. Neither method shows a noticeable two phonon satellite peak.

4.4.4 Selected metals: Cu, Nb, Pb, Ta, and V

Next, we present results for the spectral function for a few elemental solids (Fig. 4.6) representative of a range of electron-phonon couplings (See Table 4.2). To obtain the full spectral function for these materials, we use the Lanczos many-pole representation of Eq. (4.12) as shown in Fig. 4.2 for copper. The results for these metals follow similar trends with the RC and MA methods that we saw with our results for the Einstein and Debye models. Copper, which has a relatively weak coupling ($\lambda \sim 0.1$), displays near agreement between the RC and MA methods. Tantalum and vanadium, on the other hand, have medium to strong couplings, respectively, and show significant differences between the two methods. Most

noticeably, for increasing quasiparticle energies, both the distribution of weight between the quasiparticle and satellites and the location of these peaks disagree significantly, possibly enough to be noticeable experimentally. However, these differences can only be seen at low temperatures (~ 50 K). Even with the strongest coupling, vanadium does not show multiple phonon satellites, indicating Migdal's theorem is valid to high accuracy for phonons in these materials.

4.4.5 Comparison with experiment

Evidence for electron-phonon effects in the spectral function have been measured in a number of cases. For instance, the value of the mean coupling constant λ can be obtained experimentally from the slope of the quasiparticle linewidth $\Gamma \sim 2\pi\lambda k_B T$ versus temperature. [9] Thus, calculations of quasiparticle linewidths characterize the phonon-contributions to the quasiparticle broadening. Our calculated quasiparticle peak FWHM (Fig. 4.7) are comparable to those measured experimentally. [82–84] As an example, we find a slope of 0.0680 meV/K for copper at large ε_k (top panel Fig. 4.7), corresponding to $\lambda = 0.126$. Due to the redistribution of spectral weight from the quasiparticle peak to the phonon satellites for $\varepsilon_k \sim \varepsilon_F$ (lower panel Fig. 4.7), the temperature trends are significantly steeper, resulting in inflated λ values. The λ taken from the large ε_k calculations are more consistent with experimental observations, and are listed in Table 4.2. The calculated λ for the metals using Eq. (4.14) are given as well in Table 4.2, along with several experimental results for comparison. Overall, there is decent agreement with experiment. The heavier metals show more discrepancy, which is likely an effect of the absence of spin-orbit coupling in our simulations. [85, 86]

4.5 Summary and Conclusions

We have implemented a retarded cumulant (RC) expansion approach to calculate phonon contributions to electron spectral function. This approach goes beyond the standard Migdal approximation (MA) to include effects of phonon excitation satellites in the electron spectral function. Our calculations show that the phonon-contribution to the quasiparticle peak is

linearly dependent on temperature. We verify that Migdal's theorem is generally satisfied for phonons to high accuracy. Thus the effects of vertex corrections leading to deviations between the MA and RC approaches and multiple satellites in the spectral function and are generally negligible except at very low T ($T \lesssim 50$ K) and very strong electron-phonon couplings ($\lambda \gtrsim 1$), and would require roughly meV resolution to discern experimentally. With an appropriate self-energy, the method presented here can also be extended to treat insulators and molecular systems.

We thank C. Draxl, L. Reining, P. B. Allen, G. Rignanese, X. Gonze, M. Bernardi, and K. Jorissen for useful discussions, and S. R. Williams and J. Vinson for assistance in code development. The ABINIT code is a common project of the Université Catholique de Louvain, Corning Incorporated and other contributors (See http://www.abinit.org). This work was supported in part by DOE grant DE–FG02–97ER45623.

4.6 Addendum

The following related material was not included for publication of the original article above.

As another application, the value of electron-phonon coupling $\lambda$is also related to the superconducting critical temperature T_c via the McMillan relation [21, 87]

$$\begin{aligned} T_c &= \frac{\omega_{\ln}}{1.20} \exp\left[-\frac{1.04\,(1+\lambda)}{\lambda - \mu^*\,(1+0.62\lambda)}\right] \\ \omega_{\ln} &\equiv \exp\left[\frac{2}{\lambda}\int_0^\infty d\omega \frac{\ln(\omega)}{\omega}\alpha^2 F(\omega)\right], \end{aligned} \tag{4.16}$$

where μ^* is the Coulomb pseudopotential, a fitting parameter typically $\sim$ 0.1-0.2. [88] Calculated transition temperatures for the five bulk materials studied above are given in Table 4.3, along with the values of μ^* used in the McMillan relation. The T_c calculated for (non–superconducting) copper, with $\lambda = 0.155$ and $\mu^* = 0.1$, is essentially zero as expected. The other four materials, with significantly stronger electron–phonon coupling, have finite T_c and our simulations yield values on the correct order of magnitude ($\sim$ 10 K), with the coupling constants calculated from the spectral function quasiparticle peak widths giving the best agreement with experiment (see Figure 4.8).

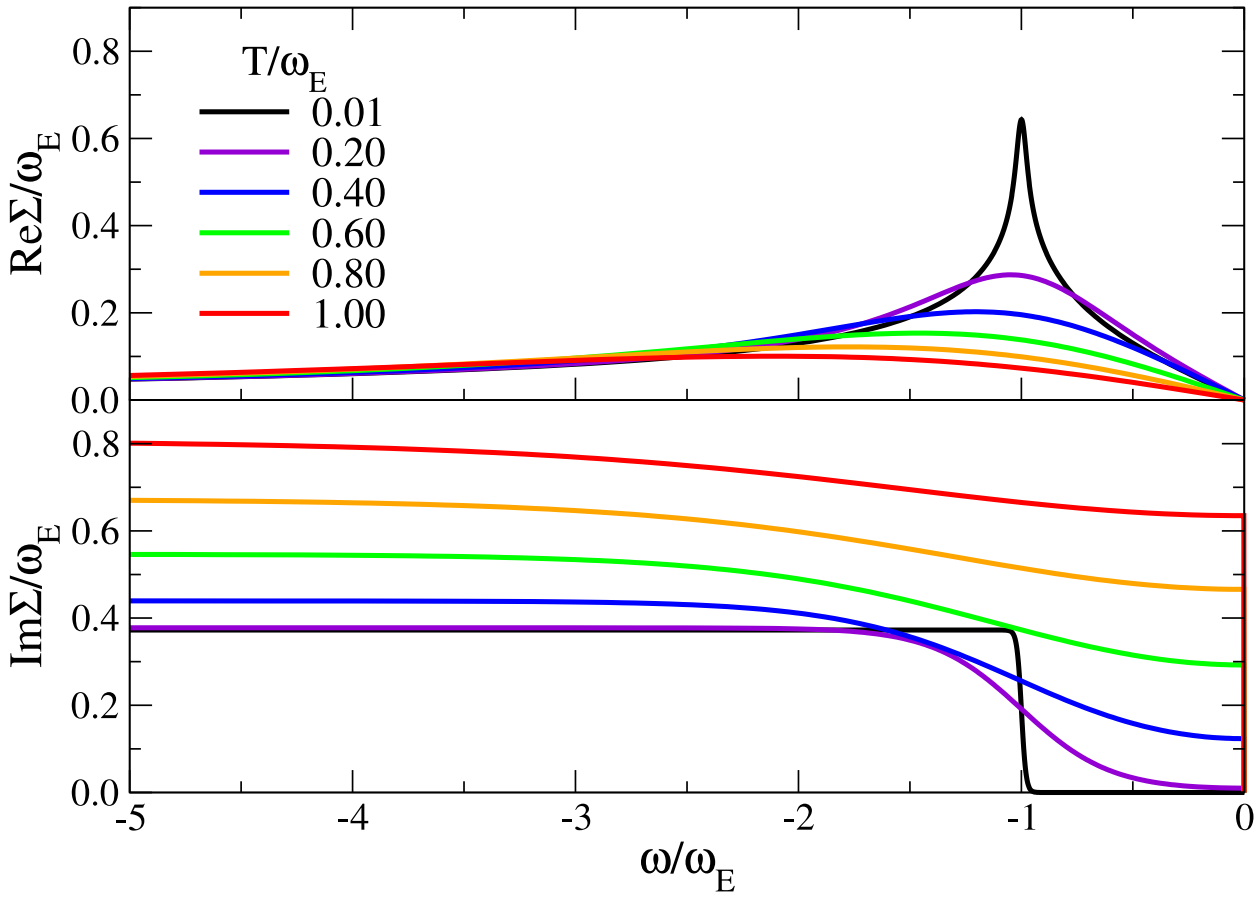

Figure 4.1: Real (top) and imaginary (bottom) parts of the self-energy $\Sigma_k(\omega)$ in Eq. (4.9) using the Einstein model for Cu, where $\omega_E = 21.6$ meV $= 251$ K (see text). Positive ω is not shown, as $\mathrm{Re}\,\Sigma$ and $\mathrm{Im}\,\Sigma$ and can be determined from the odd parity of $\Sigma_k(\omega)$.

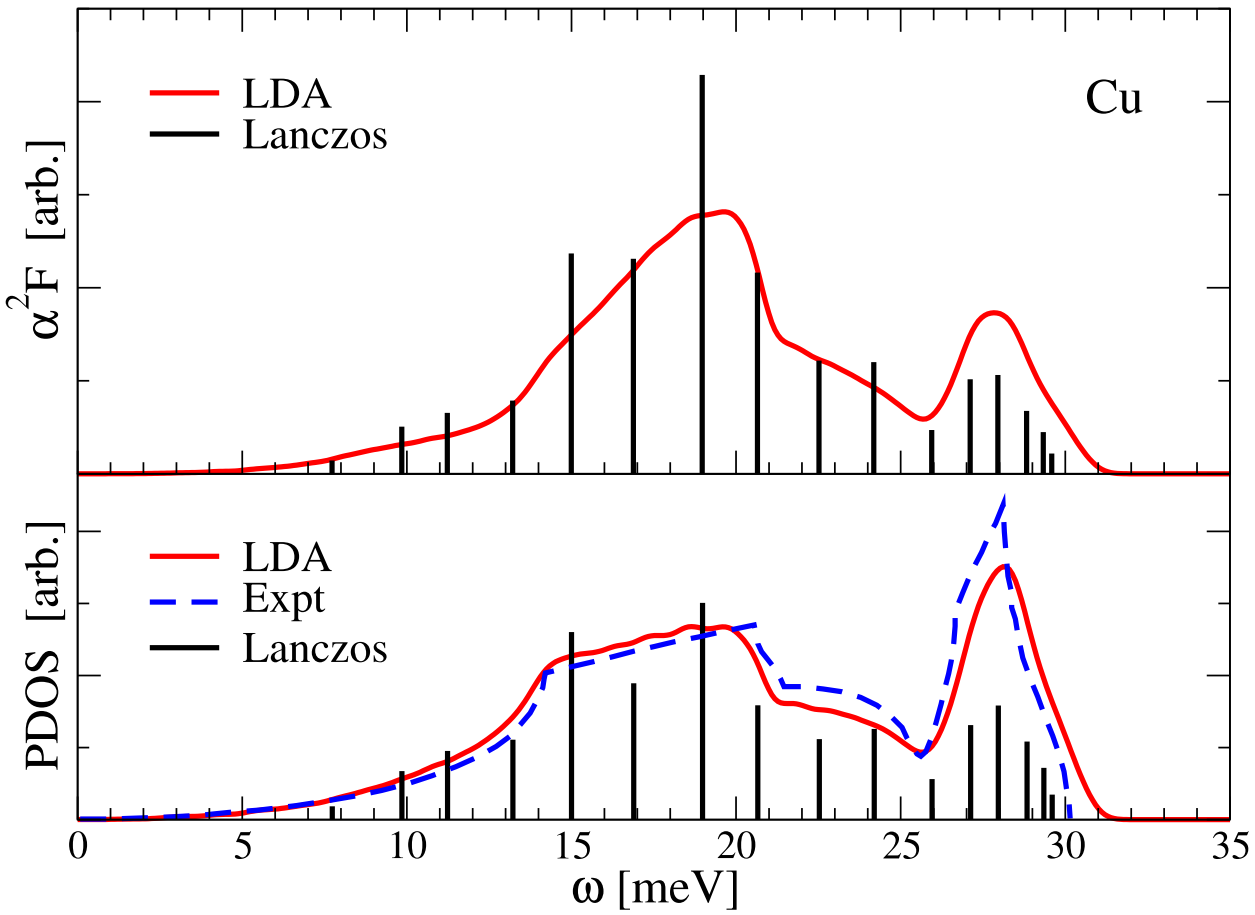

Figure 4.2: Eliashberg function (top) $\alpha^2F(\omega)$ and total density of modes (bottom) for Cu at the Fermi level $k = k_F$ obtained from ABINIT with our many-pole approximations α_i^2, F_i calculated by the Lanczos inversion tools in FEFF9 (see text). Experimental PDOS taken from Ref. [10]. Calculated frequencies have been scaled using an overall Grüneisen parameter to match the PDOS peak frequency with experiment.

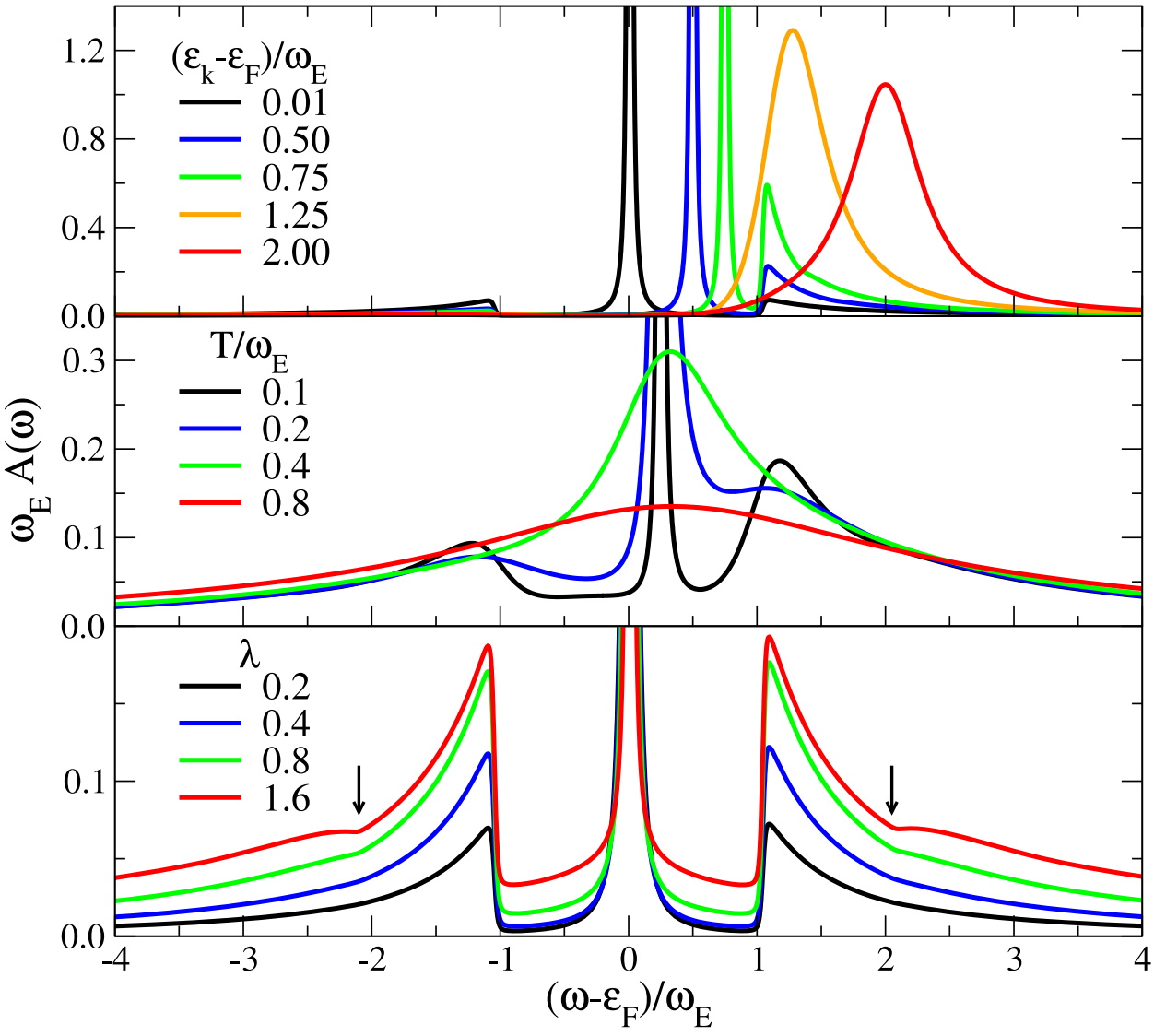

Figure 4.3: Spectral function for the Einstein model using the RC method, where ω_E is the Einstein energy. Top: varying quasiparticle energy for low temperature and weak coupling ($T = 0.01\ \omega_E$, $\lambda = 0.2$), middle: varying temperature near the Fermi energy and with medium coupling ($\varepsilon_k - \varepsilon_F = 0.25\ \omega_E$, $\lambda = 1.0$), bottom: varying electron-phonon coupling constant at low temperature near the Fermi energy ($\varepsilon_k - \varepsilon_F = T = 0.01\ \omega_E$).

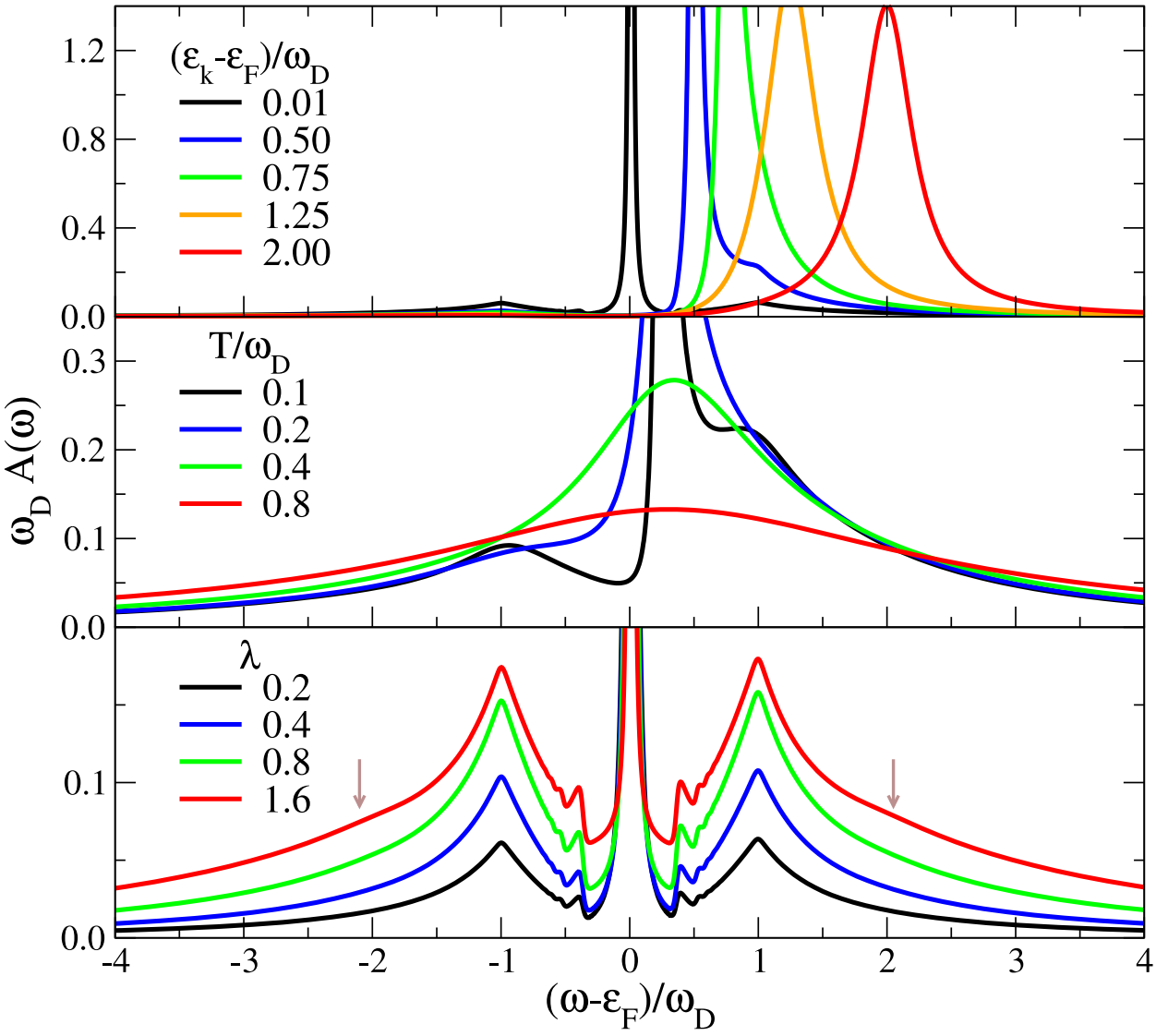

Figure 4.4: Spectral function for the Debye model using the RC method, where ω_D is energy corresponding to the Debye temperature. Top: varying quasiparticle energy for low temperature and weak coupling ($T = 0.01\ \omega_D$,$\lambda = 0.2$), middle: varying temperature near the Fermi energy and with medium coupling ($\varepsilon_k - \varepsilon_F = 0.25\ \omega_D$, $\lambda = 1.0$), bottom: varying electron-phonon coupling constant at low temperature near the Fermi energy ($\varepsilon_k - \varepsilon_F = T = 0.01\ \omega_D$).

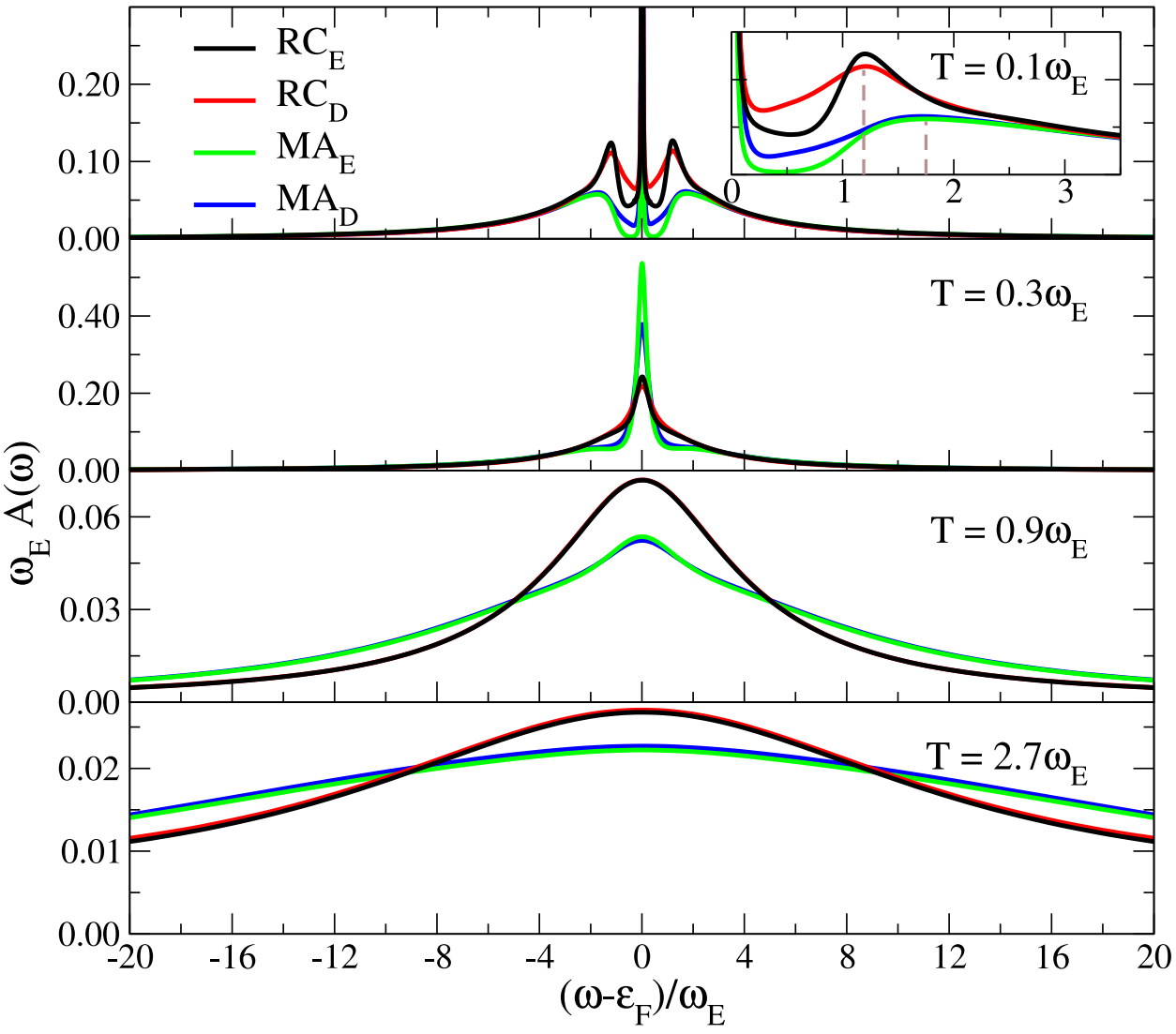

Figure 4.5: Comparison of spectral function from the RC and MA methods using the Einstein and Debye models at strong coupling (i.e., $\lambda = 1.6$) near the Fermi level ($\varepsilon_k = \varepsilon_F + 0.01\ \omega_E$) for various temperatures, where $\omega_D \approx 1.3\ \omega_E$. Inset: enlarged view of phonon satellites seen in the top panel. Note the satellite centroid for the MA method is further out.

Table 4.1: Comparison of weights for the quasiparticle peak, hole satellite, and particle satellite (Z_k, w_h, w_p respectively) using the RC and MA methods, done for the Einstein/Debye models at large coupling ($\lambda = 1.6$) and several elemental metals near the Fermi level at low temperature ($\varepsilon_k = \varepsilon_F + 0.01\ \omega_{E,\mathrm{Cu}} = 0.216$ meV, $T = 0.1\ \omega_{E,\mathrm{Cu}} = 25.1$ K).

		Z_k	w_h	w_p	λ
	Einstein	0.19	0.39	0.42	1.60
	Debye	0.18	0.39	0.43	1.60
	V	0.29	0.34	0.37	1.17
RC	Nb	0.31	0.33	0.36	1.08
	Pb	0.35	0.31	0.34	0.95
	Ta	0.37	0.30	0.33	0.91
	Cu	0.85	0.07	0.08	0.16
	Einstein	0.38	0.31	0.31	1.60
	Debye	0.37	0.31	0.32	1.60
	V	0.45	0.27	0.28	1.17
MA	Nb	0.46	0.27	0.27	1.08
	Pb	0.49	0.25	0.26	0.95
	Ta	0.50	0.24	0.26	0.91
	Cu	0.86	0.07	0.07	0.16

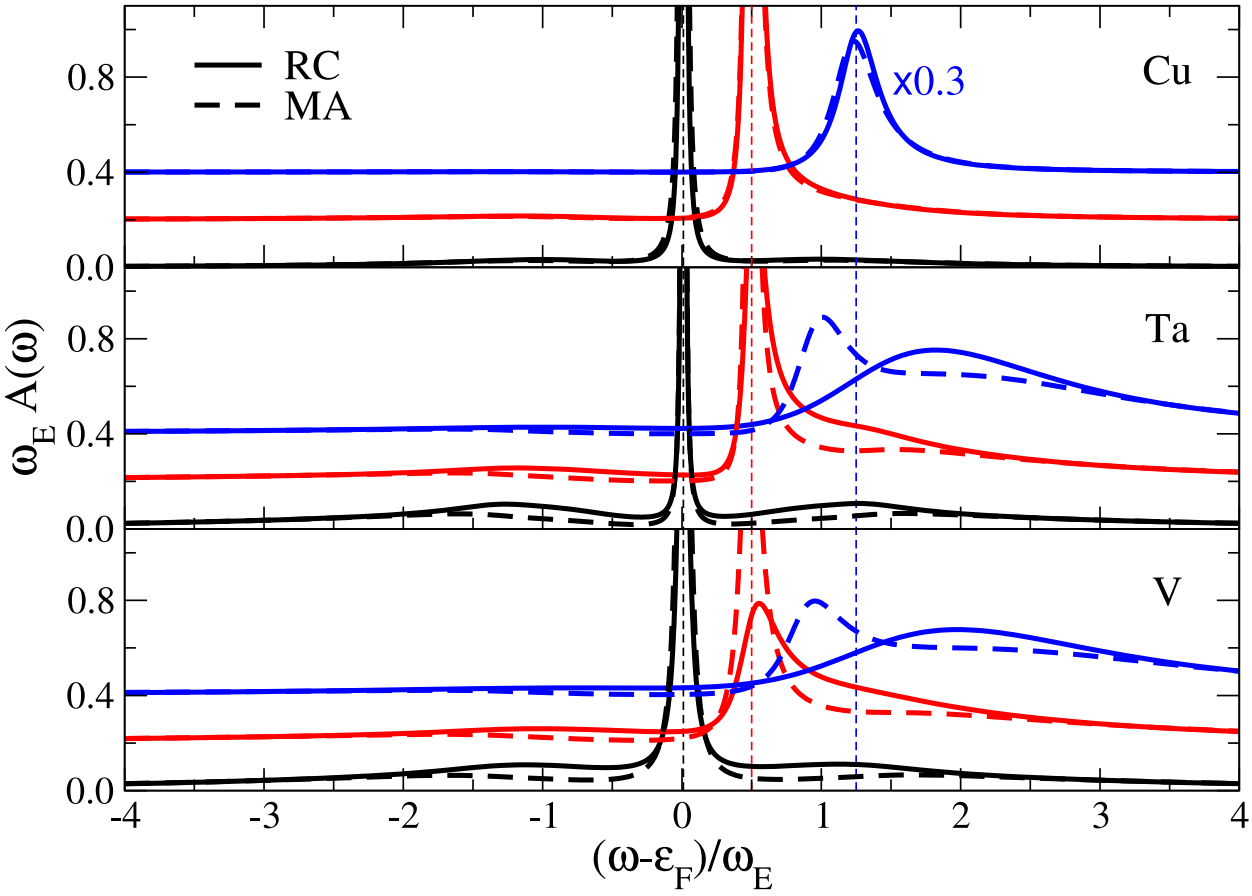

Figure 4.6: Comparing the spectral functions of the RC and MA methods for select metals at low temperature and three quasiparticle energies ε_k for $(\varepsilon_k - \varepsilon_F)/\omega_E = 0.01, 0.5, 1.25$ (bottom, middle, top vertically offset curves respectively in each panel, with arrows indicating corresponding location along the horizontal axis). For Cu, Ta, and V, $\omega_E = 21.6$, 15.0, 24.1 meV respectively and $T = 0.15\ \omega_E = 37.65$, 17.4, 42.0 K respectively. The spectral function for Cu with $\varepsilon_k - \varepsilon_F = 1.25\ \omega_E$ has been scaled vertically, as indicated.

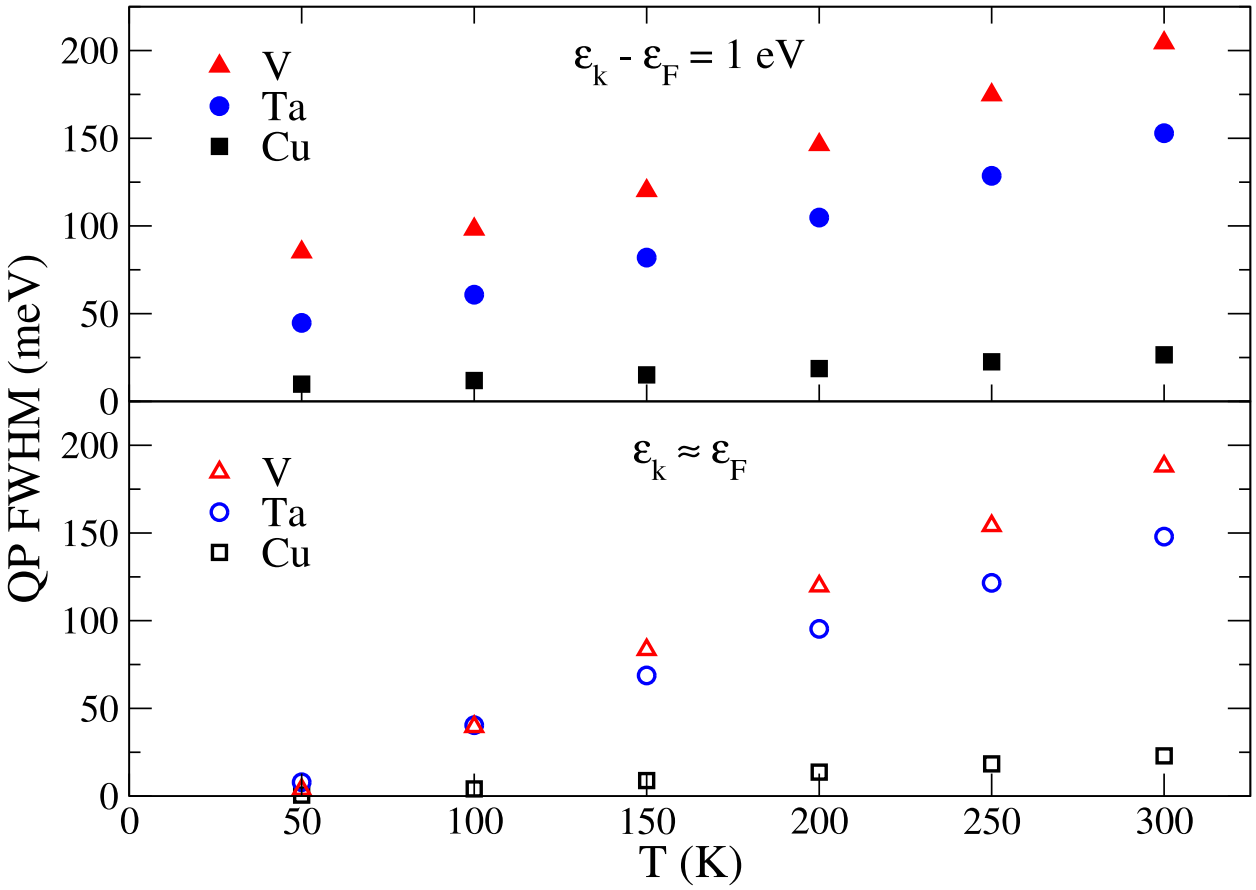

Figure 4.7: (color online) Quasiparticle peak width versus temperature at the Fermi energy $\varepsilon_k = \varepsilon_F$ (bottom), and at moderate quasiparticle energy $\varepsilon_k = \varepsilon_F + 1.0$ eV (top) for Cu, Ta, and V. The linear relationship, given by $\Gamma \sim 2\pi\lambda k_B T$, gives an estimate of the electron-phonon coupling strength λ. Table 4.2 shows λ derived from the $\varepsilon_k = \varepsilon_F + 1.0$ eV calculations, to limit any skewing of the quasiparticle widths due to phonon satellites.

Table 4.2: Calculated electron-phonon coupling constants using two methods—the inverse moment of the many-pole $\alpha^2F(\omega)$ (Eq. 4.14) and the temperature dependence of the quasiparticle linewidth taken from the spectral functions at large ε_k—and experimental results for comparison.

	λ^{MP}	λ^{Γ}	λ^{expt}		
V	1.174	0.899	0.82[b]	1.09[c]	0.80[d]
Nb	1.079	0.897	1.04[b]	1.06[c]	1.16[d]
Pb	0.946	0.955	1.55[b]	1.48[c]	1.45[d]
Ta	0.909	0.809	0.78[b]	0.87[c]	
Cu	0.155	0.126	0.10[a]	0.13[c]	0.08[d]

[a]Ref. [78] [b]Ref. [79] [c]Ref. [80] [d]Ref. [81]

Table 4.3: Calculated electron–phonon coupling constant and superconducting critical temperature for selected elemental metals using both methods given above for calculating the electron–phonon coupling constants λ, with experimental values for comparison.

	Cu	Ta	V	Nb	Pb
μ^*	0.10[a]	0.11[a]	0.12[a]	0.12[a]	0.10[a]
T_c^{MP} [K]	10^{-10}	9.0	19.9	14.4	5.0
T_c^{Γ} [K]	10^{-27}	4.2	4.0	4.6	5.3
T_c^{expt}[K]	–	4.5[b]	5.4[b]	9.5[b]	7.2[b]

[a]Ref. [88] [b]Ref. [89]

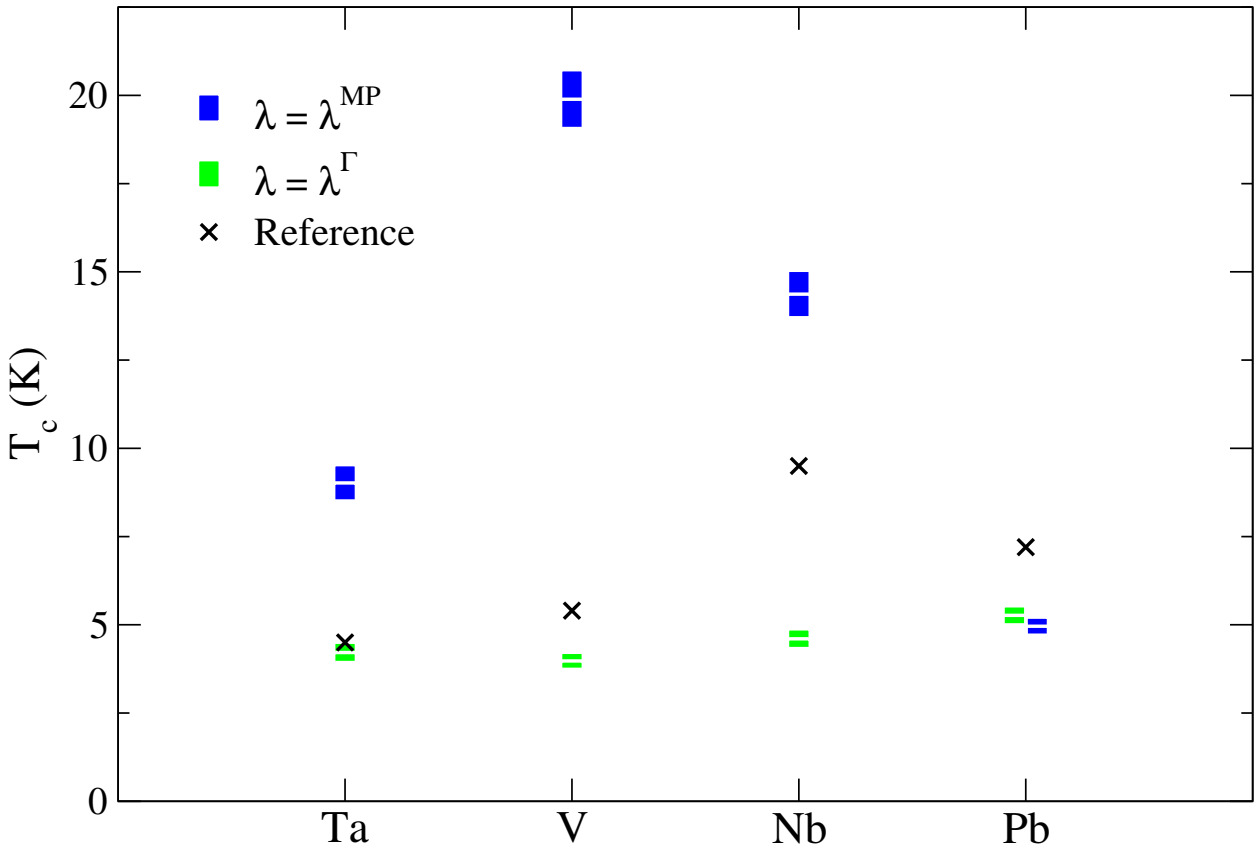

Figure 4.8: Comparison of calculated T_c with measured values from Ref. [89] using a 0.2 range of μ^* centered around the values given in Table 4.3, which are indicated by the white lines across the colored bars. Using the coupling constants calculated from the quasiparticle widths show better agreement with experiment.

Chapter 5

CORVUS: A WORKFLOW TOOL FOR MATERIALS SCIENCE AND SPECTROSCOPY

5.1 Introduction

During the past decade, our group has increased its usage of third party software to obtain properties that are unavailable in the code–base developed in–house. Most development has relied on the creation of interfaces between disparate software, using a plethora of languages and coding strategies. The end result has been a proliferation of workflows that are neither expandable nor connectable, and for which usage and development have very steep learning curves that create a barrier to entry. Existing scientific workflow software often provides a framework in which the user can construct their own workflows, [90–95] but this again requires the user to learn the vocabulary and format of the workflow software. In order to provide users of our software with a single workflow tool that minimizes their workload, allowing them to focus on physics instead of computational details, we have taken our previously disparate workflow designs and expanded to an all–encompassing tool for generically managing physical simulations that can span multiple scientific software packages. This design has been recently realized as the Corvus Python workflow tool, aimed to accommodate both beginner or impatient users only wishing to run pre–built workflows and expert or curious users wanting to use and expand on our provided functionality. By providing a solution for both target audiences, Corvus brings a unique approach to the scientific workflow landscape. This chapter presents the general design of Corvus along with a few example workflows for phonon property calculations. A more detailed application of Corvus for calculating thermal expansion is given in Chapter 6.

5.2 Corvus

The design of Corvus is divided into two main layers: (1) the top level code that interacts with the user and runs through the workflow and (2) the lower level code that interacts with the external scientific software to produce desired physical properties. This allows us to separate the implementation details specific to each external software package from the workflow controls, so we can design generic and modular workflow structures that can handle a wide variety of calculations.

Corvus relies on a *Workflow* object constructed of a sequence of modular steps that each provide a new piece of information necessary to calculated the desired physical property. These individual steps are dubbed *Exchanges*, in that a set of input properties is passed down to the external software that passes back up newly calculated output properties. The actual code that deals with the external software is separated into independent pieces, termed *Handlers*, that are each specific to one external software package. These Handlers contain the majority of our previous workflow efforts. This rest of this section gives further details on the design and the usage of Corvus.

5.2.1 Top level structure

The basic structure of running Corvus consists of the specification of the system being studied and choice of desired output properties, the creation and execution of the Corvus Workflow that will yield the target properties, and finally the reporting of the target properties. Figure 5.2 shows a flowchart for the high-level execution of a Corvus calculation.

Often, parameters in input files for scientific simulations are specific to how the simulation is run, and are therefore managed by the Handlers. This means we can have the user just provide a single minimal input file (see Section 5.2.4, which specifies the physical system. The input is parsed and stored in a system dictionary, or lookup table, that contains known physical properties. This table will update as the Corvus Workflow executes, incrementally growing the knowledge base of the system with each Exchange.

After getting the information provided by the user, the Workflow is created either by reading in a ready–made Workflow from file or automatically generating a Workflow based on the target property(ies) and available scientific software. More details on the Workflow structure itself can be found in Section 5.2.2. There is currently a rudimentary automated Workflow builder in place that can handle basic sequential workflows by building backwards from the desired property based on calculation dependencies. A fully general Workflow builder requires an algorithm that accounts for the fact that a single calculation can generate multiple desired properties and that a single property can be generated by multiple scientific software packages. As such, the standard dependency (acyclic directed) graph solutions are insufficient for these purposes (see Figure 5.1).

The CORVUS Workflow contains the sequence of Exchanges to be executed in order. We require the Handlers to implement generic methods for running an Exchange using its associated external software (see Section 5.2.3). This allows CORVUS to then run through the entire Workflow in an automated way. Each step in the calculation is run in a separate subdirectory, so the output from all the scientific software is later accessible in an organized fashion. Once finished, the output properties requested by the user are pulled from the property dictionary and written to file.

5.2.2 Workflow and Exchanges

The Workflow objects are designed to be minimal to allow CORVUS to easily iterate through the sequence of calculations. Specifically, a Workflow consists of three main components: (1) a list of target properties requested by the user, (2) a list of Exchange objects to execute in order to produce the requested properties, (3) and the minimum list of input properties required from the user. Though the sequence of Exchanges must be executed in a specific order to produce the desired properties, as determined by the dependencies of the property calculations, the actual execution of the Exchanges are independent of each other. By breaking up the Workflow into discrete Exchanges, it is easy to build in checkpointing and restarting of calculations. This is especially important when a user has a Workflow with

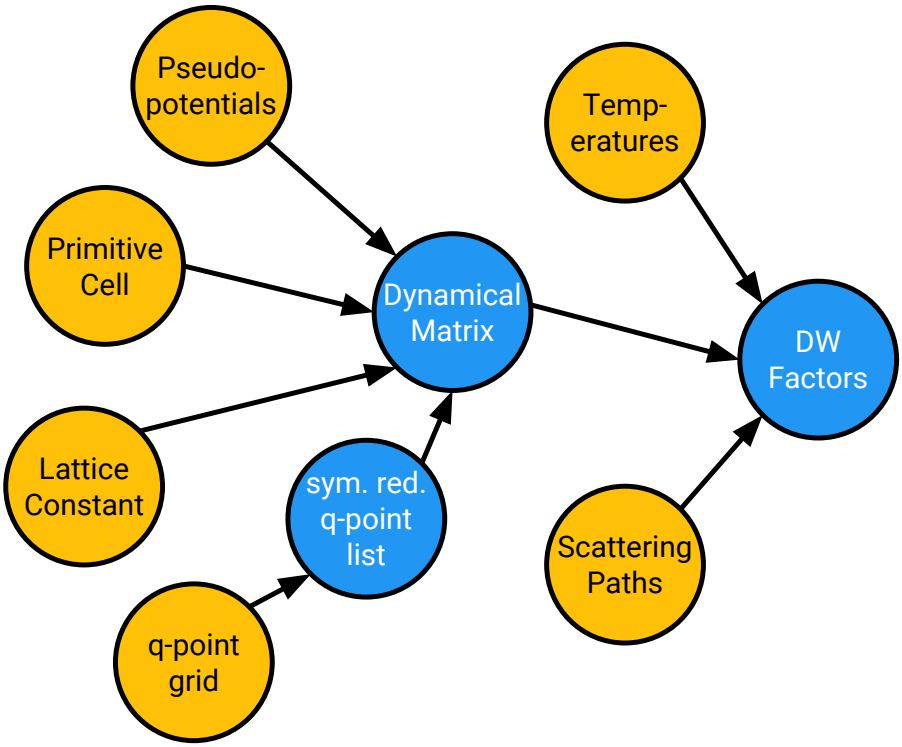

Figure 5.1: Illustration of a dependency graph for the main physical properties involved the calculation of Debye–Waller factors using the ABINIT and DMDW software packages. User–provided properties are shown in orange, and the primary calculated properties are shown in blue.

an expensive (with regards to time/resources) calculation up front, but wants to rerun the simulation with different physical parameters that only affect later parts of the calculation.

The Exchange itself is also a lightweight object, and merely stores a list of input properties to be pulled from the system property dictionary, a list of output properties to be calculated and passed back up to be stored in the system property dictionary, and the specification of which Handler (and therefore which external software) is to be used to produce the requested output. With this information, CORVUS can automatically call upon the specified Handler

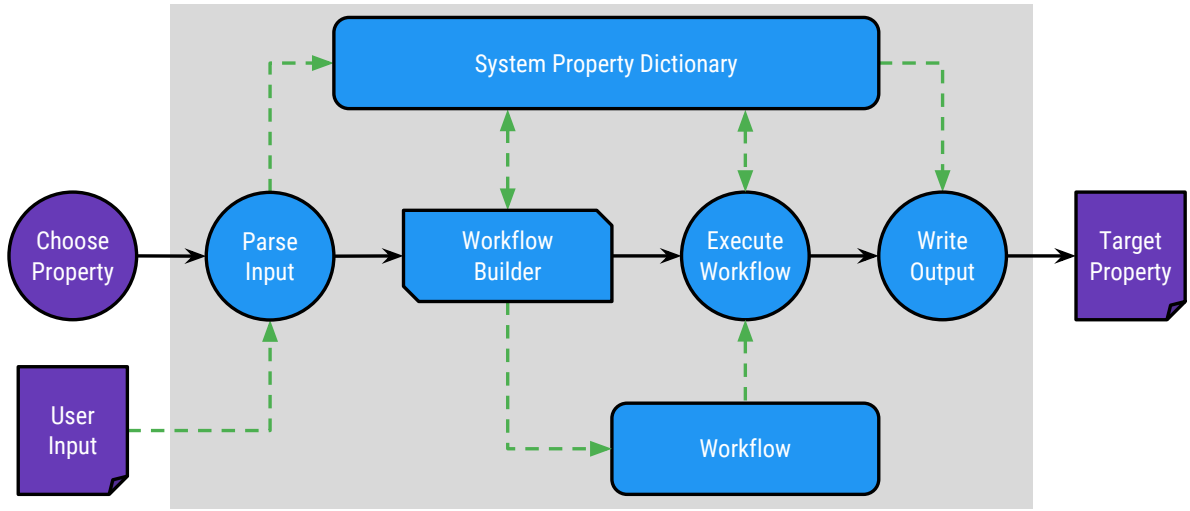

Figure 5.2: Overview of the top–level process of running CORVUS. The user responsibility is constrained to choosing the target and providing basic input. CORVUS creates a dictionary to store all information on the physical system being studied, which is continually updated as CORVUS runs through the Workflow. The solid black arrows indicate progression in time, while the dashed green arrows represent passing of information.

at each stage and have it generate the necessary output from the provided input without any knowledge of the implementation details.

In addition to the standard Exchange, there are currently two other Workflow sequence structures—Updates and Loops. An Update is used simply to overwrite a given property (or list of properties) in the system property dictionary. This is useful for tasks like initializing a property and replacing it with an optimized/converged value or parameter sweeps. The Loop structure is used to run the same Exchange multiple times, but iterating over a grid (or grids) of parameters by running an Update before for each Exchange iteration.

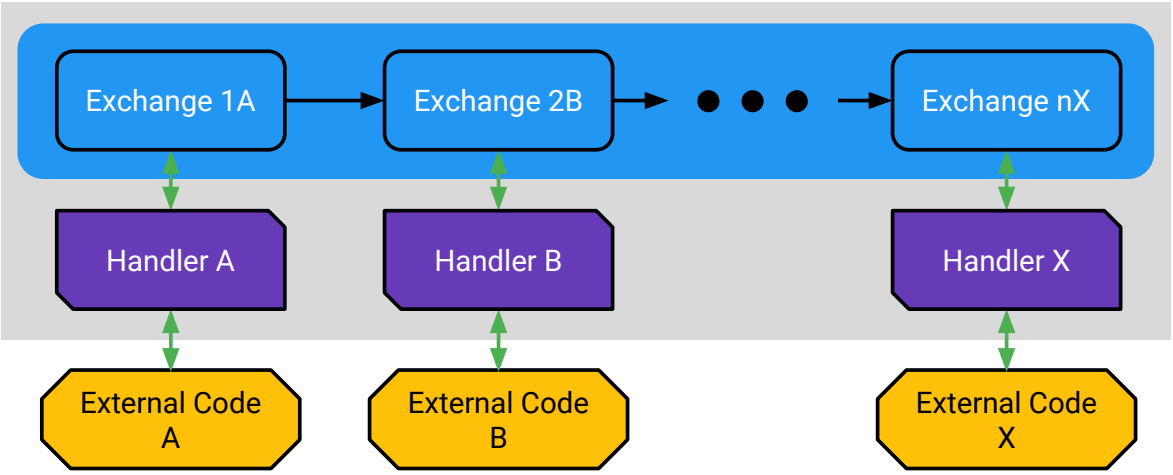

Figure 5.3: The Corvus Workflow object consists of a sequence of Exchanges. Each Exchange is an independent component of the Workflow, and uses its assigned Handler to run the external scientific software that produces the desired physical quantity. The new properties are then passed back up to the system property dictionary, so they can be accessed in subsequent Exchanges.

5.2.3 Handlers

As seen in the preceding sections, the structural design of Corvus is such that the flow control can simply operate at the top level without being hindered by technical details specific to the external software used to calculate the physical properties. This makes building Workflows nearly effortless, but relies on the necessary Exchanges being supported by the Handlers. As such, the vast majority of the technical development lies within the Handlers, which are in essence the evolution of our group's various scientific workflow tools.

The Handlers are completely independent of one another, so that the development for supporting the external scientific software packages can be done independently as well. A Handler only needs to implement a set of basic functions used by Corvus to create Workflows

and run Exchanges to interface with the top–level flow control:

`canProduce(output)` Returns True/False based on whether or not the associated external software can produce the given output(s).

`requiredInputFor(output)` Returns a list of input properties the associated external software requires to produce the given output(s). This includes any output properties not supported by this Handler.

`sequenceFor(output)` Returns the Workflow sequence required to produce the given output(s).

`prep()` Prepares a subdirectory where the external software is run.

`generateInput(input,output)` Creates the necessary input files used by the external software to produce the desired output and returns a list of them.

`run(files)` Runs the external software using the given list of input files.

`translateOutput(input,output)` Grabs the desired properties from output generated by the external software and translates them into the internal property dictionary format returned back up to update the system property dictionary.

`cleanup()` Does any necessary clean–up of the subdirectory before finishing up the Exchange.

As Corvus grows with new, more complicated Workflows and Exchanges, the above set required interfacing functions could feasibly grow as well.

5.2.4 Input

Continuing from AI2PS, we use the Rehr group generic token–based input format. The input–parsing tool has been translated into a Python version called Parsnip, which is a utility included in the CORVUS package. The input grammar is again composed of a token string followed by the input value(s) wrapped in curly braces (see Figure 5.4). Property tokens must be one string of non–whitespace characters. The input file supports standard commenting as well. Complete example input files can be found in Appendix B.

```
# comment
token1 {value} # comment
token2 {value value value}
token3 {
        value  # comment
        value  # comment
       }
token4  value
```

Figure 5.4: Example grammar for an input file read by Parsnip, the input parser used in CORVUS. As shown with `token4`, if an input value is a single entry with no whitespace, the curly braces can be dropped if desired. Comments can be denoted by the `#` and/or `!` characters.

5.2.5 Output

As CORVUS makes use of external software to calculate physical properties, there is output produced both by CORVUS and the external packages themselves. The raw output from each Exchange is kept in the subdirectory where the external software was run. The output from CORVUS is limited to the final target properties requested by the user. The CORVUS output formatting is still being finalized, but currently prints either column–based data or the Python string representation of the data.

5.2.6 Checkpointing and Savefiles

The user has the option of enabling basic checkpointing when running a CORVUS Workflow. Before running a Workflow step, the system property dictionary, the Workflow, and the most recently completed Workflow step are all recorded in a savefile. The savefile is written out using Python's `pickle` module, which is used for saving (loading) Python objects to (from) a binary–based format. This allows CORVUS to load in previously generated save states mid–Workflow and resume the calculation, i.e., tweaking settings from an old run without regenerating everything. The also lets users pass along a savefile to another user or computer and use the previously generated data as a springboard for new calculations by loading the old savefile along with a new Workflow. More details on the checkpointing/savefile usage can be found in Appendix B.

5.2.7 Requirements

One of the goals of CORVUS is to alleviate hindrances for the user, which includes the installation process. We have therefore kept CORVUS constrained to Python 2.7. Additional Python packages that provide useful analysis tools (e.g., Numpy [96]) will be treated like any other external software package, and be enabled through their own Handler. Consequently, users will only need Python 2.7 to install CORVUS, and the usage of CORVUS workflows will be determined by the specific external software packages installed on the system.

5.3 Example Workflows

In this section, we illustrate a few CORVUS Workflows for calculating various phonon properties. The first two Workflows demonstrate the current evolution of the AI2PS tool (see Chapter 3) within the CORVUS framework. Figure 5.5 shows the Workflow setup for calculating XAFS (x–ray absorption fine structure) and crystallographic Debye–Waller factors using ABINIT and DMDW. The AI2PS workflow comprises the last three steps, while the first two make use of the addition of the Update sequence to replace the user–provided lattice

constant with the optimized value. This kind of structure means users can give a rough estimate for a property, such as the lattice constant, provided the external software can determine the more optimal value.

Figure 5.6 shows the Workflow used for calculating phonon contributions to the electron self–energy and/or the electron spectral function, which brings in the work of Chapter 4. The Workflow is nearly identical to that for the Debye–Waller factors, but with the addition of the electron–phonon couplings α^2 coming along with the dynamical matrix from ABINIT. This Workflow also demonstrates the action of producing multiple output properties from a single Exchange. This is of particular interest for the automated generation of Workflows, as the generator needs to be able to detect that it can, essentially, feed two birds with one scone. The rudimentary automated Workflow generator mentioned in Section 5.2.1 is capable of generating the spectral function (and any other AI2PS-style) Workflow by using a greedy algorithm to maximize the output produced by an Exchange, which prevents the expensive ABINIT calculation from being errantly duplicated to produce the dynamical matrix and α^2 separately.

The final example (Figure 5.7 is a Workflow used to calculate the thermal expansion of solid state materials. This type of calculation is much more involved than the AI2PS-style Workflows that were first to be implemented in CORVUS, due to the need for target properties over a grid of input parameters. In the case of thermal expansion calculations, we need the dependence of the internal energy U and the vibrational free energy $F_{\text{vib}}(T)$ on the lattice constant a, which precipitated the need for the Loop structure described in Section 5.2.2. The end products of this Workflow are then $U(a)$ and $F_{\text{vib}}(T, a)$, the sum of which is minimized at $a(T)$, which gives the linear thermal expansion. The thermal expansion simulations are covered in more detail in Chapter 6.

5.4 Conclusions

CORVUS is a new physical property–based scientific workflow tool that, while inspired and informed by our previous workflow efforts, has been designed from the ground up to accom-

modate all of our past, present, and future calculations utilizing multiple software packages. We have structured Corvus to keep any technical details concerning the external scientific software well–separated from the flow control to allow for automated calculations and easy construction of Corvus Workflows. Corvus is intended for both inexperienced and ambitious users by operating both as a single–shot execution of a workflow and as an open library of functions ripe for customized Workflows. The software is currently in the alpha stage of development, meaning specific implementation details are subject to change in the future, though the core structure and design will remain unchanged. Further information for obtaining and using Corvus can be found in Appendix B.

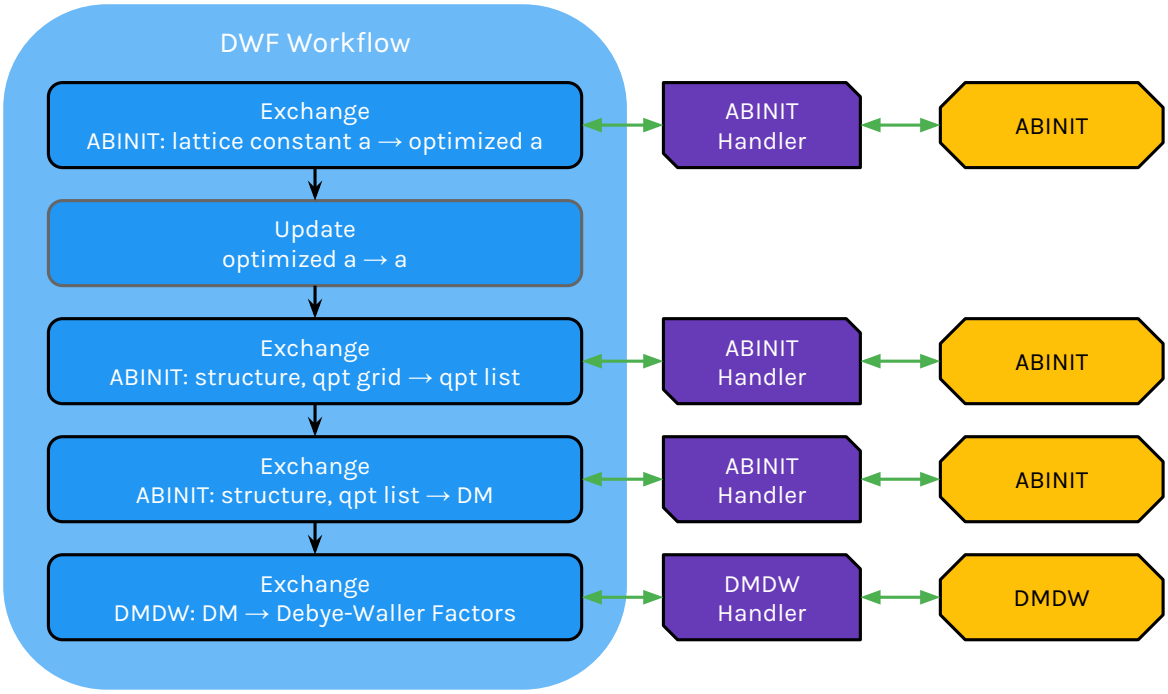

Figure 5.5: Workflow for calculating Debye–Waller factors using the ABINIT and DMDW software packages. The first two steps have been added to the automatically generated workflow to instead use the lattice constant that optimizes the total internal energy for the chosen exchange correlation functional. The automatic generation of the symmetry–reduced q–point list from a given $n \times m \times l$ grid, added from AI2PS, is a major component in reducing the user workload through automated input generation.

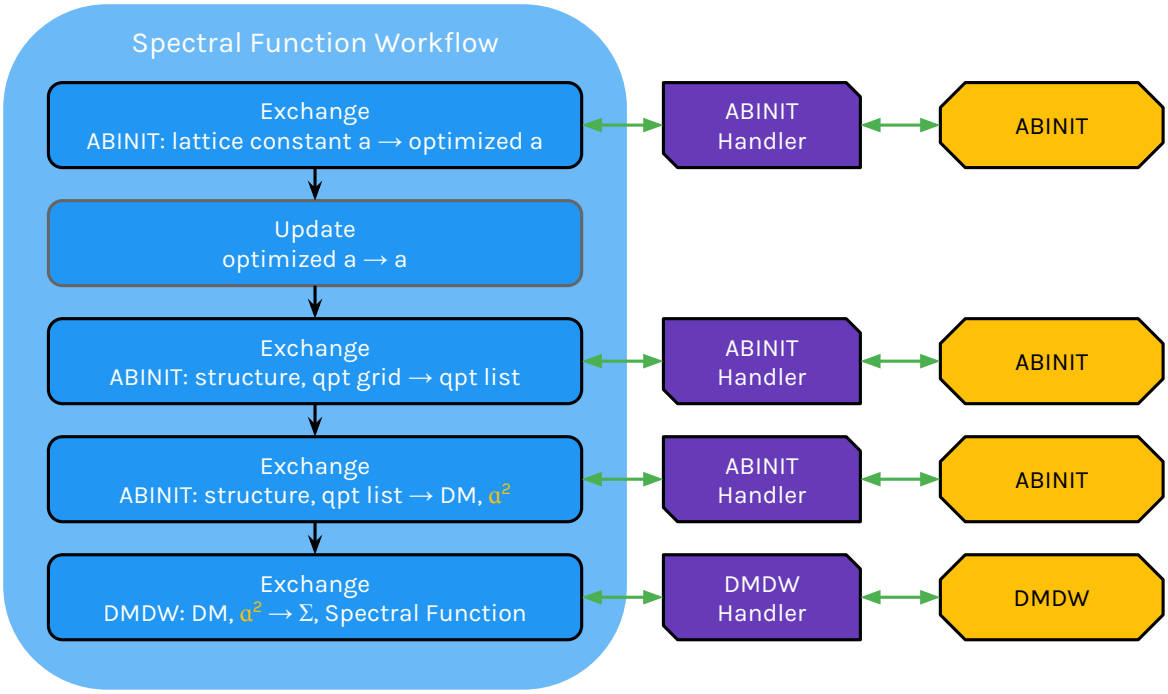

Figure 5.6: Workflow for calculating the phonon contributions to the electron spectral function (and electron self–energy Σ) using the ABINIT and DMDW software packages. This Workflow is nearly identical to the one for calculating Debye–Waller factors (Figure 5.5), but with the addition of the electron–phonon couplings α^2 needed to calculate Σ. Note the self–energy and the spectral function can be obtained from the same Exchange if both are desired by the user.

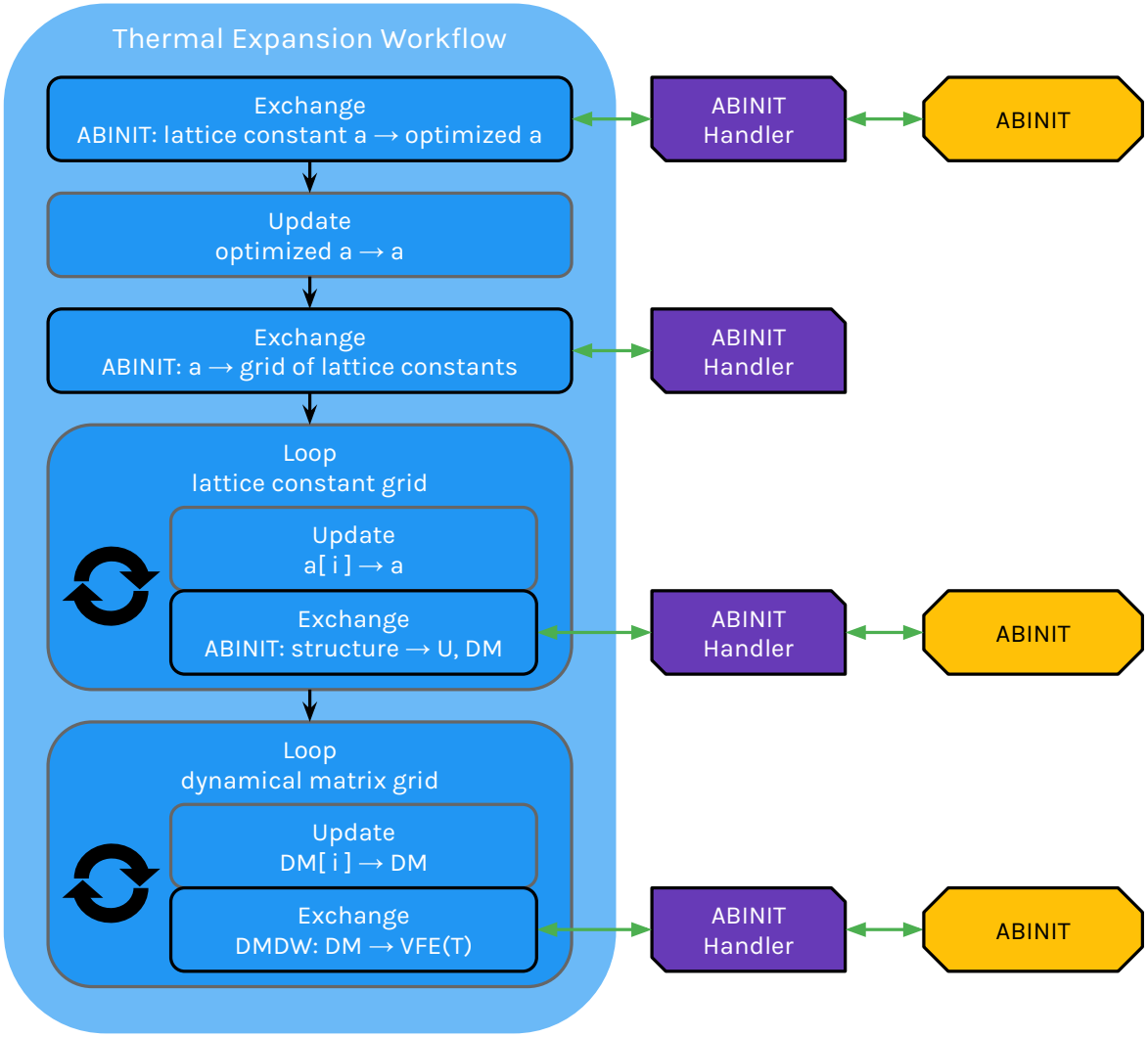

Figure 5.7: Workflow for calculating thermal expansion using the ABINIT and DMDW software packages. The first two steps have been added to the automatically generated workflow to instead use the lattice constant that optimizes the total internal energy for the chosen exchange correlation functional. The automatic generation of the symmetry–reduced q–point list from a given $n \times m \times l$ grid was a major step in reducing the user workload.

Chapter 6

MODELING THERMAL EXPANSION FOR IMPROVED SIMULATIONS OF VIBRATIONAL PROPERTIES

Accurate simulation of thermal expansion of materials is of considerable importance for calculating temperature–dependent vibrational properties, as the anharmonicity of the interatomic forces makes them sensitive to the lattice constant a. We can therefore significantly improve the results of our vibrational property calculations by first determining a good model for $a(T)$ and then using it in subsequent calculations. However, the extensive procedure for calculating $a(T)$ made it a slow, tedious process. With the development of the CORVUS workflow tool and the implementation of a thermal expansion workflow, we can simulate thermal properties more efficiently. Here we will present the recent work on thermal expansion calculations with CORVUS, using the ABINIT and DMDW software packages, for bulk copper and gallium arsenide. GaAs is of particular interest due to its *negative* thermal expansion observed at low temperatures. [97–99] Such negative thermal expansion materials are important to understand, as they can be used in the development of non–expanding composite materials. [100]

6.1 Methodology

As we saw in Chapter 2, minimizing the Helmholtz free energy with respect to the system size gives us the thermal expansion of the system. In practice, simulations parameterize the size of a solid state system in terms of the lattice constant a instead of the volume V. Equation 2.30 can be reexpressed as

$$F(a,T) = U(a) + k_B T \int_0^\infty d\omega \ln\left[2\sinh\left(\frac{\beta\hbar\omega}{2}\right)\right]\rho_a(\omega)\,. \tag{6.1}$$

Both the total internal energy and vibrational free energy can be obtained from a DFT

calculation (ABINIT [40–44] in this work), though the vibrational free energy is more efficiently calculated with the Lanczos algorithm used in DMDW. [14] As such, we will be using the DFT calculation to determine $U(a)$ for a grid of lattice constants, as well as the dynamical matrices for that same grid. The dynamical matrices are generally expensive calculations, taking on the order of hours for ~ 100 modern processors, so we will confine the lattice constant grid to $\sim 5-10$ points. DMDW can then take each grid point a_i, and calculate $F(a_i, T)$ over an arbitrary grid of finite temperatures. The efficiency of the vibrational free energy calculation in DMDW allows us to use a much finer temperature grid, as hundreds of data points can be calculated in minutes on a modern machine. The two–dimensional grid for the heat capacity at constant volume $C_V(a_i, T_i)$ can be calculated with DMDW in similar to the Helmholtz free energy. The electronic contribution to C_V has also been included through the use of the Sommerfeld model of an electron gas [4, 8]

$$C_V^{\mathrm{el}} = \frac{\pi^2}{3} k_B^2 T \, g\left(\epsilon_F\right), \tag{6.2}$$

where $g(\epsilon_F)$ is the electron DOS evaluated at the Fermi energy, which was provided by an additional DFT calculation.

Finally, the data for $U(a_i)$, $F(a_i, T_i)$, $C_V^{\mathrm{ph}}(a_i, T_i)$, and $g(\epsilon_F)$ are processed via a Python/Numpy [96] script, which is provided in Appendix C. In the quasiharmonic approximation, the temperature slices of the Helmholtz free energy $F(a, T_i)$ can be fit to a quartic to determine the lattice constant $a(T_i)$ corresponding to the equilibrium configuration. By repeating this process for a large grid of temperatures, we can determine $a(T)$. Due to the cubic symmetry of the bulk materials studied here, we can focus on the linear thermal expansion coefficient for simplicity. The volumetric and linear expansion coefficients for cubic systems are related as follows

$$\alpha_V = \frac{1}{V}\frac{dV}{dT} = \frac{3}{a}\frac{da}{dT} = 3\alpha_L. \tag{6.3}$$

Similarly, the derivatives with respect to volume used to calculate the pressure (Eq. 2.31) and bulk modulus (Eq. 2.51) can be converted to derivatives with respect to the lattice constant

using the volume of the fcc primitive cell

$$V_{\text{prim}}(a) = \det \begin{pmatrix} 0 & \frac{a}{2} & \frac{a}{2} \\ \frac{a}{2} & 0 & \frac{a}{2} \\ \frac{a}{2} & \frac{a}{2} & 0 \end{pmatrix} = \frac{a^3}{4}, \tag{6.4}$$

which is used for both fcc Cu and zincblende GaAs structures. Further details on the CORVUS thermal expansion workflow, as well as a script for the analysis described above, can be found in Appendix C.

6.2 Thermal Expansion Results for Cu

Bulk copper is a well–studied standard material, and so serves as the initial test case for our thermal expansion calculations. Figure6.1 displays temperature–cuts of the Helmholtz free energy surface $F(a, T)$, as well as the curve that traces the minimum of $F(a, T_i)$, which indicates the lattice constant at T_i. The resulting calculated thermal expansion of the Cu lattice constant $a(T)$ is given in Figure 6.2. The linear expansion coefficient results, shown in Figure 6.3, agree qualitatively quite well with experiment, though we can see the well–known behavior of the standard LDA exchange–correlation functional in underestimating the expansion coefficient in metallic systems such as Cu. [48] The bulk modulus is similarly overestimated by the LDA functional (Figure 6.4). However, these effects essentially balance out and yield good agreement for the isobaric heat capacity C_P, as seen in Figure 6.5. It would be of interest to explore the use of hybrid exchange–correlation functionals, such as PBESol [101, 102] or hGGA [14], which could possibly improve the quantitative agreement with experiment.

6.3 Negative Thermal Expansion Results for GaAs

GaAs is another good case study since (1) it is a relatively simple system to calculate using ABINIT and (2) it also displays the uncommon negative thermal expansion behavior at low temperatures, which has been observed experimentally, although it is a very small effect.

[97–99] Figure 6.6 shows the GaAs $F(a, T)$ for the same temperatures as those plotted for Cu in Figure 6.1, which indicates the equilibrium a does not start to increase quite as rapidly as with Cu. We can start to see this in the GaAs $a(T)$ plot in Figure 6.7, where the lattice constant stays roughly constant for about twice the temperature range as compared with Cu. The actual contraction, or negative expansion, becomes apparent in the results for the linear expansion coefficient (see Figure 6.8). There is a clear region of negative $\alpha(T)$ at low temperatures, although it is relatively small.

For comparison to experimental results, we focus on the lower temperature range and look at the expansion of the atomic distance between neighboring cells, i.e., the Ga–Ga separation in Figure 6.9. Note the observed contraction is on the order of a mere 10^{-5} Å, which is not a trivial computational feat. Again, it would be of great interest to investigate the effects of the functional choice in the DFT calculations, though great care should be taken as the GaAs calculations require significant computational resources—the GaAs $a(T)$ results shown here required on the order of 10,000 core–hours.

6.4 Conclusions

The latest developments of the CORVUS workflow tool has enabled us to simulate the thermal expansion of materials much more easily and efficiently. Our first test cases of Cu and GaAs demonstrate good qualitative agreement with experiment, although we predict improved quantitative performance could be achieved in the case of metals through the use of hybrid DFT functionals. Furthermore, calculating the thermal expansion is a major step in providing the means to improved simulations of thermal properties due to their sensitivity to expansion of the system. Additionally, the analysis methods used in the accompanying Python script (see Appendix C.4) were specifically written with only NumPy dependencies, which will allow them to be easily incorporated into a CORVUS NumPy Handler, giving users access to fully automated workflows for thermal properties with both temperature and volume dependence.

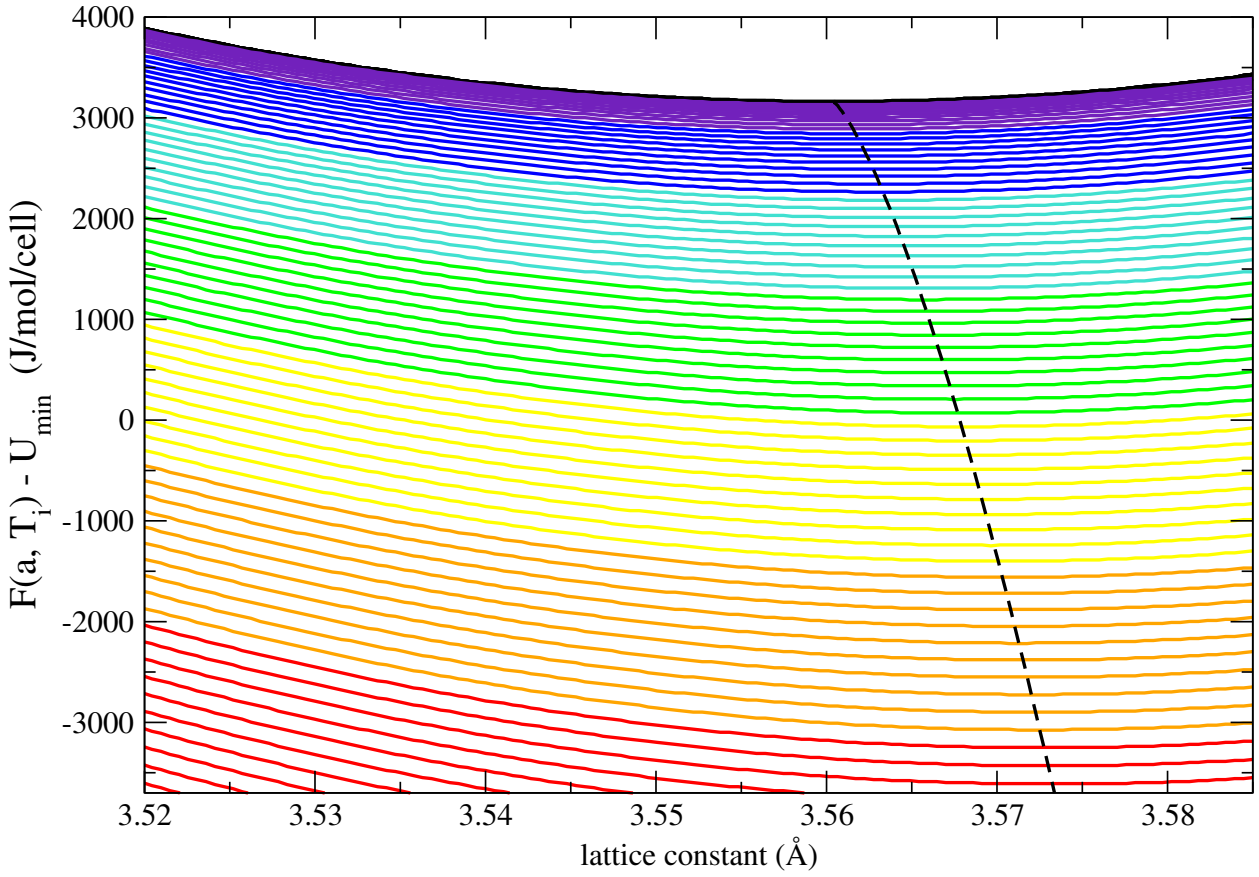

Figure 6.1: Temperature cuts of the Cu Helmholtz free energy surface $F(a, T)$ used to determine thermal expansion of the lattice constant $a(T)$, shown as the dashed line. Energies are plotted relative to the minimum of $U(a)$. The temperature range plotted, shown here increasing from purple to red, is from 3.15 K to 398.15 K in 5 degree increments.

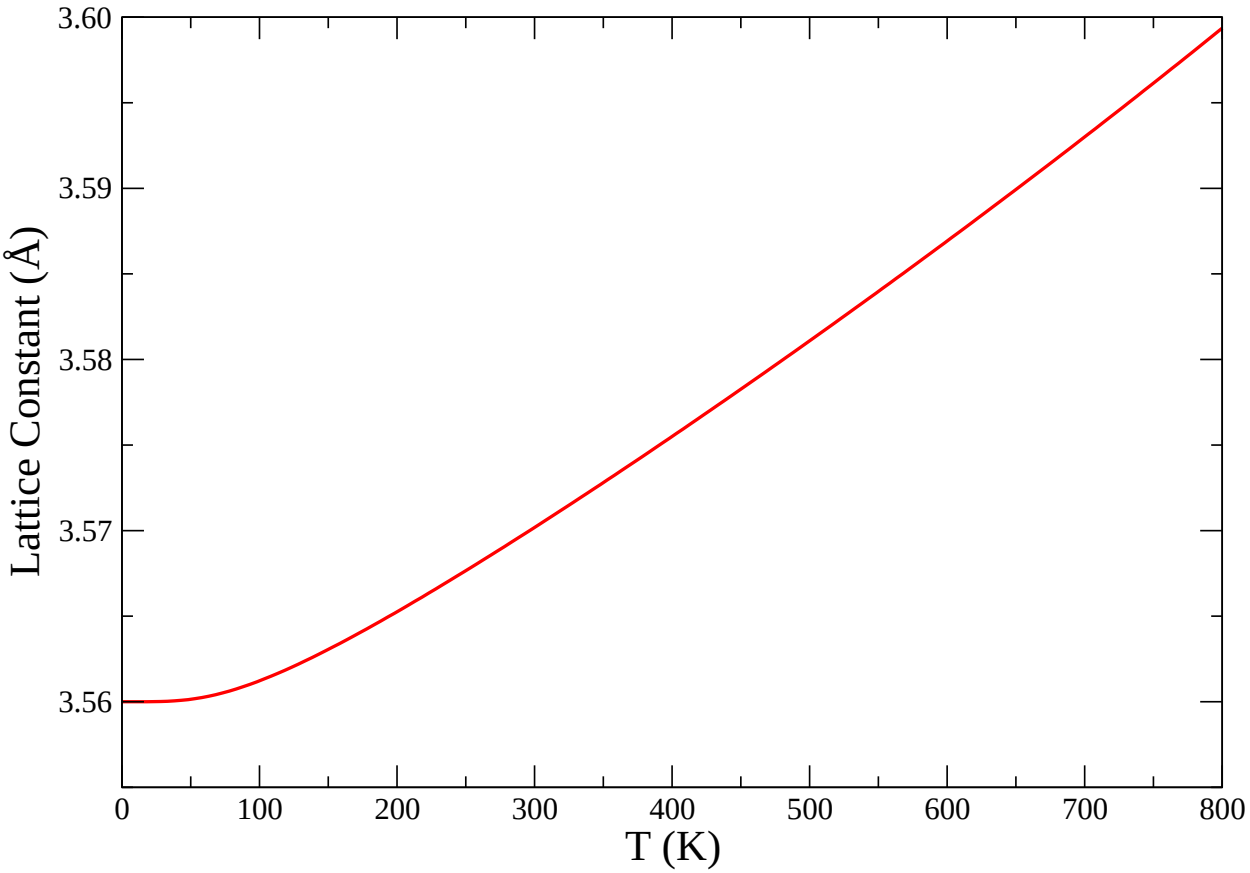

Figure 6.2: Thermal expansion of copper lattice constant calculated with the LDA exchange–correlation functional.

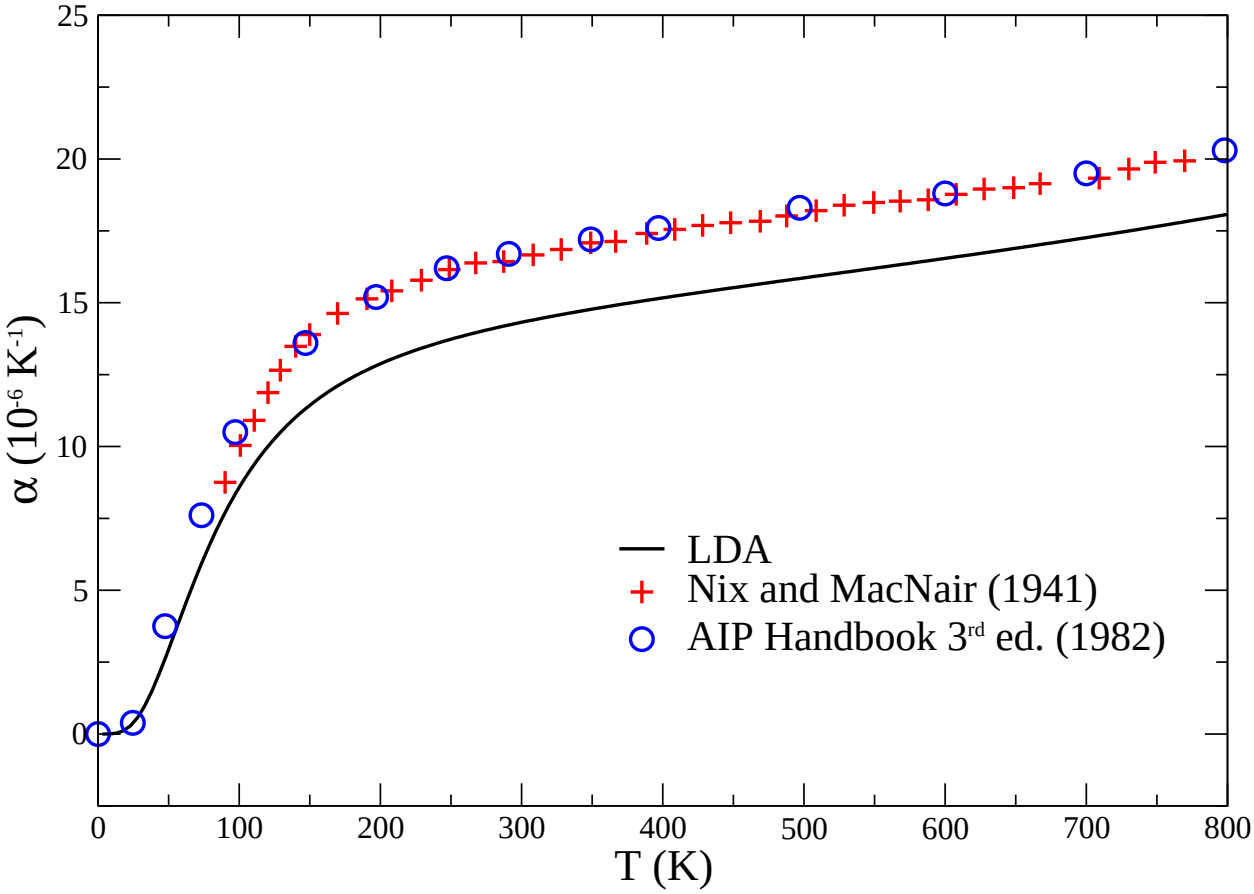

Figure 6.3: Linear thermal expansion coefficient for copper using the LDA exchange–correlation functional, with experimental results [103, 104] for comparison. The LDA functional underestimates the expansion coefficient for metals, yet captures the correct temperature trend.

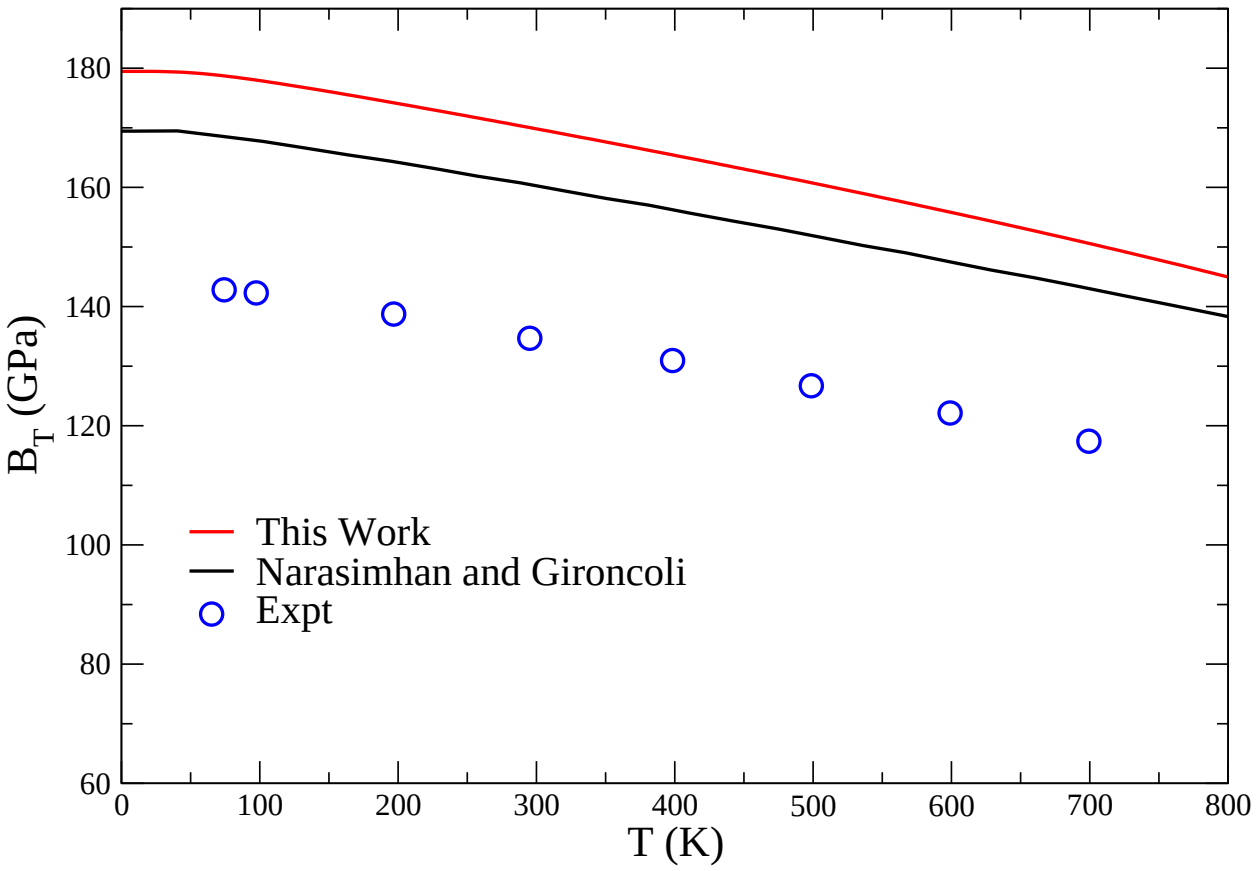

Figure 6.4: Isothermal bulk modulus of copper calculated using the LDA exchange–correlation functional for two methods: (1) a finite differences method to directly calculate $B_T = V(d^2F/d^2V)_T$ (this work, red) and (2) $F(a,T)$ fit to the fourth–order Birch–Murnaghan equation of state (Ref. [48], black). Experimental data also included for comparison [105] (blue). The finite differences method is less accurate, but manages to capture the correct temperature trend.

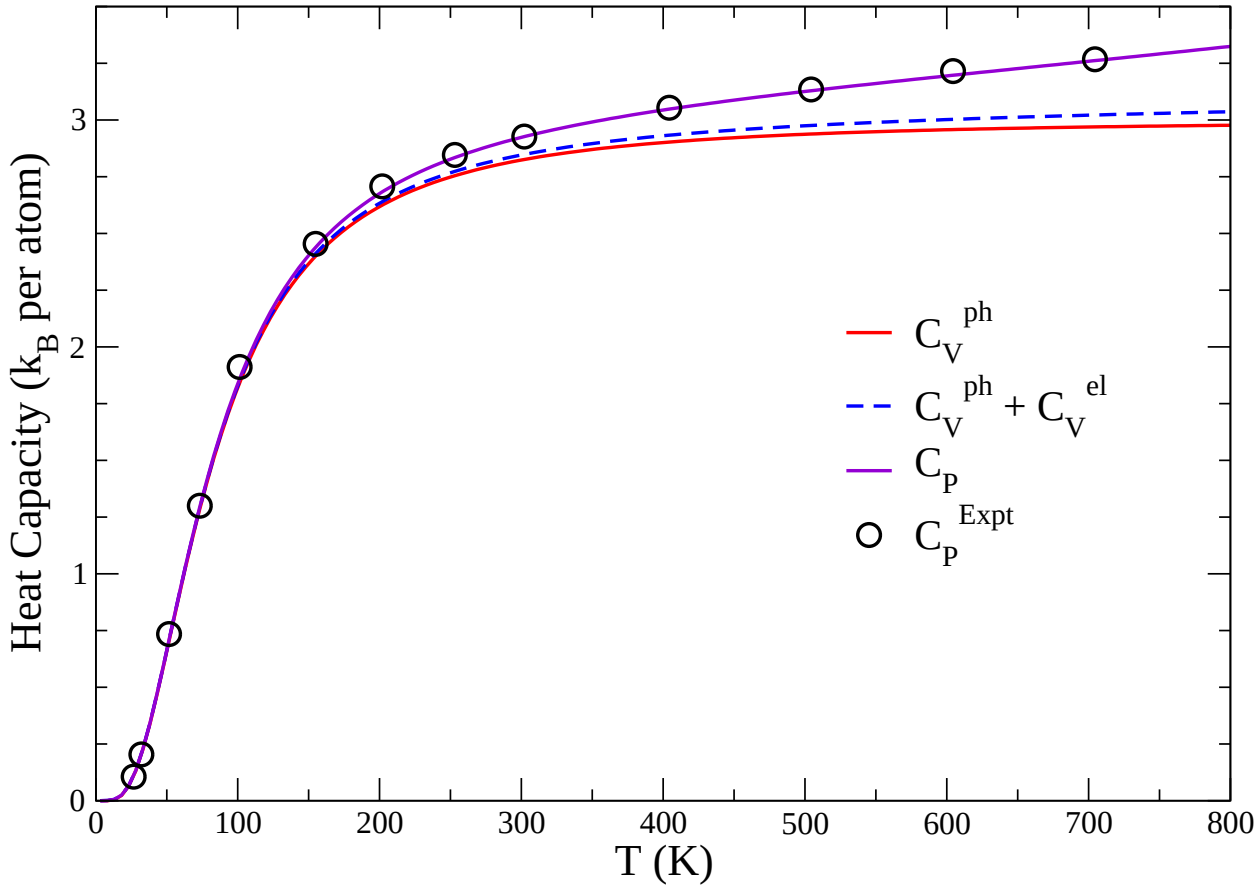

Figure 6.5: Molar heat capacity of copper at constant volume and constant pressure, with experiment for comparison. [104] Though the phonons dominate the heat capacity, the electrons provide a noticeable contribution.

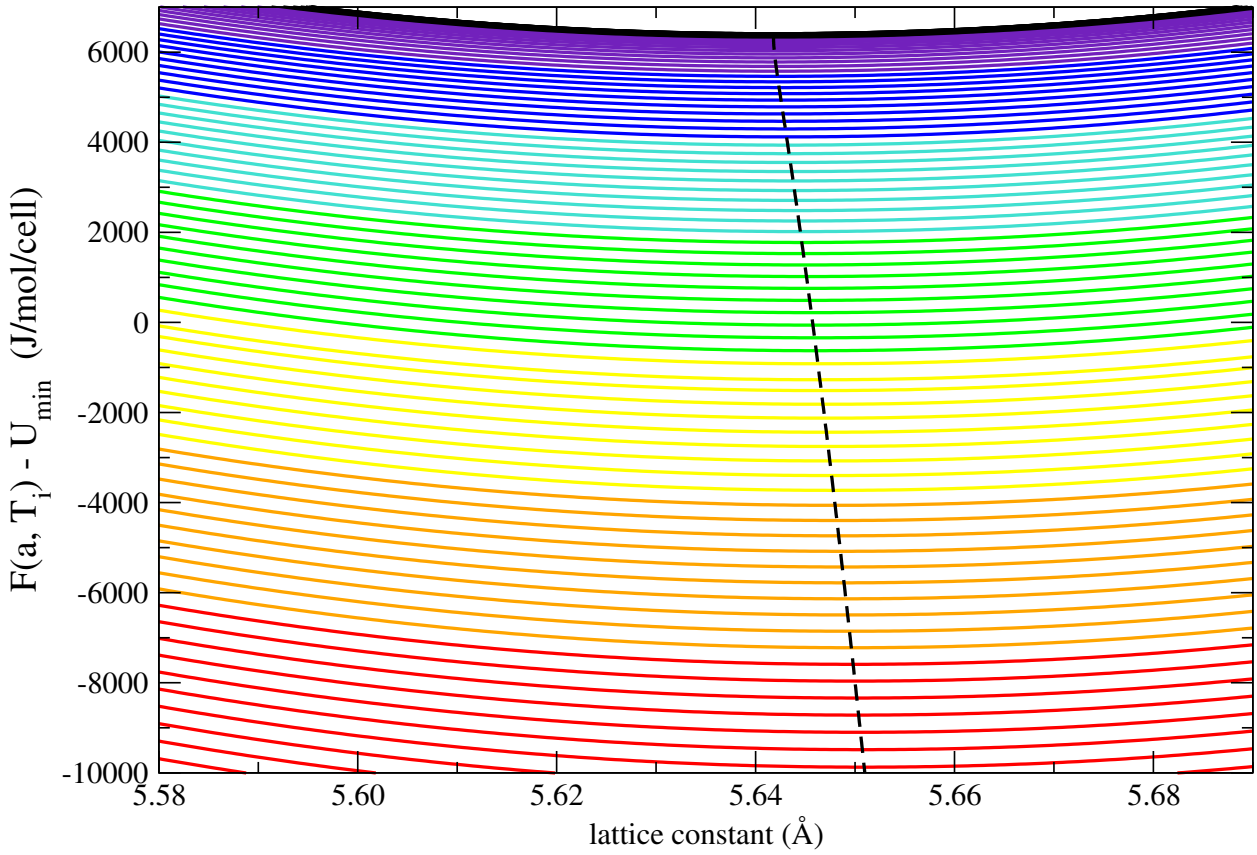

Figure 6.6: Temperature cuts of the GaAs Helmholtz free energy surface $F(a, T)$ used to determine thermal expansion of the lattice constant $a(T)$, shown as the dashed line. Energies are plotted relative to the minimum of $U(a)$. The temperature range plotted, shown here increasing from purple to red, is from 3.15 K to 398.15 K in 5 degree increments.

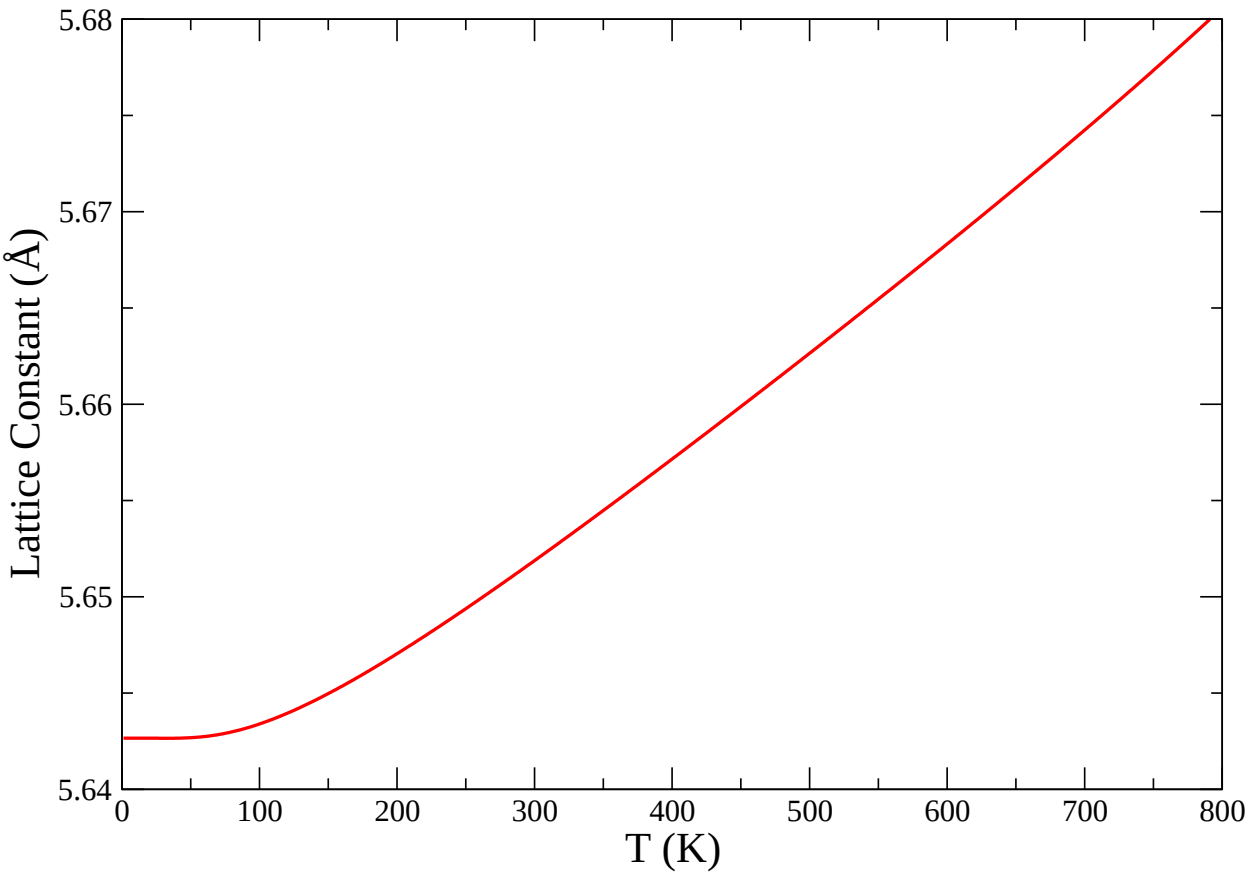

Figure 6.7: Thermal expansion of GaAs lattice constant calculated with the LDA exchange–correlation functional.

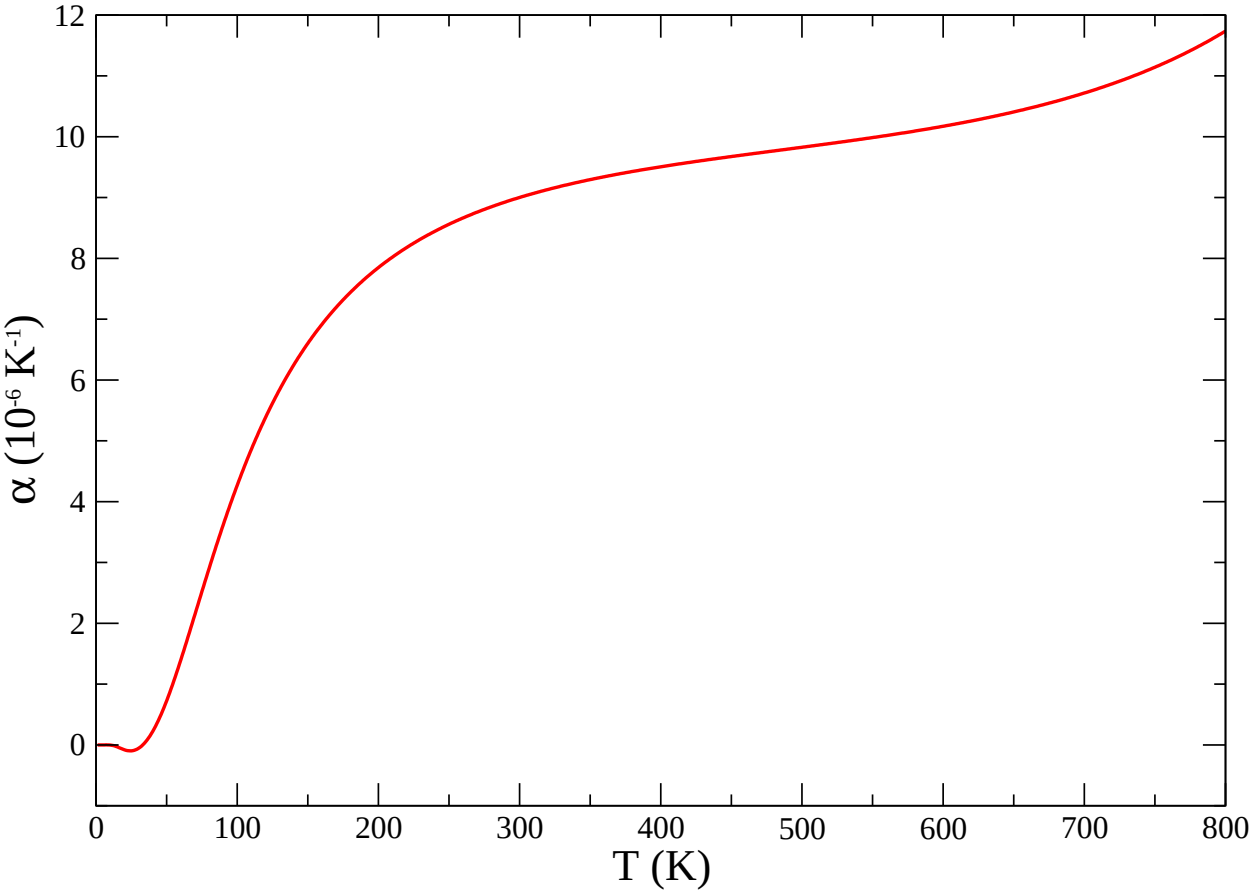

Figure 6.8: Linear thermal expansion coefficient for bulk GaAS using the LDA exchange–correlation functional.

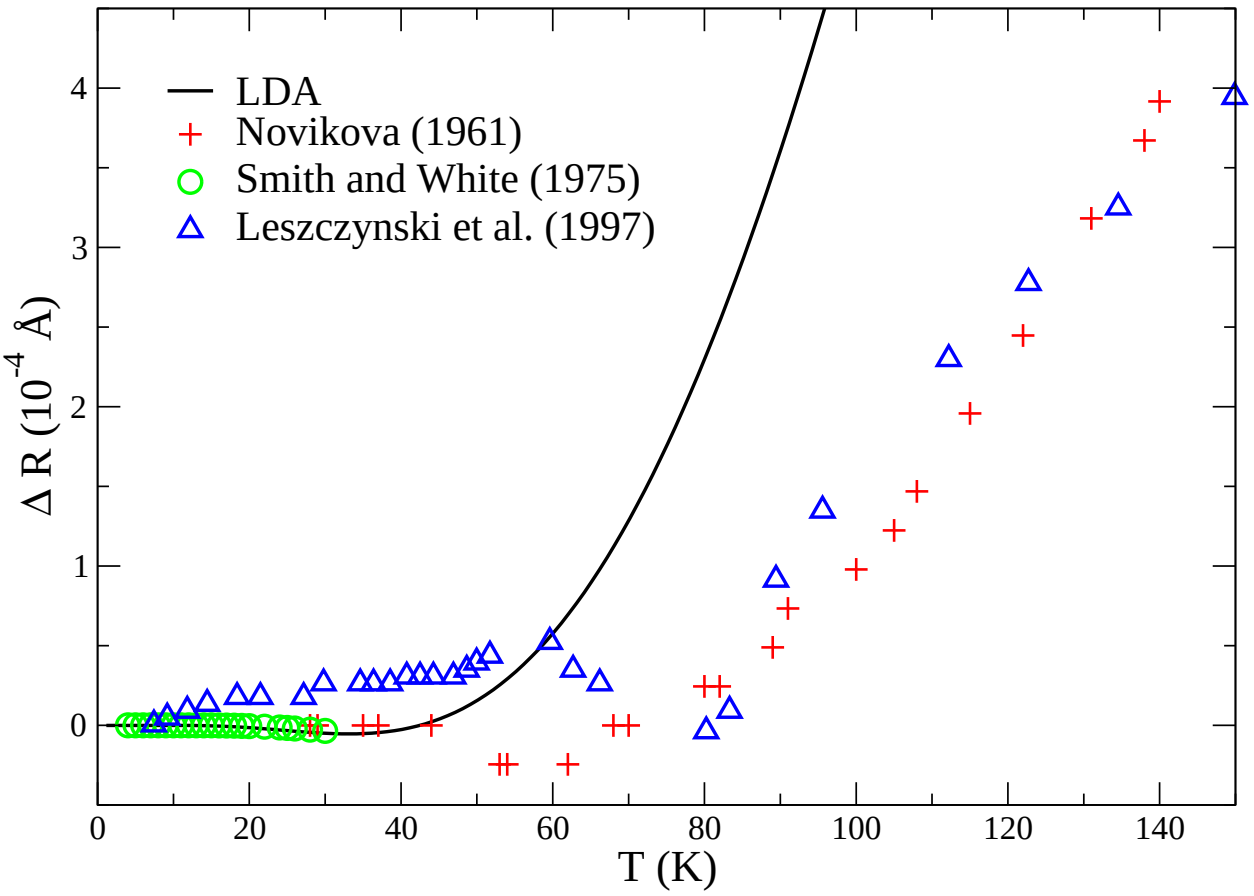

Figure 6.9: Thermal dependence of interatomic distances (Ga–Ga, As–As) in GaAs seen at low temperatures, with experimental results for comparison. [97–99] Negative thermal expansion has been observed at temperatures below 100 K and shows up in our simulation around 30 K.

Chapter 7

CONCLUSIONS

We have demonstrated that the simulation of a wide variety of phonon properties can be conceptually reduced to a common set of tasks—the production of the dynamical matrix that yields the vibrational density of states, which, combined with a number of additional physical properties of the system, can be used to calculate the desired phonon property. This functional equivalence led us to develop AI2PS, a scientific workflow tool capable of driving the full multi–code phonon property calculation, much in the vein of our previously established AI2NBSE and OCEAN packages [36–39] for the simulation of UV/visible spectra. AI2PS gave us easy access to XAFS and crystallographic Debye–Waller factors, as well as phonon contributions to the electron self–energy and electron spectral function.

The latter two were used to show that the newly developed retarded cumulant (RC) method for calculating the electron spectral function, though yielding significant improvements for plasmon satellites over the widely–used Migdal approximation method, showed good agreement with the Migdal approximation outside of the low–temperature ($\lesssim$ 50 K) and strong electron–phonon coupling. [55] As spectroscopy experiments continue to improve on the meV resolution necessary to resolve the relatively small phonon satellites, [106, 107] we will be able to test the performance of the two methods for phonons. The spectral function simulations also yielded electron–phonon coupling constants, and the related superconducting transition temperatures T_c for a number of bulk materials, which we were able to calculate to within an order of magnitude of experiment.

AI2PS was also successfully integrated into the Java GUI–based (graphical user interface) package for our group's main software product FEFF9, with access to the Amazon Web Services (AWS) cloud computing platform. [108] This let users run the computationally

expensive phonon property calculations on the cloud from their laptop. This was a good demonstration of the flexibility of the AI2PS workflow design. However, as AI2PS was being prepared for public release, it became clear that a more extensive and general tool, one that could combine our separate scientific workflow tools, would not only be incredibly useful to our users, but it would foster a more maintainable software ecosystem for our group's developers, especially for the increasingly complicated simulations (e.g., thermal expansion) we hoped to implement.

At this point, AI2PS metamorphosed into the property–based scientific workflow tool CORVUS. The updated workflow framework was designed with the typical usage of multiple physical simulations in mind, i.e., feeding physical properties into one software package to generate additional physical properties, which are then fed into another software package to generate more physical properties, and so on. By implementing the automated tools of AI2PS inside the generalized structure of CORVUS, we could trivially connect multiple calculations, which reduced the construction of the coveted thermal expansion automated workflow in the CORVUS framework to a $\sim$ 20–line Python script (see Appendix C.1).

With the recent implementation of the automated thermal expansion simulation, we are now able to easily study thermal expansion of interesting materials, especially negative thermal expansion materials. We have already shown we can simulate the extremely small ($\sim 10^{-5}$Å) negative thermal expansion observed for GaAs at low temperatures. [97–99] We would also like to used this to simulate perovskites, such as zirconium tungstate ZrW_2O_8, which displays negative thermal expansion roughly four times the order of magnitude of that in GaAs. [109] Not only can we study thermal expansion in materials with CORVUS, but we can also now include temperature *and* volume dependence in various thermal properties. It is merely a matter of first calculating the length (or volume) expansion $a(T)$ (or $V(T)$), and using those values in any subsequent thermal property calculation, as demonstrated with the C_P calculation for copper.

Finally, though the focus of this work has been on various vibrational properties, the CORVUS framework is capable of implementing our other workflow tools, such as AI2NBSE

and OCEAN, which will add support for optical and core spectra calculations. In addition, CORVUS will add support for additional DFT engines (VASP, Gaussian, NWChem, Quantum ESPRESSO) in the near future, This will allow users to simulate different types of condensed matter systems (e.g., molecules or crystals with large cells) more efficiently by catering to the specific strengths of each DFT package. The key feature of CORVUS is its ability to continually grow and improve on its capabilities.

Appendix A

AI2PS SUPPLEMENTARY DETAILS

This appendix contains more in–depth information related to the content found in Chapter 3.

A.1 Running AI2PS

A.1.1 Supported Command Line Options

AI2PS supports the following command line arguments:

`-e`

Enable frontend calculation of dynamical matrix using selected DFT engine

`-p`

Enable DMDW backend calculation of specified vibrational property

`-t [filename]`

Run translator, associated with selected DFT engine, on given file to generate DMDW dynamical matrix file

`-0` *(zero)*

Generate input files only

`-i [filename]`

Specify input file (default `ai2ps.in`)

`-c [filename]`

Specify custom config file (default `ai2ps.config`)

`-n [integer]`

Specify number of processors used for parallel calculations

A.1.2 Example Commands

The following are example commands for typical usage:

`ai2ps -ep -n64 -i my.input`

Run full calculation on 64 processors with input file `my.input`

`ai2ps -fc my.config`

Run DFT engine only using config file `my.config`

`ai2ps -t ifcinfo.out -b`

Convert ABINIT interatomic force constants in `ifcinfo.out` into dynamical matrix file for DMDW, and run DMDW

A.2 Example Input Files

A.2.1 Input for Cu XAFS Debye–Waller Factors

```
name          {Cu}
engine        {abinit}
property      {s2}
#-----------------------------------abinit----------------------------------------
pspfil        {29-Cu.LDA.fhi}      # list of pseudopotential files
natom         {1}                  # number of atoms
ntypat        {1}                  # number of types of atoms
typat         {1}                  # type of atoms
znucl         {29}                 # charge Z of the nucleus
xred          {0.00 0.00 0.00}     # vectors (x) of atom positions
                                   #    in reduced coordinates
acell         {3*6.82+00}          # experimental lattice constant
rprim         {0.0  0.5  0.5       # real space primitive translations
               0.5  0.0  0.5
               0.5  0.5  0.0}
ngkpt         { 8  8  8}           # number of grid points for k points generation
ecut          {60.0}               # energy cutoff

# additional abinit input settings appended verbatim
verbatim      {
               occopt 6
               tsmear 0.025
               nband 12
              }
#-----------------------------------anaddb----------------------------------------
ngqpt         { 4  4  4}           # coarse q-grid
ng2qpt        {16 16 16}           # fine q-grid
```

```
#-------------------------------------dmdw-----------------------------------------
dmdw.ioflag    {1}                  # more detailed output level
dmdw.nlanc     {16}                 # number of Lanczos poles
dmdw.ntemp     {100}                # number of temperature in T-grid
dmdw.temp      {5.0 500.0}          # temperature grid endpoints in K
dmdw.paths     {                    # single scattering paths for all atoms within
                1                   #   3.0 Ang of the central atom (172/343)
                2 172 0 3.0
               }
```

A.2.2 Input for Cu Crystallographic Debye–Waller Factors

```
name           {Cu}
engine         {abinit}
property       {u2}
#-------------------------------------abinit----------------------------------------
pspfil         {29-Cu.LDA.fhi}      # list of pseudopotential files
natom          {1}                  # number of atoms
ntypat         {1}                  # number of types of atoms
typat          {1}                  # type of atoms
znucl          {29}                 # charge Z of the nucleus
xred           {0.00 0.00 0.00}     # vectors (x) of atom positions
                                    #    in reduced coordinates
acell          {3*6.82+00}          # experimental lattice constant
rprim          {0.0  0.5  0.5       # real space primitive translations
                0.5  0.0  0.5
                0.5  0.5  0.0}
ngkpt          { 8  8  8}           # number of grid points for k points generation
ecut           {60.0}               # energy cutoff

# additional abinit input settings appended verbatim
verbatim       {
                occopt 6
                tsmear 0.025
                nband 12
               }
#-------------------------------------anaddb----------------------------------------
ngqpt          { 4  4  4}           # coarse q-grid
ng2qpt         {16 16 16}           # fine q-grid
#-------------------------------------dmdw-----------------------------------------
dmdw.ioflag    {1}                  # more detailed output level
dmdw.nlanc     {16}                 # number of Lanczos poles
dmdw.ntemp     {100}                # number of temperature in T-grid
dmdw.temp      {5.0 500.0}          # temperature grid endpoints in K
dmdw.paths     {                    # central atom (172/343)
                1
                1 172 1.0
               }
```

A.2.3 Input for Cu Vibrational Free Energy

```
name          {Cu-vfe}
engine        {abinit}
property      {vfe}
#-----------------------------------abinit---------------------------------------
pspfil        {29-Cu.LDA.fhi}      # list of pseudopotential files
natom         {1}                  # number of atoms
ntypat        {1}                  # number of types of atoms
typat         {1}                  # type of atoms
znucl         {29}                 # charge Z of the nucleus
xred          {0.00 0.00 0.00}     # vectors (x) of atom positions
                                   #    in reduced coordinates
acell         {3*6.82+00}          # experimental lattice constant
rprim         {0.0  0.5  0.5       # real space primitive translations
               0.5  0.0  0.5
               0.5  0.5  0.0}
ngkpt         { 8  8  8}           # number of grid points for k points generation
ecut          {50.0}               # energy cutoff

# additional abinit input settings appended verbatim
verbatim      {
               occopt 6
               tsmear 0.025
               nband 12
              }
#-----------------------------------anaddb---------------------------------------
ngqpt         { 4  4  4}           # coarse q-grid
ng2qpt        {16 16 16}           # fine q-grid
#-----------------------------------dmdw-----------------------------------------
dmdw.nlanc    {16}                 # number of Lanczos poles
dmdw.ntemp    {400}                # number of temperature in T-grid
dmdw.temp     {1.0 400.0}          # temperature grid endpoints in K
dmdw.dispopt  {full}               # full DOS
```

A.2.4 Input for Cu Electron Self–energy and Spectral Function

```
name          {Cu-sf}
engine        {abinit}
property      {sf}
#-----------------------------------abinit---------------------------------------
pspfil        {29-Cu.LDA.fhi}      # list of pseudopotential files
natom         {1}                  # number of atoms
ntypat        {1}                  # number of types of atoms
typat         {1}                  # type of atoms
znucl         {29}                 # charge Z of the nucleus
xred          {0.00 0.00 0.00}     # vectors (x) of atom positions
                                   #    in reduced coordinates
acell         {3*6.82+00}          # experimental lattice constant
rprim         {0.0  0.5  0.5       # real space primitive translations
```

```
                0.5   0.0   0.5
                0.5   0.5   0.0}
ngkpt           {32 32 32}            # number of grid points for k points generation
ecut            {50.0}                # energy cutoff

# additional abinit input settings appended verbatim
verbatim        {
                 occopt 6
                 tsmear 0.025
                 nband 12
                }
#-----------------------------------anaddb-----------------------------------
ngqpt           { 4   4   4}          # coarse q-grid
ng2qpt          {16 16 16}            # fine q-grid
#-----------------------------------dmdw-------------------------------------
dmdw.nlanc      {16}                  # number of Lanczos poles
dmdw.temp       {                     # temperature(s) in K
                   10.0
                  100.0
                 1000.0
                }
dmdw.ek         {5.0}                 # quasiparticle energy in meV
dmdw.dispopt    {x}                   # direction of projected DOS
                                      #  arbitrary for cubic symmetry
```

A.3 Obtaining AI2PS

The AI2PS tool can be obtained through the FEFF Project group at the University of Washington. Contact information can be found at `http://feffproject.org`. Development on AI2PS is currently frozen, as its full capabilities have transferred to the recently developed workflow tool CORVUS, which we recommend using in place of AI2PS.

Appendix B

CORVUS SUPPLEMENTARY DETAILS

This appendix contains more in–depth information related to the content found in Chapter 5.

B.1 Running a One–shot Corvus Calculation

B.1.1 Supported Command Line Options

CORVUS supports the following command line arguments:

`-i/--input [filename]`

Specify input file (default `Corvus.inp`)

`-w/--workflow`

Specify pre–generated Workflow file

`-t/--target [string]`

Specify target property(ies) for automated Workflow generator

`--prefix [string]`

Specify prefix/label, strongly recommended being restricted to alphanumeribeingc and `[-_.]` characters, for subdirectories and output files (default='Corvus')

`-c/--checkpoints`

Enable checkpointing

`-r/--resume`

Load savefile ('[prefix].nest') and continue from last savepoint

`-s/--save [filename]`

Specify savefile other than default

`--parallelrun [command string]`

Specify command prefix for parallelized program commands, e.g., `"mpirun -np 240"`

Note:

- Loaded input file takes precedence over loaded saved state
- Loaded Workflow file takes precedence over loaded saved state
- Uses automatically generated Workflow from loaded target string if (1) no Workflow file loaded and (2) no savefile loaded
- Current developer option `-j/--jump [integer]` allows user to start the calculation at the specified Workflow step

B.1.2 Example Commands

The following are example commands for typical usage:

`run-corvus -t "specfn,selfen" -i Cu-sf.inp -c --parallelrun "mpirun -np 96"`

Generate and run a Workflow for producing electron spectral function and self–energy using input file `Cu-sf.inp` with checkpointing enabled using the standard `mpirun` command (for parallel jobs) with 96 processors

`run-corvus -cr --prefix old --input Cu-refresh.inp`

With checkpointing enabled, load up savefile with prefix `old` ($\Rightarrow$ `old.nest`) and resume where previous Workflow left off, refreshing the values of any properties listed in the input file `Cu-refresh.inp`

`run-corvus -w expansion.wf --parallelrun "aprun -np 240"`

Run pre–generated Workflow `expansion.wf` using 240 processors on a Cray compute cluster, which uses the `aprun` command for paralle jobs

B.2 Using Corvus Interactively

In addition to operating as a Workflow driver, CORVUS can be used as a software library for any of the workflow tools written by our developers, e.g., input file generators or output file parsers and translators, as they can be imported just as any other Python module. This also allows users to use the CORVUS tools on–the–fly through the Python interpreter (useful for tasks such as post hoc analysis or construction of Workflows, input files, etc.) or through user–made Python scripts. Appendix C gives a Python script example for building the relatively involved thermal expansion Workflow.

Suppose, for example, the user has output from a previous DMDW calculation (pre–CORVUS), and would like to do some extended analysis using the spectral function. This could be accomplished as follows:

```
from corvus.dmdw import readSpecFn

# DMDW output file
SFfilename = "dmdw_Akw.dat"

# readSpecFn returns energy-grid w and Re[A(w)], Im[A(w)] as column data
energies, realSF, imagSF = readSpecFn(SFfilename)

# Additional analysis, plotting, etc.
.
.
.
```

If the user wanted to insert a change of a property's value midway through a given Workflow:

```
from corvus.structure.structures import *
import pickle

# Load in pre-made Workflow from file
wf = pickle.load(open('myworkflow.wf', 'r'))

# Want to change temperature to 300.0 K
tempUpdate = Update('temp', newValue=300.0)

# Change temperature at step 3 (index = 2)
wf.addExchangeAt(2, tempUpdate)
```

Furthermore, by enabling checkpointing, the user ensures that the entire system property dictionary is saved to file. Since the savefile is formatted using Python's included `pickle`

module, the user can load up the saved state, either in the Python interpreter or in a Python script, and manipulate any of the saved properties at will. This is a more attractive alternative to the user having the additional step of parsing and translating output data themselves.

B.3 Example Input File

Below is an example input file for calculating the crystallographic Debye–Waller factors u^2 for the Ga and As atoms in bulk GaAs.

```
# ABINIT Input
diemac 12.9
ecut 60.0
natom 2
ntypat 2
typat {1 2}
znucl {31 33}
acell {3*10.68}
rprim {
    0.0  0.5  0.5
    0.5  0.0  0.5
    0.5  0.5  0.0
}
xred {-0.125 -0.125 -0.125
       0.125  0.125  0.125}
pspfiles {
    31-Ga-3d.LDA.fhi
    33-As.LDA.fhi
}
ngqpt {4 4 4}
ngkpt {16 16 16}

# DMDW Input
paths {
  2 # number of paths
    1 1 1.0 # one atom, index 1 (central Ga), cutoff radius (arbitrary)
    1 2 1.0 # one atom, index 2 (central As), cutoff radius (arbitrary)
}
tempgrid {3 100.0 300.0}
nlanc 16
```

Note that, as opposed to the AI2PS input files seen in Appendix A, the software selection (e.g., ABINIT and DMDW) is not present as an input parameter as it now falls under the responsibility of the Workflow.

B.4 Obtaining Corvus

The Corvus software package is maintained by the FEFF Project at the University of Washington and can be found, along with complete documentation, at the Corvus GitHub page (`https://github.com/feffgroup/Corvus`).

Appendix C

THERMAL EXPANSION SUPPLEMENTARY DETAILS

C.1 Generating Corvus Thermal Expansion Workflow

The following script generates the thermal expansion workflow, using the ABINIT and DMDW software packages, that generated the internal energy $U(a)$ and vibrational free energy $F(T, a)$ data as seen in Chapter 6. Line breaks are indicated by arrows.

```
#!/bin/env python

from corvus.structures import *
from corvus.dmdw import Dmdw
from corvus.abinit import Abinit
import pickle

wf = Workflow(['acell-grid','eint-grid','vfe-grid'], desc='Data␣for␣thermal␣expansion')

# Get lattice constant that minimized total internal energy
step1 = Exchange(Abinit, Abinit.requiredInputFor('aopt'), ['aopt'], desc='Use␣ABINIT␣to␣ ↩
    ↪ get␣optimized␣a')

# Replace intial lattice constant with optimized value
step2 = Update('acell', newToken='aopt')

# Generate symmetry-reduced list of q-points from NxMxL grid
step3 = Exchange(Abinit, Abinit.requiredInputFor('qptlist'), ['qptlist'], desc='Use␣ ↩
    ↪ ABINIT␣to␣get␣list␣of␣qpts')

# Create grid of lattice constants for parameter sweep
step4 = Exchange(Abinit, Abinit.requiredInputFor('acell-grid'), ['acell-grid'], desc='Get ↩
    ↪ ␣grid␣of␣a-values␣centered␣around␣current␣a')

# Use Loop to generate U(a)
UdymExchange = Exchange(Abinit, Abinit.requiredInputFor(['eint','dynmat','pdos','a2f']), ↩
    ↪ ['eint','dynmat','pdos','a2f'], desc='Use␣ABINIT␣to␣get␣internal␣energy␣and␣ ↩
    ↪ dynamical␣matrix')
step5 = Loop(UdymExchange, 'acell', gridToken='acell-grid')

# Use Loop to generate vfe(T,a)
vfeExchange = Exchange(Dmdw, Dmdw.requiredInputFor('vfe'), ['vfe'], desc='Use␣DMDW␣to␣get ↩
    ↪ ␣vibrational␣free␣energy␣per␣cell')
step6 = Loop(vfeExchange, ['acell','dynmat','pdos','a2f'], gridToken=['acell-grid',' ↩
```

```
    ↪ dynmat-grid','pdos-grid','a2f-grid'])

# Add list of steps to Workflow, print out, and save to file
wf.addExchangeList([step1,step2,step3,step4,step5,step6])
print wf
pickle.dump(wf, open('expansion.wf','w'))
```

The script also generates the following output, which lists the details of the thermal expansion Workflow:

```
Data for thermal expansion
1) Use ABINIT to get optimized a
   input: ['pspfiles', 'natom', 'ntypat', 'typat', 'znucl', 'acell', 'rprim',
'xred', 'diemac']
  output: ['aopt']
2) Update acell with value of aopt
3) Use ABINIT to get list of qpts
   input: ['pspfiles', 'natom', 'ntypat', 'typat', 'znucl', 'acell', 'rprim',
'xred', 'ngqpt']
  output: ['qptlist']
4) Get grid of a-values centered around current a
   input: ['acell']
  output: ['acell-grid']
5) Loop over acell with dynamic grid acell-grid
   input: ['acell', 'xred', 'qptlist', 'typat', 'ntypat', 'rprim', 'natom',
'znucl', 'pspfiles']
  output: ['eint-grid', 'dynmat-grid', 'pdos-grid', 'a2f-grid']
6) Loop over ['acell', 'dynmat', 'pdos', 'a2f'] with dynamic grids from
['acell-grid', 'dynmat-grid', 'pdos-grid', 'a2f-grid'] respectively
   input: ['dynmat', 'pdos', 'a2f']
  output: ['vfe-grid']
Required User Input: ['acell', 'xred', 'typat', 'ngqpt', 'ntypat', 'diemac',
'rprim', 'natom', 'znucl', 'pspfiles']
```

An analysis script to produce the thermal expansion $a(T)$ from the generated output is given in Section C.4.

C.2 Corvus Input for Thermal Expansion of Cu

The following Corvus input file was used in generating the thermal expansion data for copper as seen in Chapter 6.

```
# ABINIT Input
diemac 1.0e6
ecut 60.0
natom 1
ntypat 1
typat 1
```

```
znucl 29
acell {3*6.7333943148E+00}
rprim {
    0.0  0.5  0.5
    0.5  0.0  0.5
    0.5  0.5  0.0
}
xred {0.0 0.0 0.0}
verbatim {
    occopt 6
    tsmear 0.025
    nband 12
}
pspfiles {29-Cu.LDA.fhi}
ngqpt {4 4 4}
ngkpt {16 16 16}

# DMDW Input
nlanc 16
tempgrid {159 13.15 803.15}
```

C.3 Corvus Input for Thermal Expansion of GaAs

The following Corvus input file was used in generating the thermal expansion data for gallium arsenide as seen in Chapter 6.

```
# ABINIT Input
diemac 12.9
ecut 60.0
natom 2
ntypat 2
typat {1 2}
znucl {31 33}
acell {3*10.68}
rprim {
    0.0  0.5  0.5
    0.5  0.0  0.5
    0.5  0.5  0.0
}
xred {-0.125 -0.125 -0.125
       0.125  0.125  0.125}
pspfiles {
    31-Ga-3d.LDA.fhi
    33-As.LDA.fhi
}
ngqpt {4 4 4}
ngkpt {16 16 16}

# DMDW Input
```

```
tempgrid {159 13.15 803.15}
nlanc 16
```

C.4 Generating Thermal Expansion from Corvus Output

The following script was used to generate the thermal expansion $a(T)$ from the output produced by Corvus when running the Workflow above.

```
#!/bin/env python

import numpy as np
from math import pi

# COPPER SPECIFIC: electronic DOS at the Fermi energy in 1/J/cell
dosFermi = 7.16 * 2.293712e17

# FCC SPECIFIC: cell volume for volumetric <-> linear expansion
def V(a):
    return 0.25 * pow(a, 3.0)
def dVda(a):
    return 0.75 * pow(a, 2.0)

# Read in grid of lattice constants (Bohr)
output = eval(open('Corvus.acell-grid.out','r').read())
agrid = [float(a.split()[0]) for a in output]

# Read in grid of internal energies U and convert from Hartree/cell to J/mole-cell
output = eval(open('Corvus.eint-grid.out','r').read())
Ugrid = [(4.35974434*6.022140857*1.0e5)*u for u in output]
fit = np.poly1d(np.polyfit(agrid, Ugrid, 2))
Umin = fit(fit.deriv().r[0].real)

# Read in vibrational free energy (J/mole-cell) and temperature (K) grids
output = eval(open('Corvus.vfe-grid.out','r').read())
Tgrid = output[0]['Temp␣(K)']
vfegrid = [[None for j in range(len(agrid))] for i in range(len(Tgrid))]
for i in range(len(Tgrid)):
    for j in range(len(agrid)):
        vfegrid[i][j] = output[j]['VFE␣(J/mol-c)'][i]

# Sum U(a) and vfe(a,T) to get Helmholtz free energy F(a,T)
Fgrid = [[None for j in range(len(agrid))] for i in range(len(Tgrid))]
for i in range(len(Tgrid)):
    for j in range(len(agrid)):
        Fgrid[i][j] = Ugrid[j] + vfegrid[i][j]

# Read in Cv and convert to kB per atom
output = eval(open('Corvus.Cv-grid.out','r').read())
Cvgrid = [[None for j in range(len(agrid))] for i in range(len(Tgrid))]
```

```
for i in range(len(Tgrid)):
    for j in range(len(agrid)):
        Cvgrid[i][j] = output[j]['Cv␣(J/K/mol)'][i] / (1.38064852*6.022140857)

aofT = []
FofT = []
GofT = []
BofT = []
Pgrid = [[None for j in range(len(agrid))] for i in range(len(Tgrid))]
for iT in range(len(Tgrid)):
    # Fit temperature slices F(a,T_i) to a quartic and minimize to get a(T_i)
    fit = np.poly1d(np.polyfit(agrid, Fgrid[iT], 4))
    deriv = fit.deriv()
    amin = filter(lambda a: agrid[0] < a < agrid[-1], deriv.r)[0].real
    aofT.append(amin)
    # Calculate P = -dF/dV using V(a) = a^3/4 for fcc cell
    for ia in range(len(agrid)):
        Pgrid[iT][ia] = -1.0 * deriv(agrid[ia]) / dVda(agrid[ia])
    # Calculate Helmholtz and Gibbs free energies at a(T)
    Pfit = np.poly1d(np.polyfit(agrid, Pgrid[iT], 4))
    FofT.append((fit(aofT[iT]) - Umin)/1000.0) # kJ/mole-cell
    PV = (Pfit(aofT[iT]) * V(aofT[iT]))/1000.0 # kJ/mole-cell
    GofT.append(FofT[iT] + PV) # kJ/mole-cell
    # Calculate isothermal bulk modulus using B = -V^-1 dP/dV
    Pderiv = Pfit.deriv()
    BofT.append(-1.0 * V(aofT[iT]) * Pderiv(aofT[iT]) / dVda(aofT[iT]))

# Fit a(T) to a 10th order polynomial and calculate linear expansion coefficient
alpha = [x/aofT[iT] for x in np.gradient(np.array(aofT), np.gradient(np.array(Tgrid)))]

# Write results to file with appropriate units
outfile = open('expansion.dat','w')
for iT in range(len(Tgrid)):
    aang = aofT[iT]*0.529177211 # Angstrom
    alphaV = 3.0 * alpha[iT] # 1/K
    B = BofT[iT] * (6.7483345e30/6.022140857e23/1.0e9) # GPa
    BV = B * V(aang) * (1.0e-30*1.0e9/1.38064852e-23) # kB
    Cvfit = np.poly1d(np.polyfit(agrid, Cvgrid[iT], 4))
    Cv = Cvfit(aofT[iT]) # kB
    Celec = (pi*pi/3.0) * (1.38064852e-23 * dosFermi)* Tgrid[iT] # kB
    Cp = Cv + Celec + alphaV*alphaV * BV * Tgrid[iT] # kB
    # Units: K, Angstrom, 1/K, GPa, kB, kB, kB
    cols = [Tgrid[iT], aang, alpha[iT], B, Cv, Celec, Cp, FofT[iT], GofT[iT]]
    outfile.write('␣␣'.join(map(str, cols)) + '\n')
```

VITA

Shauna Story graduated from the University of Arizona with a B. S. in Physics, Math, and Astronomy in 2010. She was a graduate student in Physics at the University of Washington, Seattle from 2010 to 2015.

BIBLIOGRAPHY

[1] Joseph John Thomson. Xl. cathode rays. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 44(269):293–316, 1897.

[2] Paul Drude. Zur elektronentheorie der metalle. *Annalen der Physik*, 306(3):566–613, 1900.

[3] Paul Drude. Zur elektronentheorie der metalle; ii. teil. galvanomagnetische und thermomagnetische effecte. *Annalen der Physik*, 308(11):369–402, 1900.

[4] Arnold Sommerfeld. Zur elektronentheorie der metalle auf grund der fermischen statistik. *Zeitschrift für Physik*, 47(1-2):1–32, 1928.

[5] M. Eckert. Max von laue and the discovery of x-ray diffraction in 1912. *Annalen der Physik*, 524(5):A83–A85, 2012.

[6] William Lawrence Bragg. The diffraction of short electromagnetic waves by a crystal. In *Proceedings of the Cambridge Philosophical Society*, volume 17, page 4, 1913.

[7] Felix Bloch. Über die quantenmechanik der elektronen in kristallgittern. *Zeitschrift für physik*, 52(7-8):555–600, 1929.

[8] Neil W Ashcroft and N David Mermin. Solid state phys. *Saunders, Philadelphia*, page 293, 1976.

[9] Göran Grimvall. *The electron-phonon interaction in metals*. North-Holland Amsterdam, 1981.

[10] R. M. Nicklow, G. Gilat, H. G. Smith, L. J. Raubenheimer, and M. K. Wilkinson. Phonon frequencies in copper at 49 and 298°k. *Phys. Rev.*, 164(3):922–928, Dec 1967.

[11] A. D. B. Woods. Lattice dynamics of tantalum. *Phys. Rev.*, 136:A781–A783, Nov 1964.

[12] Y. Nakagawa and A. D. B. Woods. Lattice dynamics of niobium. *Phys. Rev. Lett.*, 11:271–274, Sep 1963.

[13] R. Colella and B. W. Batterman. X-ray determination of phonon dispersion in vanadium. *Phys. Rev. B*, 1:3913–3921, May 1970.

[14] F. D. Vila, J. J. Rehr, H. H. Rossner, and H. J. Krappe. Theoretical x-ray absorption debye-waller factors. *Phys. Rev. B*, 76(1):014301, July 2007.

[15] Peter Deuflhard and Andreas Hohmann. *Numerical Analysis: A First Course in Scientific Computation*. Walter de Gruyter & Co., 1995.

[16] AV Poiarkova and JJ Rehr. Multiple-scattering x-ray-absorption fine-structure debye-waller factor calculations. *Physical Review B*, 59(2):948, 1999.

[17] AV Poiarkova and JJ Rehr. *J. Synchrotron Radiat.*, 8:313, 2001.

[18] HJ Krappe and HH Rossner. Bayes-turchin approach to x-ray absorption fine structure data analysis. *Physical Review B*, 66(18):184303, 2002.

[19] M. Newville. private communication.

[20] Göran Grimvall. *Thermophysical properties of materials*. Elsevier, 1986.

[21] W. L. McMillan. Transition temperature of strong-coupled superconductors. *Phys. Rev.*, 167(2):331, 1968.

[22] G Grimvall. Many-body corrections to the effective electron mass in the noble metals. *Phys. kondens. Materie.*, 11(4):279–285, 1970.

[23] Philip B Allen. Neutron spectroscopy of superconductors. *Phys. Rev. B*, 6(7):2577, 1972.

[24] S Yu Savrasov and D Yu Savrasov. Electron-phonon interactions and related physical properties of metals from linear-response theory. *Phys. Rev. B*, 54(23):16487, 1996.

[25] P. B Allen. The electron-phonon coupling constant λ. In C. P. Poole, editor, *Handbook of Superconductivity*, chapter 9, pages 478–483. Academic Press, 1999.

[26] P. B. Allen and B. Mitrović. Theory of superconducting tc. In Henry Ehrenreich, Frederick Seitz, and David Turnbull, editors, *Solid State Physics*, volume 37 of *Solid State Physics*, pages 1–92. Academic Press, 1982.

[27] Asier Eiguren and Claudia Ambrosch-Draxl. Complex quasiparticle band structure induced by electron-phonon interaction: Band splitting in the 1×1H/W(110) surface. *Phys. Rev. Lett.*, 101:036402, Jul 2008.

[28] S. Engelsberg and J. R. Schrieffer. Coupled electron-phonon system. *Phys. Rev.*, 131:993–1008, Aug 1963.

[29] Philippe Nozieres. *Theory of interacting Fermi systems*. Addison-Wesley, 1964.

[30] AB Migdal. Interaction between electrons and lattice vibrations in a normal metal. *Sov. Phys. JETP*, 7(6):996–1001, 1958.

[31] Lars Hedin. On correlation effects in electron spectroscopies and the gw approximation. *J. Phys.: Condens. Matter*, 11(42):R489, 1999.

[32] J. J. Kas, J. J. Rehr, and L. Reining. Cumulant expansion of the retarded one-electron green function. *Physical Review B*, 90(8):085112, 2014.

[33] D.A. McQuarrie. *Statistical mechanics*. Harper's chemistry series. HarperCollins, 1976.

[34] Daniel V Schroeder. *An introduction to thermal physics*, volume 60. Addison Wesley New York, 2000.

[35] Brent Fultz. Vibrational thermodynamics of materials. *Progress in Materials Science*, 55(4):247–352, 2010.

[36] H. M. Lawler, J. J. Rehr, F. Vila, S. D. Dalosto, E. L. Shirley, and Z. H. Levine. Optical to uv spectra and birefringence of sio_2 and tio_2: First-principles calculations with excitonic effects. *Phys. Rev. B*, 78:205108, Nov 2008.

[37] J. Vinson, J. J. Rehr, J. J. Kas, and E. L. Shirley. Bethe-salpeter equation calculations of core excitation spectra. *Phys. Rev. B*, 83:115106, Mar 2011.

[38] J. Vinson, J. J. Kas, F. D. Vila, J. J. Rehr, and E. L. Shirley. Theoretical optical and x-ray spectra of liquid and solid h_2o. *Phys. Rev. B*, 85:045101, Jan 2012.

[39] J. Vinson and J. J. Rehr. *Ab initio* bethe-salpeter calculations of the x-ray absorption spectra of transition metals at the l-shell edges. *Phys. Rev. B*, 86:195135, Nov 2012.

[40] Xavier Gonze. First-principles responses of solids to atomic displacements and homogeneous electric fields: Implementation of a conjugate-gradient algorithm. *Phys. Rev. B*, 55(16):10337, 1997.

[41] Xavier Gonze and Changyol Lee. Dynamical matrices, born effective charges, dielectric permittivity tensors, and interatomic force constants from density-functional perturbation theory. *Phys. Rev. B*, 55(16):10355, 1997.

[42] X Gonze, J. M. Beuken, R Caracas, F Detraux, M Fuchs, G.M. Rignanese, L Sindic, Matthieu Verstraete, G Zerah, F Jollet, et al. First-principles computation of material properties: the abinit software project. *Comput. Mat. Sci.*, 25(3):478–492, 2002.

[43] Xavier Gonze. A brief introduction to the abinit software package. *Z. Kristallogr.*, 220(5/6/2005):558–562, 2005.

[44] The ABINIT code is a common project of the Université Catholique de Louvain, Corning Incorporated, and other contributors (URL `http://www.abinit.org`).

[45] John J Rehr, Joshua J Kas, Fernando D Vila, Micah P Prange, and Kevin Jorissen. Parameter-free calculations of x-ray spectra with feff9. *Phys. Chem. Chem. Phys.*, 12(21):5503–5513, 2010.

[46] More information: `http://www.feffproject.org/`.

[47] P. Fornasini, S. a Beccara, G. Dalba, R. Grisenti, A. Sanson, M. Vaccari, and F. Rocca. Extended x-ray-absorption fine-structure measurements of copper: Local dynamics, anharmonicity, and thermal expansion. *Phys. Rev. B*, 70:174301, Nov 2004.

[48] Shobhana Narasimhan and Stefano De Gironcoli. Ab initio calculation of the thermal properties of cu: Performance of the lda and gga. *Phys. Rev. B*, 65(6):064302, 2002.

[49] P. A. Flinn, G. M. McManus, and J. A. Rayne. Effective x-ray and calorimetric debye temperature for copper. *Phys. Rev.*, 123:809–812, Aug 1961.

[50] M Schowalter, A Rosenauer, JT Titantah, and D Lamoen. Computation and parametrization of the temperature dependence of debye-waller factors for group iv, iii-v and ii-vi semiconductors. *Acta Crystallographica Section A: Foundations of Crystallography*, 65(1):5–17, 2008.

[51] D Strauch and B Dorner. Phonon dispersion in gaas. *Journal of Physics: Condensed Matter*, 2(6):1457, 1990.

[52] Gorelik S S Shumskii M G, Bublik V T and Gurevich M A. *Soviet Phys. - Crystallogr.*, 16:674, 1972.

[53] M. Gomm, 1977. Dissertation Erlangen-Nürnberg, *unpublished*.

[54] U. Pietsch. X-ray bond charge in gaas and insb. *physica status solidi (b)*, 103(1):93–100, 1981.

[55] S. M. Story, J. J. Kas, F. D. Vila, M. J. Verstraete, and J. J. Rehr. Cumulant expansion for phonon contributions to the electron spectral function. *Phys. Rev. B*, 90:195135, Nov 2014.

[56] T Cuk, DH Lu, XJ Zhou, Z-X Shen, TP Devereaux, and N Nagaosa. A review of electron–phonon coupling seen in the high-tc superconductors by angle-resolved photoemission studies (arpes). *Physica Status Solidi (b)*, 242(1):11–29, 2005.

[57] Marco Bernardi, Maurizia Palummo, and Jeffrey C. Grossman. Extraordinary sunlight absorption and one nanometer thick photovoltaics using two-dimensional monolayer materials. *Nano Lett.*, 13(8):3664–3670, 2013.

[58] Diana Y Qiu, H Felipe, and Steven G Louie. Optical spectrum of mos 2: Many-body effects and diversity of exciton states. *Phys. Rev. Lett.*, 111(21):216805, 2013.

[59] Mark S. Hybertsen and Steven G. Louie. Electron correlation in semiconductors and insulators: Band gaps and quasiparticle energies. *Phys. Rev. B*, 34:5390–5413, Oct 1986.

[60] Giovanni Onida, Lucia Reining, and Angel Rubio. Electronic excitations: density-functional versus many-body green's-function approaches. *Rev. Mod. Phys.*, 74:601–659, Jun 2002.

[61] P Steiner, H Höchst, S Hüfner, L Ley, and M Cardona. Photoemission in solids ii. *Topics Appl. Phys.*, 27:349, 1979.

[62] F Offi, WSM Werner, M Sacchi, P Torelli, M Cautero, G Cautero, A Fondacaro, S Huotari, G Monaco, G Paolicelli, et al. Comparison of hard and soft x-ray photoelectron spectra of silicon. *Phys. Rev. B*, 76(8):085422, 2007.

[63] O. Gunnarsson, V. Meden, and K. Schönhammer. Corrections to migdal's theorem for spectral functions: A cumulant treatment of the time-dependent green's function. *Phys. Rev. B*, 50:10462–10473, Oct 1994.

[64] Ryogo Kubo. Generalized cumulant expansion method. *J. Phys. Soc. Jpn.*, 17:1100, 1962.

[65] Lars Hedin. Effects of recoil on shake-up spectra in metals. *Physica Scripta*, 21(3-4):477, 1980.

[66] F. Aryasetiawan, L. Hedin, and K. Karlsson. Multiple plasmon satellites in na and al spectral functions from *Ab Initio* cumulant expansion. *Phys. Rev. Lett.*, 77:2268–2271, Sep 1996.

[67] Matteo Guzzo, Giovanna Lani, Francesco Sottile, Pina Romaniello, Matteo Gatti, Joshua J Kas, John J Rehr, Mathieu G Silly, Fausto Sirotti, and Lucia Reining. Valence electron photoemission spectrum of semiconductors: Ab initio description of multiple satellites. *Phys. Rev. Lett.*, 107(16):166401, 2011.

[68] Michele Casula, Alexey Rubtsov, and Silke Biermann. Dynamical screening effects in correlated materials: Plasmon satellites and spectral weight transfers from a green's function ansatz to extended dynamical mean field theory. *Phys. Rev. B*, 85(3):035115, 2012.

[69] Philip B Allen and Volker Heine. Theory of the temperature dependence of electronic band structures. *J. Phys. Chem.*, 9(12):2305, 1976.

[70] PB Allen and M Cardona. Theory of the temperature dependence of the direct gap of germanium. *Phys. Rev. B*, 23(4):1495, 1981.

[71] PB Allen and M Cardona. Temperature dependence of the direct gap of si and ge. *Phys. Rev. B*, 27(8):4760, 1983.

[72] Lars Hedin and Stig Lundqvist. Effects of electron-electron and electron-phonon interactions on the one-electron states of solids. *Solid State Physics*, 23:1–181, 1970.

[73] B.I. Lundqvist. Single-particle spectrum of the degenerate electron gas. *Phys. kondens. Materie.*, 6(3):193–205, 1967.

[74] B.I. Lundqvist. Single-particle spectrum of the degenerate electron gas - ii. numerical results for electrons coupled to plasmons. *Phys. kondens. Materie.*, 6(3):206–217, 1967.

[75] J. J. Kas, A. P. Sorini, M. P. Prange, L. W. Cambell, J. A. Soininen, and J. J. Rehr. Many-pole model of inelastic losses in x-ray absorption spectra. *Phys. Rev. B*, 76:195116, Nov 2007.

[76] See `http://www.abinit.org`.

[77] MPAT Methfessel and AT Paxton. High-precision sampling for brillouin-zone integration in metals. *Phys. Rev. B*, 40(6):3616, 1989.

[78] PM Chaikin, G Arnold, and PK Hansma. Phonon structure in the tunneling characteristics of thin proximity effect sandwiches. *J. Low Temp. Phys.*, 26(1-2):229–264, 1977.

[79] Edward L Wolf. *Principles of electron tunneling spectroscopy*. Oxford Univ. Press, 1985.

[80] Philip B. Allen. Empirical electron-phonon λ values from resistivity of cubic metallic elements. *Phys. Rev. B*, 36:2920–2923, Aug 1987.

[81] SD Brorson, A Kazeroonian, JS Moodera, DW Face, TK Cheng, EP Ippen, MS Dresselhaus, and G Dresselhaus. Femtosecond room-temperature measurement of the electron-phonon coupling constant γ in metallic superconductors. *Phys. Rev. Lett.*, 64(18):2172, 1990.

[82] BA McDougall, T Balasubramanian, and E Jensen. Phonon contribution to quasiparticle lifetimes in cu measured by angle-resolved photoemission. *Phys. Rev. B*, 51(19):13891, 1995.

[83] A. Eiguren, B. Hellsing, F. Reinert, G. Nicolay, E. V. Chulkov, V. M. Silkin, S. Hüfner, and P. M. Echenique. Role of bulk and surface phonons in the decay of metal surface states. *Phys. Rev. Lett.*, 88:066805, Jan 2002.

[84] F. Reinert, B. Eltner, G. Nicolay, D. Ehm, S. Schmidt, and S. Hüfner. Electron-phonon coupling and its evidence in the photoemission spectra of lead. *Phys. Rev. Lett.*, 91:186406, Oct 2003.

[85] Matthieu J Verstraete, Marc Torrent, François Jollet, Gilles Zérah, and Xavier Gonze. Density functional perturbation theory with spin-orbit coupling: Phonon band structure of lead. *Phys. Rev. B*, 78(4):045119, 2008.

[86] R. Heid, K.-P. Bohnen, I. Yu. Sklyadneva, and E. V. Chulkov. Effect of spin-orbit coupling on the electron-phonon interaction of the superconductors pb and tl. *Phys. Rev. B*, 81:174527, 2010.

[87] RC Dynes. Mcmillan's equation and the t_c of superconductors. *Solid State Communications*, 10(7):615–618, 1972.

[88] P Morel and PW Anderson. Calculation of the superconducting state parameters with retarded electron-phonon interaction. *Phys. Rev.*, 125(4):1263, 1962.

[89] C. Kittel. *Introduction to Solid State Physics*. Wiley, 2005.

[90] G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky. AiiDA: Automated Interactive Infrastructure and Database for Computational Science. *ArXiv e-prints*, April 2015.

[91] Michael Wilde, Mihael Hategan, Justin M Wozniak, Ben Clifford, Daniel S Katz, and Ian Foster. Swift: A language for distributed parallel scripting. *Parallel Computing*, 37(9):633–652, 2011.

[92] Ola Spjuth, Tobias Helmus, Egon L Willighagen, Stefan Kuhn, Martin Eklund, Johannes Wagener, Peter Murray-Rust, Christoph Steinbeck, and Jarl ES Wikberg. Bioclipse: an open source workbench for chemo-and bioinformatics. *BMC bioinformatics*, 8(1):59, 2007.

[93] Jeremy Goecks, Anton Nekrutenko, James Taylor, et al. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol*, 11(8):R86, 2010.

[94] Ilkay Altintas, Chad Berkley, Efrat Jaeger, Matthew Jones, Bertram Ludascher, and Steve Mock. Kepler: an extensible system for design and execution of scientific workflows. In *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, pages 423–424. IEEE, 2004.

[95] Salman AlSairafi, Filippia-Sofia Emmanouil, Moustafa Ghanem, Nikolaos Giannadakis, Yike Guo, Dimitrios Kalaitzopoulos, Michelle Osmond, Anthony Rowe, Jameel Syed, and Patrick Wendel. The design of discovery net: Towards open grid services for knowledge discovery. *International Journal of High Performance Computing Applications*, 17(3):297–315, 2003.

[96] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[97] S. I. Novikova. *Fizika Tverdogo Tela*, 3:178, 1961.

[98] TF Smith and GK White. The low-temperature thermal expansion and gruneisen parameters of some tetrahedrally bonded solids. *Journal of Physics C: Solid State Physics*, 8(13):2031, 1975.

[99] M Leszczynski, VB Pluzhnikov, A Czopnik, J Bak-Misiuk, and T Slupinski. Thermal expansion of gaas: Te and algaas: Te at low temperatures. *Journal of applied physics*, 82(9):4678–4680, 1997.

[100] W Miller, CW Smith, DS Mackenzie, and KE Evans. Negative thermal expansion: a review. *Journal of materials science*, 44(20):5441–5451, 2009.

[101] John P. Perdew, Adrienn Ruzsinszky, Gábor I. Csonka, Oleg A. Vydrov, Gustavo E. Scuseria, Lucian A. Constantin, Xiaolan Zhou, and Kieron Burke. Restoring the density-gradient expansion for exchange in solids and surfaces. *Phys. Rev. Lett.*, 100:136406, Apr 2008.

[102] John P. Perdew, Adrienn Ruzsinszky, Gábor I. Csonka, Oleg A. Vydrov, Gustavo E. Scuseria, Lucian A. Constantin, Xiaolan Zhou, and Kieron Burke. Erratum: Restoring the density-gradient expansion for exchange in solids and surfaces. *Phys. Rev. Lett.*, 102:039902, Jan 2009.

[103] FC Nix and D MacNair. The thermal expansion of pure metals: copper, gold, aluminum, nickel, and iron. *Physical Review*, 60(8):597, 1941.

[104] *AIP Handbook of Physics*. McGraw-Hill, 3rd edition, 1982.

[105] Y. Austin Chang and Ralph Hultgren. The dilation contribution to the heat capacity of copper and α-brass at elevated temperatures. *The Journal of Physical Chemistry*, 69(12):4162–4165, 1965.

[106] T Kiss, T Shimojima, K Ishizaka, A Chainani, T Togashi, T Kanai, XY Wang, CT Chen, S Watanabe, and S Shin. A versatile system for ultrahigh resolution, low temperature, and polarization dependent laser-angle-resolved photoemission spectroscopy. *The Review of scientific instruments*, 79(2 Pt 1):023106–023106, 2008.

[107] Ondrej L Krivanek, Tracy C Lovejoy, Niklas Dellby, and RW Carpenter. Monochromated stem with a 30 mev-wide, atom-sized electron probe. *Microscopy*, page dfs089, 2013.

[108] Kevin Jorissen, Fernando D Vila, and John J Rehr. A high performance scientific cloud computing environment for materials simulations. *Computer Physics Communications*, 183(9):1911–1919, 2012.

[109] Koshi Takenaka. Negative thermal expansion materials: technological key for control of thermal expansion. *Science and Technology of Advanced Materials*, 13(1):013001, 2012.