New Perspectives on *Ab Initio* Calculation and Physical Insights Gained Through Linkage to Continuum Theories

by

Sohrab Ismail-Beigi

Submitted to the Department of Physics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Physics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2000

Author .....
Department of Physics
April 28th, 2000

Certified by .....
Tomás A. Arias
Professor
Thesis Supervisor

Certified by .....
John D. Joannopoulos
Professor
Thesis Supervisor

Accepted by .....
Thomas J. Greytak
Professor, Associate Department Head for Education

New Perspectives on *Ab Initio* Calculation and Physical Insights Gained Through Linkage to Continuum Theories

by

Sohrab Ismail-Beigi

Submitted to the Department of Physics
on April 28th, 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Physics

Abstract

We explore the use of *ab initio*, density-functional methods for the study of large-scale materials problems. Three examples are presented: (i) the interplay of surface and edge reconstructions in long silicon nanowires, where we examine the effects on electrical and mechanical properties; (ii) the calculation of solvation effects based on *ab initio* dielectric models, where we derive the dielectric treatment as a coarse-grained molecular description, apply our method to the hydrolysis reaction of methylene chloride, and examine simplifications to our *ab initio* method; and (iii) the study of screw dislocation cores in bcc molybdenum and tantalum, where we find core structures contrary to those commonly accepted and barriers to dislocation motion in better agreement with experiment.

Methodologically, we present a new matrix-based, algebraic formalism for *ab initio* calculations which modularizes and isolates the roles played by the basis set, the energy functional, the algorithm used to achieve self-consistency, and the computational kernels. Development and implementation of new techniques amounts to derivation and transcription of algebraic expressions. Modularizing the computational kernels yields portable codes that are easily optimized and parallelized, and we present highly efficient kernels for scalar, shared, and distributed memory computers.

We conclude with an analytical study of the spatial locality of the single-particle density matrix in solid-state systems. This locality reflects the localization of electronic states and is essential for real-space and $O(N)$ methods. We derive new behavior for this spatial range contrary to previous proposals, and we verify our findings in model semiconductors, insulators, and metals.

Thesis Supervisor: Tomás A. Arias
Title: Professor

Thesis Supervisor: John D. Joannopoulos
Title: Professor

Acknowledgments

"No man is an island, entire of itself." – John Donne, 1624

First and foremost, Prof. Arias has been an ideal thesis advisor for me. Ever since our first interaction in the fall of 1993, he's always communicated the excitement and joy of doing physics regarding both the big questions as well as the details of research. I have learned immeasurably from interacting with him regardless of whether the subject concerned basic physics, finding an interesting problem or question to answer, learning to separate relevant versus irrelevant issues, the art of writing and communicating one's work, or the highly non-trivial task of giving a good talk. He always allowed me to explore questions that I find interesting without any prior bias while at the same time providing the correct overall guidance so as to avoid getting lost on a tangent.

Next are friends who have helped make the last six years here an enjoyable experience. Gabriele Migliorini and Dirk Lumma are among the first students I met here at MIT and, although our first group activities were late night (more like early morning) problem sets for 8.333 and 8.334, they have both been wonderful companions and friends. I would like to thank Dirk especially for his friendship, helpful advice, criticism, and reality-checks on all the issues that enter into living a life. Alkan Kabakçioglu was the first CMT graduate student I got to know, and *every* occasion when we did something together was enjoyable and relaxed; I very much like his easy-going attitude, intelligence, and excellent sense of humor. Dicle Yesilleten, fellow group-member and flatmate for five years, was the second Turkish person I met at MIT: it has been good to have someone fun, and as close as a sister for a house mate. I met Mikhail Brodsky through Alkan, and I have learned much discussing physics, work, world affairs, and life in general and have been impressed by his mature, unique, and usually humorous viewpoint on these issues.

My officemates Gabor Csanyi, Torkel Engeness, and Darren Segall were certainly the most mess-tolerant bunch I've met! I've enjoyed the interactions, collaborations, and especially our political discussions. Veteran DFT++ collaborators on the CMT

corridor include Gabor, who has provided wonderfully helpful discussion and criticism about programming (the good, the bad, and the *very* ugly), Tairan Wang, who put much work into the project, and Nikolaj Moll, who has provided good advice right next door.

Although I met her only two and a half years ago, I owe much to Yaoda Xu. She has made my life here so much fuller than I could have imagined it to be, and I've found myself enjoying my life and work more thoroughly while being more productive than ever. Her sense of humor and her warmth have helped me relax and learn to be myself. For me, our dancing has always been a metaphor for far more, with its pushes, pulls, closeness, and exuberance, and I hope it will continue.

Finally, my family rightly deserves the greatest thanks for making my studies here possible. My sisters have been very supportive by providing me with love, humor, and advice, and for keeping in touch with me even when I was aloof. As for my parents, I still remember clearly the afternoon when I had to decide on my graduate career and when they urged me to go to MIT and be a theorist because that was what I *wanted* to do without regard as to whether there *might* be better prospects if I changed fields. This is but one example of the marvelous guidance and support that they have given me, and I wish everyone fortunate enough to have such parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Density-functional theory (DFT) [43, 58] provides a rigorous, first principles approach to finding the ground-state energy and single-particle properties of any electronic system. When combined with the local-density approximation to the exchange-correlation energy [58], one has an accurate and computationally tractable method for calculating the Born-Oppenheimer energy and atomic forces for any atomic configuration. The method has proven highly successful in the study of atomic, molecular, and solid-state systems [76, 77].

The works presented in this thesis focus on using *ab initio* DFT methods to study large-scale materials problems. When doing so, two distinct but related issues arise, corresponding to the two topics in the thesis title.

(I) The first issue regards how one can link *ab initio*, quantum mechanical modeling to appropriate continuum theories. The philosophical viewpoint here begins with the fact that although computational power has increased dramatically in recent times, we still can not simulate macroscopic collections of atoms quantum mechanically. In fact, such a treatment is generally not useful since continuum theories already exist which describe the coarse-grained behavior of large portions of a material. The true value of first principles methods arises in studying regions in the material where important localized defects exist and where the effective theories are no longer applicable. Therefore, a fruitful approach will apply *ab initio* methods to study the defects and then link these results to continuum descriptions so as both to incor-

porate long-range effects as well as extract key macroscopic properties of the total system.

My work includes three examples of such an approach. Chapter 2 presents a study of long, narrow silicon nanobars with cross sections $\sim$ 30 Å, length scales which experimental lithographic techniques will soon access. The aim is to see the effect of atomic surface reconstructions on the electronic and elastic properties of the bars. Two competing reconstructions are found: one is energetically favored by the surfaces, the other is favored by the edges of the bars, and a size-dependent phase transition between the two is predicted to occur as the cross section of the bar increases. The two reconstructions are predicted to have very different electrical properties (one is insulating whereas the other is essentially metallic), but differences in long-range vibrational properties are not significant [47].

In Chapter 3, I turn to the study of solvation effects for chemical reactions in water. Clearly, quantum mechanical methods must be used to describe the reactants since electronic states are modified in a reaction. In principle, the solvent can also be modeled by a large set of molecules simulated quantum-mechanically through molecular dynamics, but this is both computationally expensive and largely unnecessary when the solvent is chemically inert, which is the case I deal with. In my work, a continuum dielectric models the solvent, which rigorously provides the correct thermodynamics far from the reactants, and which must have some adjustable details close by. I present a careful and detailed derivation of the steps and approximations that coarse-grain the molecular description to yield a continuum dielectric. Next, the dielectric model is applied to study the solvation of the hydrolysis of methylene chloride. The predicted solvation energies are in promising agreement with experimental results. In addition, a systematic study and comparison is made among some of the most popular dielectric models used in the literature.

Finally, in Chapter 4 I describe an *ab initio* study of dislocation core structures in bcc molybdenum and tantalum. Dislocations control macroscopic plasticity, and the atomic arrangement of the dislocation core is the key to finding the size and nature of the barriers that must be overcome to begin plastic deformation. Again,

first principles methods are required to treat the core since the atomic arrangement is severely distorted. Regions far from the core can be treated using elasticity theory. In this work, I find core structures with geometries contrary to what has hitherto been believed to be correct for bcc screw dislocations based on interatomic potentials. Predictions for the stresses needed to move screw dislocations are also in closer agreement with experimental values [50]. As a technical matter, when studying dislocations within periodic boundary conditions, there are an infinite array of interacting dislocations. The infinite sum of interactions can be regularized within continuum elasticity theory, and the resulting energy can be subtracted off, allowing the extraction of the core energetics alone. Furthermore, by judicious choice of unit cell and arrangement of dislocations, we are able to perform our calculations reliably using an order of magnitude fewer atoms than employed in previous studies using classical potentials.

(II) The second key issue dealt with in this thesis involves finding strategies for large-scale calculations to be performed effectively. In reality, such computations are run on a variety of computational platforms and they use a variety of physical approximations (e.g. the local-density approximation with or without spin, gradient corrections to the former, dielectrics, etc.). This question is *not* merely one of computer programming. A formalism is required that modularizes and compartmentalizes the physical and computational issues so that an inordinate amount of time is not spent on computer coding when trying a new approximation or changing computational platforms.

Chapter 5 develops a new formalism, DFT++, that is applicable to any single-particle, quantum mechanical theory [49]. The formulation is basis-set independent so that one can focus attention on the physics of a given approximation and the conceptual structure of the computation. The formalism organizes and isolates the computational issues into a small number of modules, so that the physics inherent in an energy functional is placed clearly in the foreground. As an additional practical benefit, this modularity has led to readable and easily extensible computer code that obtains very good computational performance with small investments of time.

Finally, in the future one expects to simulate ever larger systems. Traditionally, the electronic wave functions of a system are expanded in terms of all the basis functions used in the calculation, which, while straightforward, leads to a computation whose burden scales with the cube of the number of electrons. An alternative approach, based on the locality of electronic states, expands each wave function first in terms of localized Wannier functions, and then each Wannier function is expanded in terms of a spatially localized basis set. This approach leads to great computational savings as one can truncate the Wannier functions outside of some range (for some desired accuracy), and the approach has a computational burden that scales only linearly with the number of electrons.

However, for such a scheme to be practical, one must first understand the degree of locality of the Wannier functions or, equivalently, the electronic density matrix of the system since the prefactor of the linear scaling depends quite strongly on this locality. The final chapter of this thesis, Chapter 6, explores the locality of density matrices and Wannier functions in solids [48] and provides useful estimates and bounds of the locality in terms of key system parameters such as the lattice constant and band gap for insulators or the temperature and Fermi level for metals.

Chapter 2

Interplay of Surface and Edge Physics in Silicon Nanoresonators

As our understanding of bulk and surface properties of materials matures, the physics of nanoscale structures opens new fundamental questions. What are the ground state structures of nanoscale collections of matter and to what extent can they be predicted by simply scaling down bulk and micron-level behavior or scaling up the behavior of small clusters? What new considerations must be taken into account? Do nanoscale structures exhibit fundamentally different electronic or mechanical properties due to the large fraction of atoms at surfaces *and* edges, i.e., at the intersection of two surfaces? What are the the effects of nanoscale structure on the reconstruction of surfaces?

Clearly, for sufficiently small structures, edges become important. One key issue is the identification of the scale at which this happens and in particular whether edges come into play for anything larger than a small cluster of atoms. Also, one must determine the phenomena by which this importance manifests itself. In this chapter, we use *ab initio* calculations to show that edge effects indeed become important in silicon on length-scales on the order of a few nanometers, only a factor of two or three times smaller than what can be achieved by recent technology [95]. We find that the presence of edges has a profound effect on the reconstruction on the surface of a structure and thereby its electronic structure. Specifically, we predict that for

long bars along the [001] direction, the edges drive a surface reconstruction transition from the familiar "2x1" family to the "c(2x2)" family [46] at a cross section of 3 nm $\times$ 3 nm.

Long "bars" (as illustrated in Figure 2-1) provide the ideal laboratory for studying the nature of edges and their interaction with surfaces. An isolated edge implies an infinite system, whereas a bar consists of a series of edges bounding a finite area and thus may be studied within the supercell framework. In addition, such structures are studied experimentally. Using lithographic techniques, long bars of silicon can be created in the form of suspended bridges between bulk silicon supports [93]. The heat flow and vibrational properties of such structures should be unique, reflecting quantum confinement and quantization of bulk phonons [93]. Furthermore, since the initial report of bright visible luminescence from "porous silicon" [14], there have been many efforts to explain this phenomenon based on quantum confinement in silicon wires or bars [13, 45, 104, 44, 105].

In this chapter, we take on the question of determining the ground-state structure of nanometer sized bars of silicon as a central issue. Calculations to date, where this has not been the central issue, all have been done with hydrogen-passivated silicon surfaces and place the atoms at their ideal bulk coordinates [104] or simply relax them to the closest energy minimum without exploring alternate constructions [13, 45]. Hydrogen-passivation of silicon surfaces prevents many different types of reconstructions, a subject of interest when silicon surfaces are exposed to vacuum and which we study here. Furthermore, the question of the "rounding" of such bars by the formation of facets along their edges has generally been ignored.

2.1 Background

All of the *ab initio* electronic structure calculations which we report here were carried out within the total energy plane-wave density-functional pseudopotential approach [77], using the Perdew-Zunger [78] parametrization of the Ceperley-Alder [15] exchange-correlation energy and a non-local pseudopotential of the Kleiman-Bylander

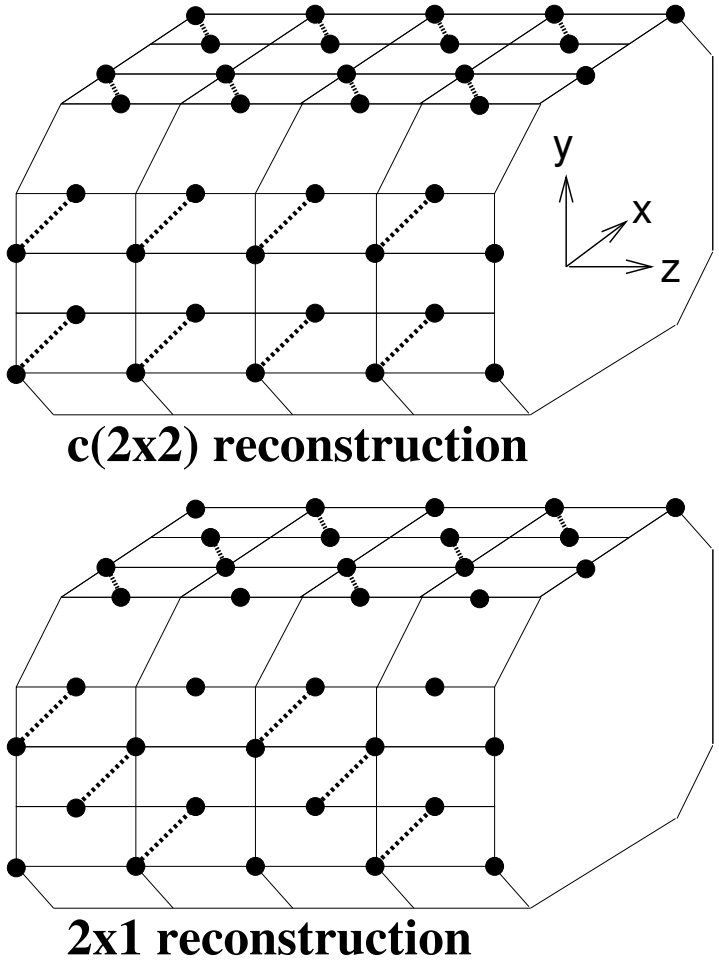

Figure 2-1: A schematic view of the bar and the bonds formed on the surface of the bar in the c(2x2) reconstruction (top) and in the 2x1 reconstruction (bottom).

form [53] with p and d non-local corrections. In all cases, we used a plane wave basis set with a cutoff energy of 12 Ry. Electronic minimizations were carried out using the analytically continued functional minimization approach [4].

To establish baseline information which we shall need later, we computed *ab initio* energies for various reconstructions of the Si(100) surface. These calculations were carried out in a supercell geometry with slabs of twelve atomic layers containing 48 silicon atoms and separated by 9 Å of vacuum. We sampled the Brillouin zone using the four k-points $(0, \pm\frac{1}{4}, \pm\frac{1}{4})$. Among the 2x1, p(2x2), and c(2x2) reconstructions of the (100) surface (Figure 2-2), we find in good agreement with previous calculations

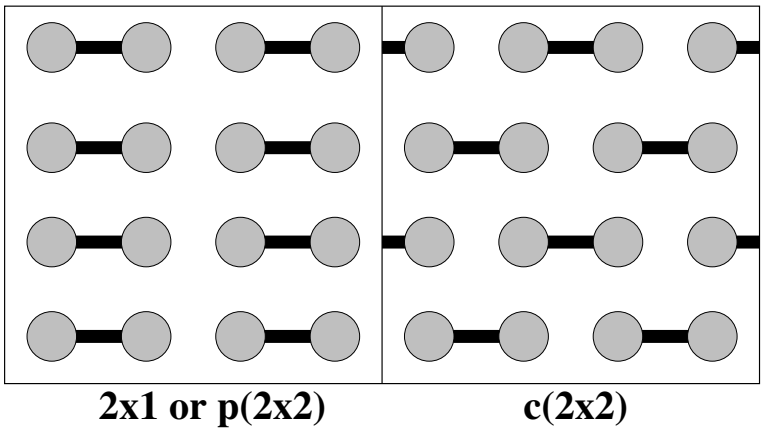

Figure 2-2: These two schematic top views of the Si(100) surface show the dimerization patterns for both the 2x1 or p(2x2) (left) and c(2x2) reconstructions (right).

[83, 31] that the p(2x2) is lowest in energy with a binding energy of 1.757 eV/dimer, and that the 2x1 reconstruction is higher in energy than the p(2x2) by 0.114 eV/dimer. Furthermore, we find the c(2x2) reconstruction to be higher in energy than the p(2x2) by 0.154 eV/dimer. To determine the expected size of the facets along the edges of the bars we require the Si(100) and Si(110) surface energies. These are known experimentally to be 1.36 and 1.43 J m^{-2} respectively [28].

Our *ab initio* study of edges is carried out on bars with a cross section of 2.5×2.5 cubic unit-cells in the (001) plane. We apply periodic boundary conditions in the [001] direction with a periodicity of two cubic unit cells. The ball-and-stick diagram in Figure 2-3 depicts the projection of this structure on to the (001) plane. *Ab initio* calculations on the bars were performed using the same pseudopotential and energy cutoff as used for the surfaces. For the bars, we sampled the Brillouin zone at the two k-points $(0, 0, \pm\frac{1}{2})$ and provided for a minimum of 6 Å of vacuum between periodic images of the bars.

2.2 Prediction of a finite-size transition

Before determining the reconstructions along the edges and surfaces, we first must determine the overall cross-sectional geometry of the bar. To establish this, we per-

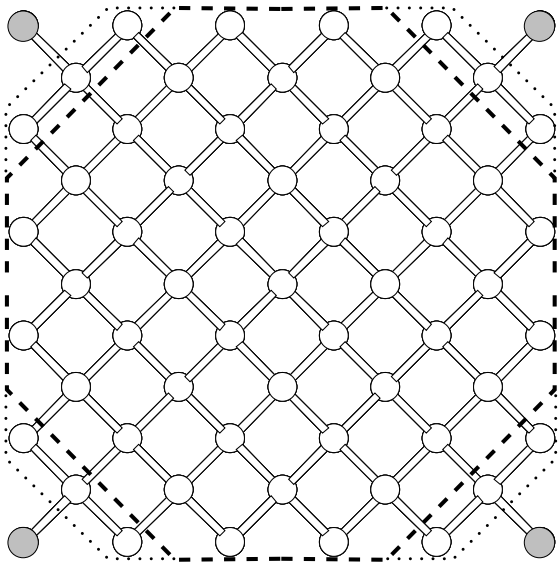

Figure 2-3: Cross sectional view of the silicon bars showing the two different Wulff constructions that result from using either the experimental (dashed lines) or the tight-binding surface energies (dotted lines). The final structure used in this study is formed by removing the shaded atoms along the edges.

formed the Wulff construction using the experimental surface energies given above. Figure 2-3 shows the resulting shape when {110} facets connect {100} surfaces of silicon, scaled to the lateral size of our bars. The atomic-scale structure most consistent with the Wulff construction in shape and aspect ratio appears in Figure 2-3, where we have removed the four columns of shaded atoms along the edges. The projection in the (001) plane of the final structure is an octagon. With the aforementioned periodicity along [001], the final structure contains 114 atoms. Without relaxation, all atoms on the surfaces of the bar are two-fold coordinated.

Having determined the overall geometry, we may now turn to the more subtle issue of relaxation and possible reconstructions along the surfaces and edges. We have found two competing structures: one which best satisfies the system in terms of the total number of bonds, and the other which best satisfies the system in terms of the configuration of the exposed surfaces. We find that the system cannot satisfy

both conditions simultaneously.

Starting from the unrelaxed configuration, there is one unique surface atom with which each edge atom may bond in order to become three-fold coordinated. This bonding does not change the periodicity along the edge, which is the [001] vector of the crystal lattice. This periodicity, however, is incompatible with the periodicity of the p(2x2) low energy state of the {100} facets. To maintain maximal bonding, the arrangement of atoms on the {100} facets must then revert to the higher energy c(2x2) reconstruction. We denote this configuration of the bar as the "c(2x2) reconstruction" (See Figure 2-1).

While the c(2x2) reconstruction maximizes the number of bonds in the system, the increased energy represented by the c(2x2) arrangement on the {100} facets will eventually outweigh the benefit of maximal bonding along the one dimensional edges for a sufficiently large system. When the {100} facets assume the p(2x2) configuration, the periodicity along the [001] direction is doubled. The edge atoms are no longer equivalent, and every other atom now cannot form a new bond and remains only two-fold coordinated. Because of bonding restrictions into the interior of the structure, the dimer rows along alternate faces assumes a pattern which leads us to denote this configuration as the "2x1 reconstruction." (See Figure 2-1.)

The ground state of the bar is thus determined by the balance between bonding along the edges and the surface energy of the facets. We find that for our bar the effects from the edges overcome the natural tendency of the surfaces. The c(2x2) configuration is lower in energy than the 2x1 configuration by 1.94 eV per [001] vector along the length of the bar. There is therefore a size-dependent transition in this system as we increase the lateral dimension of the bar and place relatively more atoms on the {100} facets. Our calculations place the crossover point at approximately 5 dimer pairs on each (100) facet per unit cell along [001], corresponding to a bar with cross sectional dimension of approximately 3.0 nm $\times$ 3.0 nm, far larger than the scale of an atomic cluster. We therefore predict an important edge-driven transition at a scale only two or three smaller than what has been achieved experimentally to date [95].

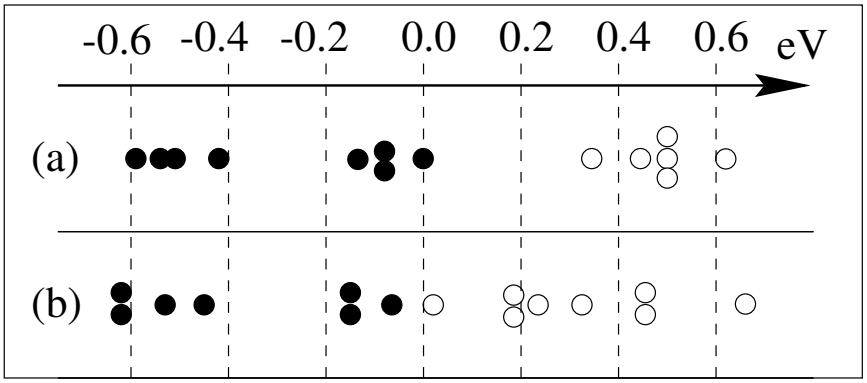

Figure 2-4: Eigen-energies of the electronic states for the two reconstructions of the bars in the vicinity of their Fermi levels at the k-points used in the calculations: (a) is the 2x1 configuration, (b) is the c(2x2) configuration. Filled states are denoted by filled circles and empty states by empty circles. The zero of energy is arbitrary.

More generally, we see that the compatibility of competing surface reconstructions with the translational symmetry of the edges plays a pivotal role in determining the ground state of nanoscale structures. In our specific case, it is the principal physical mechanism giving rise to the size-dependent transition for our bars.

2.3 Implications of the transition

Comparing the electronic structures of the two reconstructions (see Figure 2-4), we find that the 2x1 configuration has a gap of 0.35 eV across the Fermi level. Furthermore, the topmost filled states consist of four nearly degenerate states which are localized on the four edges of the bars, and this cluster is separated from states below it in energy by a gap of 0.30 eV. This nearly symmetric placement with sizeable gaps as well as the spatial localization of the edge states leads us to conclude that the 2x1 bar is insulating.

On the other hand, the electronic structure of the c(2x2) configuration is more subtle. The states in the vicinity of the Fermi level are localized on the surfaces of the bars, and the gap across the Fermi level is only 0.09 eV. We believe the c(2x2) configuration is a small-gap semiconductor or even perhaps metallic. Thus the nature

of the low energy electronic states and excitations differ between the two bars and should manifest themselves in physical measurements such as electrical conductivity or optical spectra.

Next, considering possible differences in mechanical properties, we carried out molecular dynamics simulations using a tight-binding model to compute transverse acoustic phonon frequencies for the bars. We believe that the tight-binding model will give us a qualitative view of the differences which may exist between the two bars. If any large differences are found, we should examine this issue in more detail using the more demanding *ab initio* techniques.

We used the semi-empirical tight-binding model of Sawada [86] with the modification proposed by Kohyama [59]. This model provides a good qualitative description of bulk, dimer, and surface energetics of silicon. In using the Sawada model, we only keep Hamiltonian matrix elements and repulsive terms between atoms closer than $r_{nn} = 6$ Å.

In order to calculate the band-structure energy, we used the fully parallelizable $O(N)$ technique of Goedecker and Colombo [33]. By checking the convergence of the total energy to its ground-state value (determined by exact diagonalization), we found it necessary to use the following set of parameters to ensure convergence of energies to within an accuracy of 10^{-4} eV/atom: in the nomenclature of [33], we have $k_B T = 0.125$ eV, $n_{pl} = 300$, and $r_{loc} = 15.0$ Å.

Table 2.1 summarizes the results of the Sawada model for the various key physical values in this study. Although the Sawada model correctly predicts asymmetric dimers as the ground-state of the Si(100) surface, it is not sensitive to the delicate rocking of dimers that differentiates the 2x1 and p(2x2) reconstructions and hence finds the 2x1 reconstruction to be lower in energy than the p(2x2). However, in our study, the relevant energy difference is that between the c(2x2) and the lowest-energy reconstruction of the surface, and this quantity is reproduced rather well. The Sawada model also does well in predicting surface energies: Figure 2-3 shows the result for the Wulff construction when we use the tight-binding surface energies, and the resulting geometry is very similar to the experimentally derived one. Thus we believe that the

	Expt/*Ab initio*	TB
Bulk properties		
Binding energy (eV/atom)	-4.63[a]	-4.79
Bulk modulus (Mbar)	0.975[a]	0.906
Phonon frequencies		
Γ (THz)	15.5[b]	18.1
$\vec{k} = (\pi/4a)\hat{z}$ LA (THz)	3.9[b]	3.8
$\vec{k} = (\pi/4a)\hat{z}$ TA (THz)	2.4[b]	3.0
Si(100) reconstructions		
lowest energy (eV/dimer)	p(2x2): 1.757	2x1: 2.05
c(2x2) (eV/dimer)	1.603	1.93
Surface energies		
Si(100) (J m^{-2})	1.36[c]	1.44
Si(110) (J m^{-2})	1.43[c]	1.77
Differences between 2x1 and c(2x2) bars		
Energy difference (eV/a)	1.94	1.57
Cross-over (width in nm)	3.0	3.0

Table 2.1: Comparison of the Sawada model (TB) with experimental and *ab initio* results: (a) is reference [52], (b) is reference [24], and (c) is reference [28]. The energy difference between the 2x1 and c(2x2) reconstructions of the silicon bars are given in eV per unit cell along [001], and the cross-over refers to the approximate width of a bar when the two reconstructions have the same energy (see text).

Sawada model provides a good semi-quantitative description for the physics of our bars.

Using the Sawada model, we computed transverse acoustic phonon frequencies for the $\vec{k} = (\pi a/4)\hat{z}$ mode (along Δ) for both reconstructions of our bars, the longest allowed wavelength along the length of the bar consistent with the periodic boundary conditions. We ran (N, V, E) molecular dynamics simulations using the Verlet algorithm with a time-step of 2.4 fs for 900 time steps. Phonon frequencies were identified as peaks in the frequency-domain power-spectrum of the velocity autocorrelation function as estimated by auto-regressive fits. We found frequencies of 1.98 ± 0.02 THz and 1.93 ± 0.02 THz for the c(2x2) and 2x1 configurations respectively. Thus we see that edge effects do not have a significant effect on the long wavelength vibrations of the bars.

2.4 Conclusions

We have performed an *ab initio* study of the energetics of long bars of silicon in vacuum. We found useful the atomic-scale version of the Wulff construction as a first step in determining nanoscale structure. Next, we found that one cannot ignore the interplay between the edges and surfaces in silicon structures with dimensions of a few nanometers, where the compatibility of the surface reconstructions with the symmetry of the edges plays an important role. In particular, the ground-state of our bars changes from one surface reconstruction to another, signalling a cross section-dependent phase transition, with significant influence on the electronic structure of the system. Finally, we find that even on the scale of a few nanometers, the surface-edge interplay has little effect on mechanical properties.

Chapter 3

A Priori Calculation of Energies of Solvation

The study of reactions in solution and the calculation of solvation energies are problems of importance and interest in physics, chemistry, and biology. For a polar solvent such as water, the effects of solvation on the energetics of reactions can be crucial. However, modeling solvated reactions is a challenging problem as one needs both a reliable quantum mechanical treatment of the reactants in order to correctly describe bond rearrangements as well as thermodynamic integration over the solvent degrees of freedom.

An idealized model would describe both the reactants and the large number of solvent molecules quantum mechanically using an *ab initio* approach. However, simulating such a large number of particles quantum mechanically poses a prohibitive computational burden even on todays' largest computers, and we require some sort of simplification or approximation scheme. Clearly, the electronic states of the reacting molecules must be treated quantum mechanically, so that one must concentrate on simplifying the description of the solvent. The most common and direct approach replaces the solvent by a dielectric continuum that surrounds a solvent-excluded cavity about the reactants, and we refer the reader to the excellent review of [92].

Upon examining the state of the art, one sees that current methodologies are highly variegated and that they rely on semiempirical fits or rule-of-thumb prescrip-

tions for describing the dielectric cavity, the charge density of the reactants, or the polarization of the dielectric. We instead consider systematically the impact of each stage of approximation in coarse-graining from an *ab initio* calculation one could perform in principle to the dielectric treatment used in practice.

In addition to this new perspective, we provide a novel approach for dealing with the important issue of constructing appropriate dielectric cavities. The current literature uses van der Waals spheres to construct the cavity. The results, however, can be sensitive to the radii of the spheres requiring empirical adjustment [92, 68]. Rather than this *a posteriori* approach, we construct the cavity based on the *a priori* consideration that the dielectric, by definition, reproduces the thermodynamic response of the molecular solvent to electrostatic perturbations. Below, we effect this by choosing the dielectric cavity boundary as an isosurface of the reactant electron density which reproduces the correct solvent response.

Once we specify the dielectric cavity, we still face the problem of solving the electrostatic equation. For this, we introduce a powerful preconditioned conjugate-gradients method exploiting Fourier transform techniques to solve the electrostatic problem in the presence of the dielectric cavity.

To explore the efficacy of our approach, we consider a reaction that is highly sensitive to the dielectric response of the solvent, the hydrolysis of methylene chloride (CH_2Cl_2). There is both experimental (e.g. [29, 85, 64, 91, 82]) and theoretical interest (e.g. [63]) in the aqueous breakdown of this industrial toxin (methylene chloride). Previous theoretical studies of this reaction showed promising results, but were based on the simple approximations of a dipole in a spherical cavity. In addition to providing a more systematic analysis of the hydrolysis of methylene chloride, we study in a controlled manner the impact of these common approximations on the resulting solvation energies.

We begin in Section 3.1 with a discussion of the hydrolysis reaction under consideration. There we present the overall hydrolysis reaction for methylene chloride and identify the rate limiting step, the choice of reaction geometry and reaction coordinate, and the *ab initio* density-functional method used to treat the reactants. Next,

Section 3.2 describes our theoretical approach beginning with how one in principle calculates free energies *ab initio* and then presents a detailed analysis of the chain of approximations that lead to the dielectric model. We then present our method for creating the dielectric cavity followed by our new algorithm for solving the electrostatic problem. Finally, in Section 3.3 we present results for the hydrolysis of methylene chloride followed by a detailed analysis of the impact of the dipole and sphere approximations and the importance of self-consistency in solving the Poisson equation.

3.1 Pathway for hydrolysis of methylene chloride

Methylene chloride (CH_2Cl_2) is an important industrial solvent. Recently, there has been interest in treating aqueous wastes with super critical water oxidation (SCWO) methods (oxidation in water in conditions above its critical point at 374^o C and 221 bar [63]). However, it is found that significant destruction of CH_2Cl_2 occurs in subcritical conditions through hydrolysis rather than oxidation. Perhaps more surprisingly, the hydrolysis reaction rate is found to decrease dramatically as temperature increases through the critical point [63, 85], signalling the importance of solvation effects for this reaction.

The overall reaction describing the hydrolysis of methylene chloride is

$$CH_2Cl_2 + H_2O \rightarrow HCHO + 2HCl\,.$$

This reaction is known to be a two step process [29]. The first step is a slow, rate-limiting substitution process that dictates the overall rate of the above reaction,

$$CH_2Cl_2 + H_2O \rightarrow CH_2ClOH + HCl\,. \tag{3.1}$$

The species CH_2ClOH is unstable and undergoes a fast internal rearrangement that

expels H^+ and Cl^- to form the final HCHO,

$$CH_2ClOH \rightarrow HCHO + HCl\,.$$

Therefore, we need only concentrate our attention on the rate-limiting step of Eq. (3.1).

The first step of the reaction of Eq. (3.1) requires an H_2O molecule to approach the CH_2Cl_2 molecule very closely and for a Cl^- ion to leave. This proceeds via the creation of a transition state,

$$CH_2Cl_2 + H_2O \rightarrow CH_2ClOH_2 + Cl^- \tag{3.2}$$

Below, we concentrate on understanding the energetics of the transition-state complex of Eq. (3.2), which should provide us with the energy barrier of the reaction of Eq. (3.1).

The chemistry of the reaction of Eq. (3.2) is known to be of the S_N2 variety, where the oxygen approaches the carbon from the side opposite to the chlorine ion that leaves the CH_2Cl_2 molecule. However, the precise orientation of the water molecule during this reaction is not known. Figure 3-1 shows that the hydrogens in the water molecule may be oriented in one of two ways, either (a) in the same plane as the carbon and chlorine atoms, or (b) rotated by 90^o. Resolution of this question requires *ab initio* calculations.

First principles calculations provide us unambiguous, *a priori* results for energies and forces. We perform *ab initio* calculations to determine reactant geometries, energies, and charge densities in vacuum. We carry out these calculations within the pseudopotential plane-wave density-functional approach in the local-density approximation [77] using the Perdew-Zunger parameterization [78] of the Ceperly-Alder exchange-correlation energy [15]. Non-local pseudopotentials of the Kleinmann-Bylander form [53] constructed using the optimization scheme of Rappe *et al.* [80] describe the interaction of valence electrons with the ionic cores. The pseudopotential for carbon has a non-local projector for the s channel, and the oxygen and chlorine have projectors for the p channel. The plane-wave cutoff is 40 Rydbergs for a total of

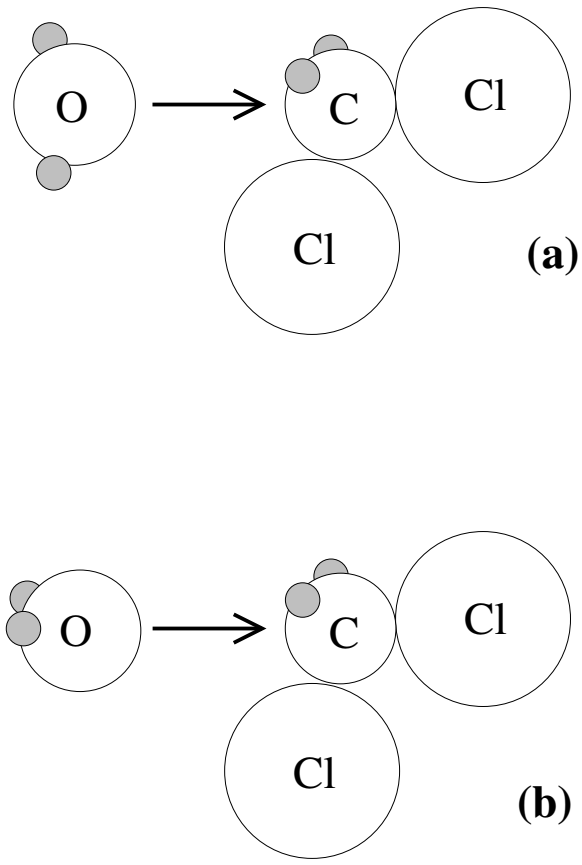

Figure 3-1: Schematic diagrams of the stereochemistry of CH_2Cl_2 hydrolysis. Shaded spheres represent hydrogen atoms and other atoms are labeled. The arrow shows the direction of approach of the H_2O molecule. Possibilities (a) and (b) are discussed in the text.

35,000 coefficients for each electronic wave function. For a given choice of ionic positions, electronic minimizations are carried out using a parallel implementation of the conjugate-gradient technique of reference [77]. The supercell has dimensions of 15 Å × 9 Å × 9 Å, which separates periodic images of the reactants sufficiently to minimize suprious interaction effects, even for the elongated transition states. We fix the carbon atom at the origin of our simulation cell and place the reaction coordinate λ, defined as the oxygen-carbon distance of the reactants in Eq. (3.2), along the long, 15 Å x-axis of our cell. We determine optimized molecular structures by moving the ionic cores along the Hellman-Feynman forces until all ionic forces (except along the *fixed* reaction coordinate λ) are less than 0.1 eV/Å in magnitude. To illustrate the

Bond Lengths (Å)		
bond	*ab initio*	experimental [61]
C-H	1.11	1.09
C-Cl	1.79	1.77
O-H	0.99	0.96
Dipole Moments (Debye)		
molecule	*ab initio*	experimental [67]
H_2O	1.89	1.85
CH_2Cl_2	1.79	1.6

Table 3.1: *Ab initio* bond lengths and dipole moments

accuracy of these calculations, Table 3.1 compares experimental and *ab initio* values for bond-lengths and permanent dipole moments of the isolated molecules.

The *ab initio* calculations establish that pathway (b) of Figure 3-1 is preferred, being 0.21 eV lower in energy for a typical value of the reaction coordinate λ. We thus consider only this pathway in the remainder of our work. Towards this end, we catalogue optimized geometries for a series of values of reaction coordinate along this pathway.

3.2 Theoretical methodology

Having determined the energies and configurations along the pathway in vacuum, we now turn to the much more challenging problem of the calculation of Gibbs free energies. Of particular interest is ΔG^*, the difference between the Gibbs free energy of the solvated transition state $G^\ddagger$ and the Gibbs free energy of the solvated reactants at infinite separation G^j,

$$\Delta G^* = G^\ddagger - \sum_j G^j \,. \tag{3.3}$$

Next we define the free energy of solvation G^i_{solv} as the difference in Gibbs free energy of a configuration i in vacuum G^i_0 (which we have calculated above *ab initio*) and in solution G^i,

$$G^i = G^i_0 + G^i_{solv} \,. \tag{3.4}$$

The task is to compute G^i_{solv} which describes the interaction of the solvent with the reactants.

3.2.1 Microscopic treatment of Gibbs free energies

Direct calculation of the Gibbs free energy G^i requires the evaluation of a large phase space integral,

$$e^{-G^i/k_BT} = \int' dq_r \int dq_s \; e^{-[H_r(q_r)+H_{r,s}(q_r,q_s)+H_s(q_s)]/k_BT}, \tag{3.5}$$

where q_r and q_s are coordinates describing the positions of the reactant and solvent nuclei, respectively, and H_r, H_s, and $H_{r,s}$ are the Hamiltonians describing the isolated reactants, isolated solvent, and their interaction, respectively. We work within the Born-Oppenheimer approximation, and therefore these Hamiltonians are the corresponding system energies for fixed nuclear coordinates (q_r, q_s). Finally, the prime on the outer q_r integral indicates that we only sum over reactant coordinates that are compatible with the configuration i.

In principle, there is no fundamental difficulty in (a) preparing a cell containing the reactants and a large collection of solvent molecules, (b) computing the system energy $H_r + H_s + H_{r,s}$ in Eq. (3.5) within density-functional theory, and (c) integrating over the phase space with appropriate molecular dynamics or Monte Carlo methods. The only hindrance is the prohibitive computational effort required. Accurate description of the bonding rearrangements of the chemically active reactants requires quantum mechanical calculation of the reactant Hamiltonian H_r. Therefore, the only option to render the computation tractable while maintaining accuracy is somehow to coarse-grain the detailed microscopic description of the solvent.

3.2.2 Coarse-graining the solvent

Our approach to coarse-graining the solvent replaces the detailed molecular arrangement of the solvent molecules by the electrostatic field $\Phi(r)$ which they generate. To accomplish this, we separate the interaction of the reactant charge distribution

$\rho_r(r)$ and the solvent electrostatic field Φ from all other terms in the reactant-solvent interaction Hamiltonian,

$$H_{r,s}(q_r, q_s) = \int d^3r\, \rho_r(r)\Phi(r) \;+\; V(q_r, q_s)\,. \tag{3.6}$$

The term V includes all remaining reactant-solvent interactions such as hard-core repulsions and van der Waals forces.

Next, we define the free energy $G_s[\Phi]$ of the solvent *given* that it produces a field $\Phi(r)$ through

$$e^{-G_s[\Phi]/k_BT} = \int_{q_s\to\Phi} dq_s\; e^{-[H_s(q_s)+V(q_r,q_s)]/k_BT}, \tag{3.7}$$

where $q_s \to \Phi$ means that we only integrate over configurations q_s that give rise to the field Φ. The only dependence of the free energy G_s on the configuration of the reactants q_r is through the interaction in V, the most important being the hard-core repulsions that create the solvent-excluded cavity about the reactants. Therefore, $G_s[\Phi]$ is the coarse-grained description of the free energy of the solvent in the presence of the cavity.

Next, we expand G_s about its minimum at Φ_c,

$$G_s[\Phi] = G_s[\Phi_c] + \frac{1}{8\pi}\int d^3r\; (\Phi(r) - \Phi_c(r))\,\hat{K}\,(\Phi(r) - \Phi_c(r)) + ... \tag{3.8}$$

Here, $\hat{K}$ is a positive-definite symmetric kernel which depends on the cavity shape and which specifies the thermodynamic response of the solvent in the presence of the cavity, and Φ_c is the electrostatic field created by the solvent in the presence of an empty cavity. As we shall discover below, retaining only the quadratic expansion of G_s ultimately leads to a familiar dielectric description.

The free energy G^i of Eq. (3.5) now can be rewritten in terms of G_s by integrating over all possible solvent fields Φ,

$$e^{-G^i/k_BT} = \int' dq_r\, e^{-H_r(q_r)/k_BT} \int d\Phi\; e^{-\left[\int d^3r\, \rho_r(r)\Phi(r)\; +\; G_s[\Phi]\right]/k_BT}, \tag{3.9}$$

which within the quadratic approximation of Eq. (3.8) becomes

$$\begin{aligned} e^{-G^i/k_BT} &= \int' dq_r \int d\Phi \; e^{-\left[H_r(q_r)+\int d^3r\,\rho_r\Phi+G_s[\Phi_c]+\frac{1}{8\pi}\int d^3r\,(\Phi-\Phi_c)\hat{K}(\Phi-\Phi_c)\right]/k_BT} \\ &= \int' dq_r \; e^{-\left[H_r(q_r)+\frac{1}{2}\int d^3r\,\rho_r(\phi_s+\Phi_c)+G_c\right]/k_BT}. \end{aligned} \tag{3.10}$$

In going from the first to the second line, we have performed the Gaussian integral over the solvent field Φ, which introduces two new quantities. The variable $\phi_s(r)$ in Eq. (3.10) is the electrostatic potential at the maximum of the Gaussian integrand as determined by the condition,

$$\hat{K}\left[\phi_s(r) - \Phi_c(r)\right] = -4\pi\rho_r(r)\,, \tag{3.11}$$

from which it is clear that the linear operator $\hat{K}$ relates the electrostatic response of the solvent $\phi_s(r)$ to the presence of the reactant charges $\rho_r(r)$. The constant G_c is the free energy of formation of the empty cavity. Physically, the contents of the square brackets of the exponent in Eq. (3.10) represent the total free energy of the system for a fixed reactant configuration q_r. This free energy consists of the internal energy of the reactants $H_r(q_r)$, the electrostatic interaction of the reactants with response of the solvent $\int d^3r\,\rho_r\,\phi_s$, the interaction of the solvent with the cavity potential $\int d^3r\,\rho_r\,\Phi_c$, and the cavitation free energy G_c.

For the case of present interest, the solvation of molecules with permanent electrical moments, we would expect the induced solvent potential ϕ_s to greatly exceed the cavitation potential Φ_c, which arises from an electrostatically neutral cavity, and therefore that the electrostatic reactant-solvent interactions will be the dominant contribution to the free energy. Indeed, studies of polar molecules show the total solvation free energy to be strongly correlated with the aforementioned electrostatic interaction. Although these two quantities have absolute offset of about 0.2 eV, free energy differences between configurations may be computed to within 0.05 eV from differences in the electrostatic interaction alone [92]. Mathematically, this implies that we can set $\Phi_c = 0$ and that G_c may be taken to be independent of the reactant

configuration q_r, resulting in

$$e^{-G^i/k_BT} = e^{-G_c/k_BT} \int' dq_r \; e^{-\left[H_r(q_r)+\frac{1}{2}\int d^3r\,\rho_r(r)\phi_s(r)\right]/k_BT}, \tag{3.12}$$

$$\hat{K}\phi_s(r) = -4\pi\rho_r(r)\,. \tag{3.13}$$

The free energy of solvation for configuration q_r is the electrostatic integral

$$G_{solv} = \frac{1}{2}\int d^3r \; \rho_r(r)\,\phi_s(r)\,. \tag{3.14}$$

Eq. (3.13) shows that the reactant charge ρ_r induces a linear response in the solvent which gives rise to the solvent potential ϕ_s. This relation, therefore, also gives the total electrostatic potential $\phi = \phi_s + \phi_r$ as a linear response to the charge of the reactants,

$$\left[\hat{K}^{-1} + \nabla^{-2}\right]^{-1}\phi = -4\pi\rho_r(r).$$

This latter connection corresponds precisely to the standard macroscopic Maxwell's equation,

$$(\nabla\cdot\epsilon\nabla)\phi = -4\pi\rho_r(r)\,, \tag{3.15}$$

and serves to *define* the precise form of ϵ in terms of $\hat{K}$, which we have already defined microscopically.

3.2.3 Comparison of continuum and molecular response

Having arrived at a coarse-grained description of the solvent in terms of its dielectric function in Eq. (3.15), we now face the problem of specifying ϵ. In general, the true microscopic dielectric is a non-local function which relates to the cavity in a highly complicated manner. However, as we now show, quite simple, *computationally tractable* models can describe very well the underlying physics.

Within the solvent-excluded cavity surrounding the reactant, there is no solvent dielectric response and therefore we expect $\epsilon = 1$. Far from the cavity, we expect the dielectric response to be that of the bulk solvent $\epsilon = \epsilon_b(P,T)$. Here, we include the

dependence of the bulk solvent's dielectric constant ϵ_b on pressure P and temperature T as we wish to investigate dielectric effects near the critical point of water where ϵ is a strong function of these parameters. Finally, we can expect the dielectric response somehow to interpolate smoothly between these extremes.

To investigate the suitability of such a simple dielectric description of an ordered molecular solvent, we in principle could work *ab initio*. We could place a large number of water molecules in a simulation cell, perform molecular dynamics or Monte Carlo sampling, and compare the response to electrostatic perturbation of this system with the response of a dielectric model. This, however, would be both computationally expensive and largely unnecessary as the solvent molecules remain chemically inert and interact with the reactants primarily via electrostatic and repulsive forces. Therefore, to investigate the impact of coarse-graining the molecular details of the solvent, we employ a simpler model where the solvent's electronic degrees of freedom are not described explicitly. We use the microscopic SPC model for water which has electrostatic point-charges for the three atoms in the H_2O molecule and Lennard-Jones interactions between the oxygen atoms to incorporate short-range repulsive and long-range van der Waals interactions [9].

We extract the linear response of H_2O to electrostatic perturbations by simulating the behavior of SPC water molecules about a spherical cavity. We represent the cavity as a Lennard-Jones potential with a diameter of 4.25 Å and with a well-depth of 0.0030 eV centered at the origin of the simulation cell. (This corresponds to $\sigma = 2.125$ Å and $\varepsilon = 0.0030$ eV in the notation of [9].) This sphere has roughly the same size and radius of curvature as the reactant complex that we study below, and the depth parameter is chosen small so that the potential primarily presents a repulsive core to oncoming water molecules. At the center of the cavity, we place a point charge of $Q = 0, \pm 0.1e, \pm 0.2e$ so that $\rho_r(r) = Q\delta^3(r)$ in Eq. (3.15). For each value of Q, we run constant (N, P, T) molecular dynamics simulations [1, 72] with N=256 H_2O molecules, P=1 atm, and $T = 298$ K, which corresponds to an average cell volume of 7,800 Å^3. We equilibrate the cell for 25 ps before gathering statistics over a run-time of 250 ps. The time-averaged oxygen-cavity radial distribution functions show a large

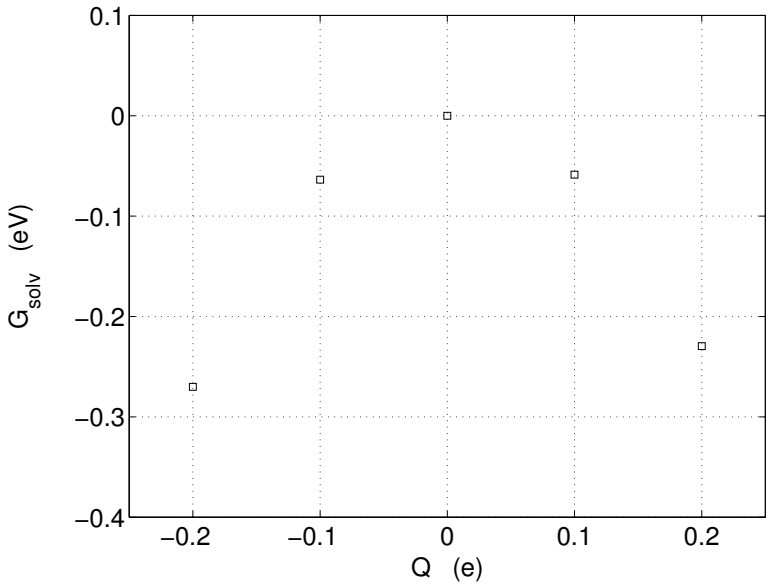

Figure 3-2: Electrostatic solvation free energy for the spherical cavity SPC simulations as a function of Q, the strength of the point-charge at the center of the cavity.

peak at a radius of 3.0±0.1 Å and no oxygen presence for smaller radii. This gives an effective "hard core" radius of 3 Å for this spherical cavity.

Binning and averaging the instantaneous charge distributions over the 250 ps sampling period gives the average induced charge density in the solvent ρ_s, from which we may calculate ϕ_s using $\nabla^2\phi_s = -4\pi\rho_s$. We then compute the solvation free energy of Eq. (3.14). Figure 3-2 shows quadratic behavior in Q, the first indication that the dielectric approach is appropriate.

As discussed above, we now compare this response with that of a dielectric function which interpolates smoothly from the interior of the cavity to deep within the solvent,

$$\epsilon(r) = 1 + \frac{(\epsilon_b(P,T) - 1)}{2} \operatorname{erfc}\left(\frac{r_\epsilon - r}{\sqrt{2}\sigma}\right) . \tag{3.16}$$

Here, r_ϵ is the radius where the dielectric changes and σ measures the distance over which the change occurs.

If a continuum dielectric description is viable, then an appropriate choice of r_ϵ and σ will ensure that, in general, the response of the model dielectric matches that

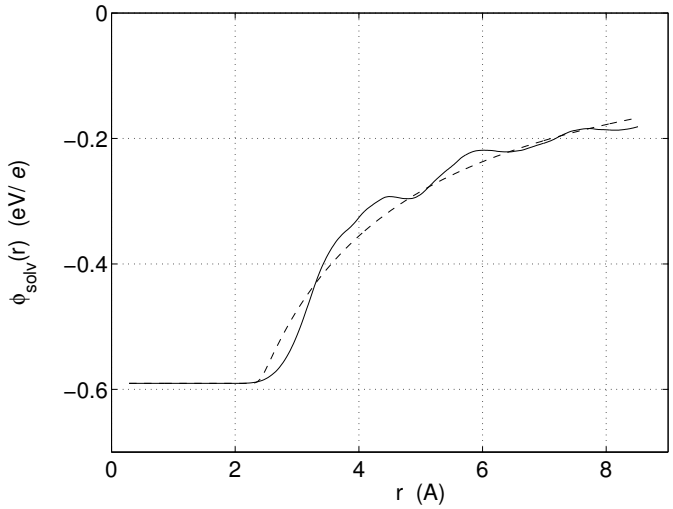

Figure 3-3: Electrostatic potentials created by the solvent ϕ_s for the spherical cavity versus distance r from the cavity center. The solid line is calculated from the time-averaged solvent charge density of the SPC molecular dynamics simulations. The dashed line is the potential from the dielectric model. Here, a charge $Q = +0.1e$ was placed at $r = 0$ and the cavity has Lennard-Jones diameter of 4.25 Å. The dielectric model has $r_\epsilon = 2.65$ Å, $\sigma = 0.11$ Å, and $\epsilon_b = 80$.

of the molecular solvent and, in particular, that the electrostatic free energies match as closely as possible. Figure 3-3 shows ϕ_s for $Q = +0.1e$ as calculated from the molecular simulations and the dielectric of Eq. (3.16) with $r_\epsilon = 2.65$ Å, $\sigma = 0.11$ Å, and $\epsilon_b = 80$ as appropriate for water at ambient conditions. For this demonstration, $\sigma = 0.11$ Å is fixed and r_ϵ was chosen to minimize the mean square difference between the molecular and dielectric potentials ϕ_s. Although some shell structure is evident in the molecular calculation, the two potentials track one another quite closely. Moreover, because we are dealing with a point charge, the solvation free energy of Eq. (3.14) may be read off as simply the value of ϕ_s at the origin, where the two calculations agree quite well.

3.2.4 Specification of dielectric cavity

As we have just seen, the dielectric approximation to the response of the solvent appears quite successful in practice. Encouraged by this, we now specify how we

construct the dielectric cavity for our reactant system. Specifically, we must choose a closed surface surrounding the reactants that specifies the boundary where the dielectric changes from its value in vacuum $\epsilon = 1$ to its value in the bulk solvent $\epsilon = \epsilon_b(P, T)$.

The current state of the art for choosing the cavity begins by placing spheres with empirically adjusted van der Walls radii on atomic or bond sites. These adjusted radii are scaled from the experimentally fit radii by factors in the range 1.15-1.20 [92]. Next, either the volume enclosed by the intersecting spheres is taken as the dielectric cavity, or a further spherical probe is rolled over this volume and either the surface of contact of the probe or the surface traced by its center is used as the cavity boundary [92]. The latter two choices tend to either underestimate the solvation energy or to produce unphysical inward-bulging regions that lead to computational difficulties and unphysical sensitivities to slight changes in reactant geometry [92]. When this approach is stable, calculated solvation energies can depend strongly on the size and placement of the spheres [68]. Aside from the empirical nature of the approach, it is not obvious why spherical shapes should be used in molecules where electron densities can differ significantly from a sum of spherical atomic densities, or why it is appropriate to use atomic van der Waals radii in a molecule.

Our *a priori* approach is based on the principle that the dielectric cavity models the thermodynamic response of the solvent to the charge density of the reactants. Our strategy for applying this idea has two parts. First, we know that strong repulsive forces between reactant and solvent molecules create the cavity, and that these repulsive forces are active when the reactant and solvent electron clouds overlap thereby causing the system's energy to rise rapidly due to the Pauli exclusion principle. Therefore, the shape of the surface of closest approach for the solvents and hence the shape of the dielectric cavity should be well approximated by isosurfaces of the electron density of the reactants. Second, to choose the precise value of the electron density specifying the isosurface, we demand that the dielectric so chosen lead to the correct solvation free energy as predicted by an *ab initio* molecular description. In practice, this requirement is very difficult to enforce for an arbitrarily

shaped cavity. Therefore, as a necessary practical compromise, we instead ensure that (a) our dielectric model produces the correct molecular response of the solvent for a computationally manageable model system, and (b) we use relevant *ab initio* calculations to calibrate the results of the model calculations when applying them to the real system. We now provide the details below.

Molecular-dielectric connection

Our first step is to connect the molecular description to the dielectric one so as to extract key physical parameters for use in our *ab initio* modeling below. To this end, we concentrate on the results of our SPC-cavity simulations described above.

The first important parameter is the radius r_O, defined as the position of the first maximum in the cavity-oxygen radial distribution function and therefore the closest-approach distance of the oxygen atoms to the cavity center. As stated above, we have $r_O = 3.0 \pm 0.1$ Å.

The second important parameter is r_{peak}, defined to be the position of the induced charge peak within the dielectric model. For a radial dielectric function such as that of Eq. (3.16), the induced charge density in response to a point-charge Q at the origin is given by

$$\rho_{ind}(r) = \frac{Q}{4\pi r^2}\frac{d}{dr}\frac{1}{\epsilon(r)} = -\frac{Q}{4\pi r^2}\frac{\epsilon'(r)}{\epsilon(r)^2}\,. \tag{3.17}$$

Thus r_{peak} is the radius r where ρ_{ind} has its largest magnitude. For each value of σ, we choose the optimal dielectric model by finding the r_ϵ that minimizes the mean square difference between the ϕ_s generated by the SPC calculation and that of our dielectric model. Two important results emerge from this fitting. First, for all values of Q and σ, r_{peak} is found to be essentially constant $r_{peak} = 2.3 \pm 0.2$ Å, which means that the spatial position of the induced charges is fixed (whereas their magnitude can vary). Second, our fitting procedure provides us with the relation $r_\epsilon = r_{peak} + 2.3\sigma$, where the σ dependence is correct to within $\pm$ 0.03 Å.

This analysis leads to the following physical picture: the oxygen atoms are fixed at $r_O = 3.0$ Å while the induced charge density is centered at $r_{peak} = 2.3$ Å. Furthermore,

their separation $r_O - r_{peak}$, here 0.7 Å, does not change as a function of Q. While we expect that the closest-approach distance r_O to depend on the details of the repulsive potential thus not to be transferable, the difference $r_O - r_{peak}$ should be much more transferable. This is because this separation measures how far inwards from the oxygen positions the statistically-averaged charge density of the solvent extends, and this should depend on the molecular details of H_2O alone, for which the SPC model is sufficient, and not strongly on the form of the repulsive interaction.

The final step in making the molecular-dielectric connection involves finding correct parameters for non-ambient conditions when $\epsilon_b \neq 80$. In principle, we can perform molecular simulations for a set of (P, T) values which correspond to different values of $\epsilon_b(P, T)$, and we can then repeat the above fitting procedure and thus generate a table of r_O and r_{peak} values as a function of ϵ_b. We, however, choose to keep $r_O - r_{peak}$ fixed independent of the value of the bulk dielectric ϵ_b. We believe this to be a good procedure because (1) at ambient conditions, our fits explicitly show this separation to be fixed, and (2) as we have argued above, we believe this difference to be much more transferable to different physical conditions that either parameter alone. Operationally, we find the position of the peak of ρ_{ind} of Eq. (3.17) as a function of σ, ϵ_b, and r_ϵ, and the relation $r_\epsilon = r_{peak} + f(\epsilon_b)\sigma$ holds where $f(80) = 2.3$, in agreement with our result above.

Construction of the dielectric cavity

We can now apply all of the above findings to construct the dielectric cavity. The first problem we face is to find the distance of closest approach of the solvent water molecules to our reacting complex.

Starting with an isolated CH_2Cl_2 molecule, we let an H_2O molecule approach the carbon atom in a direction opposite to a chlorine atom. Furthermore, we orient the H_2O molecule so that the hydrogens are pointing away from the carbon, thereby allowing for the closest approach of oxygen and carbon atoms. We calculate the *ab initio* energy of the system as a function of the carbon-oxygen separation and find a rapid rise over and above k_BT for a separation of 2.5 Å. This provides us with an

ab initio closest-approach radius of $r_O = 2.5$ Å. Therefore, we calibrate our previous results while keeping $r_O - r_{peak}$ fixed at 0.7 Å, so that $r_{peak} = 1.8$ Å. The dielectric radius is given by $r_\epsilon = r_{peak} + f(\epsilon_b)\sigma$.

To convert r_ϵ to an electron density, we start with an isolated CH_2Cl_2 molecule and scan its electron density along the same direction of approach of the H_2O above. We find the electron density value n_ϵ at a distance r_ϵ from the carbon. We then use the isosurface of the electron density at value n_ϵ to define the boundary of the dielectric cavity in the calculations.

We would like to emphasize that our dielectric cavity has a smooth surface and has a very physical shape based on the charge density. We do *not* choose the size and shape of the cavity from tabulated databases of van der Waals radii or excluded volumes that may not be applicable for the energetics of the molecules at hand. Rather, we base our choice on considerations which attempt to reproduce relevant *ab initio* results combined with thermodynamic solvent-reactant free energy of interaction from model calculations as closely as possible.

3.2.5 Solving the Poisson equation

In the remainder of this section, we will describe how we solve the Poisson equation (3.15) for the total potential $\phi(r)$. When solving for ϕ, we face two computational issues: (a) how we choose to represent continuous fields ϕ, ρ_r, and ϵ, and (b) to what level of accuracy we solve Eq. (3.15).

(a) Concerning representation, we use a periodic Fourier expansion to represent all functions (ϕ, ρ_r, ϵ, ϕ_s, etc.). The Fourier grid is $200 \times 120 \times 120$ for our 15 Å $\times$ 9 Å $\times$ 9 Å cell resulting in a grid spacing of 0.075 Å. Since we represent ρ_r and ϵ on a grid, we must address the issue of the discreteness of the grid. The total reactant charge density ρ_r is the sum of a smooth electronic charge density and the point-like charge densities of the ionic nuclei, the latter presenting the essential difficulty when representing ρ_r on the grid. The dielectric function ϵ likewise has rapid variations across the boundary of the cavity. We use Gaussian smoothing to deal with both problems: we replace each ionic point-charge by a Gaussian distribution, and

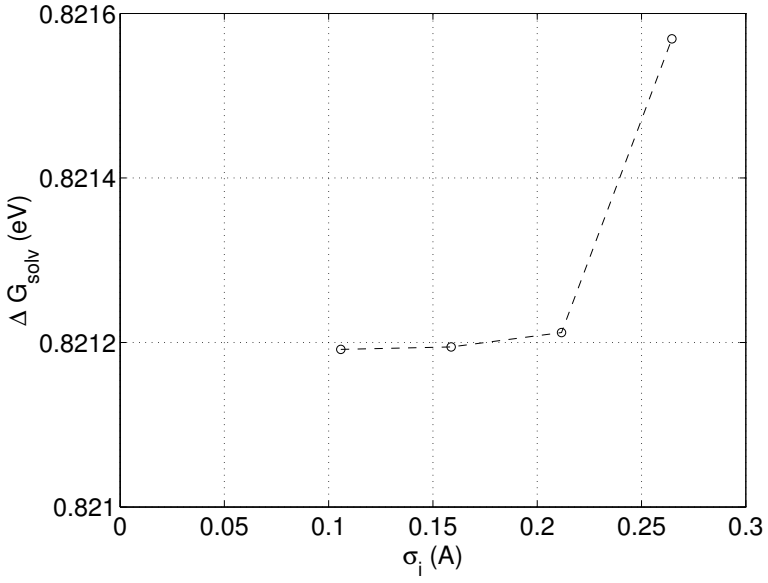

Figure 3-4: The difference of electrostatic solvation free energy between configuations $\lambda = 1.41$ Å and $\lambda = 1.84$ Å versus the ionic smoothing length σ_i. The dielectric smoothing is held fixed at $\sigma_\epsilon = 0.11$ Å.

we smooth the dielectric by convolving it with the same Gaussian. The standard deviation σ of the Gaussian smoothing must be (i) small enough to faithfully recover the same energy differences that we would obtain with true true point charges, and (ii) larger than the grid spacing so that we can faithfully represent ρ_r and ϵ.

Our high-resolution grid allows us to satisfy both constraints. Figure 3-4 shows the behavior of the electrostatic solvation free energy difference between two reactant configurations for a fixed dielectric as a function of σ. Choosing $\sigma \approx 0.1$ Å introduces errors in free energy differences of less than 10^{-4} eV. Therefore, we fix $\sigma = 0.11$ Å for all the calculations reported below. We point out that since any function can be expanded in a Fourier basis, our choice of representation is quite general and free of any *a priori* bias. By simply increasing the size and density of the Fourier grid, we are guaranteed to ensure complete convergence of our results.

(b) To solve the Poisson equation (3.15) accurately, we minimize an auxiliary

quadratic functional $\mathcal{L}[\phi]$:

$$\mathcal{L}[\phi] = \frac{1}{8\pi} \int d^3r\, \epsilon(r) \left|\vec{\nabla}\phi(r)\right|^2 - \int d^3r\, \rho_r(r)\phi(r). \tag{3.18}$$

By construction, the ϕ that minimizes $\mathcal{L}[\phi]$ also satisfies Eq. (3.15). We now provide the details of the minimization process. We represent $\phi(r)$ in our Fourier basis via

$$\phi(r) = \sum_q \hat{\phi}(q)\, e^{iq\cdot r}, \tag{3.19}$$

where q ranges over the Fourier wave-vectors, and $\hat{\phi}(q)$ are Fourier expansion coefficients. Calculation of $\vec{\nabla}\phi$ is done in q-space where it corresponds to multiplying $\hat{\phi}(q)$ by iq. Multiplication by the dielectric function $\epsilon(r)$ and reactant charge density $\rho_r(r)$ are performed in r-space where we multiply by the value of ϵ or ρ_r at each grid point. Integration is replaced by a weighted sum over grid points in r-space, and we use Fast Fourier transforms to effect the change of representation from q to r space and vice versa. In this way, $\mathcal{L}$ becomes a quadratic function of the Fourier coefficients $\hat{\phi}(q)$, and minimization of a quadratic function is an ideal case for the use of conjugate gradient methods. We accelerate convergence by preconditioning each component of the gradient $\frac{\partial \mathcal{L}}{\partial \hat{\phi}(q)}$ by the multiplicative factor $\frac{4\pi}{q^2}$ for $q \neq 0$. With this choice, the minimization requires only twenty to twenty five iterations to reach machine precision. Once we have found ϕ, we easily obtain $\phi_s = \phi - \phi_r$ and calculate the solvation free energy of Eq. (3.14).

3.2.6 The rigid solute approximation

Up to this point, we have not dealt with the reactant coordinates q_r. Even within a dielectric model, performing the integral of Eq. (3.12) poses a difficult problem. We approximate this integral over q_r by evaluation at its maximum. Denoting $\tilde{q}_r$ as the value of q_r that maximizes the exponent of Eq. (3.12), we arrive at the following expression for G^i:

$$G^i = H_r(\tilde{q}_r) + \frac{1}{2} \int d^3r\, \tilde{\rho}_r(r)\tilde{\phi}_s(r)\,, \tag{3.20}$$

where tildes emphasize that both ρ_r and ϕ_s depend on the reactant coordinates $\tilde{q}_r$. In general, when the solvent polarizes, it in turn creates an electrostatic field that acts upon the reactants and alters their charge distribution. Therefore, finding the reactant and solvent charge densities poses a tedious self-consistent problem: to find $\tilde{q}_r$, we must maximize the exponent of Eq. (3.12) over all q_r while always obeying the condition of Eq. (3.13).

The simplest and most popular approach is to simply ignore the polarizability of the reactants and to use their optimal configurations in vacuum. That is, we choose $\tilde{q}_r$ to minimize H_r alone. This "rigid solute" approximation, when compared to the full self-consistent minimization required to find the true $\tilde{q}_r$, creates absolute errors in the free energy ranging from 0.26 eV to 0.52 eV, but differences of free energies are generally accurate to better than 0.03 eV [92], which is sufficiently accurate for our study.

3.3 Results and Discussion

3.3.1 Reaction profile and barrier

Our choice of Fourier representation together with the accurate solution of Eq. (3.15) based on conjugate gradients minimization techniques assures us that within the approximations of (a) a dielectric continuum model, (b) a "rigid solute", and (c) periodic boundary conditions, we are solving the Poisson equation to machine precision.

Based on the above methodology, we first perform *ab initio* calculations at a set of reaction coordinates in the range 1.4 Å $< \lambda <$ 2.0 Å. We also perform one calculation at $\lambda = 7.5$ Å, the largest separation possible in our cell, which serves as our reference configuration and which we label as $\lambda = \infty$ (reactants at essentially infinite separation). The *ab initio* energy of each configuration is then G_0^i in Eq. (3.4).

Next, using the *ab initio* charge density for each λ, we create appropriate dielectric cavities for various values of the bulk dielectric constant ϵ_b as described above. We solve the Poisson equation for each case and calculate the solvation free energy G_{solv}^i.

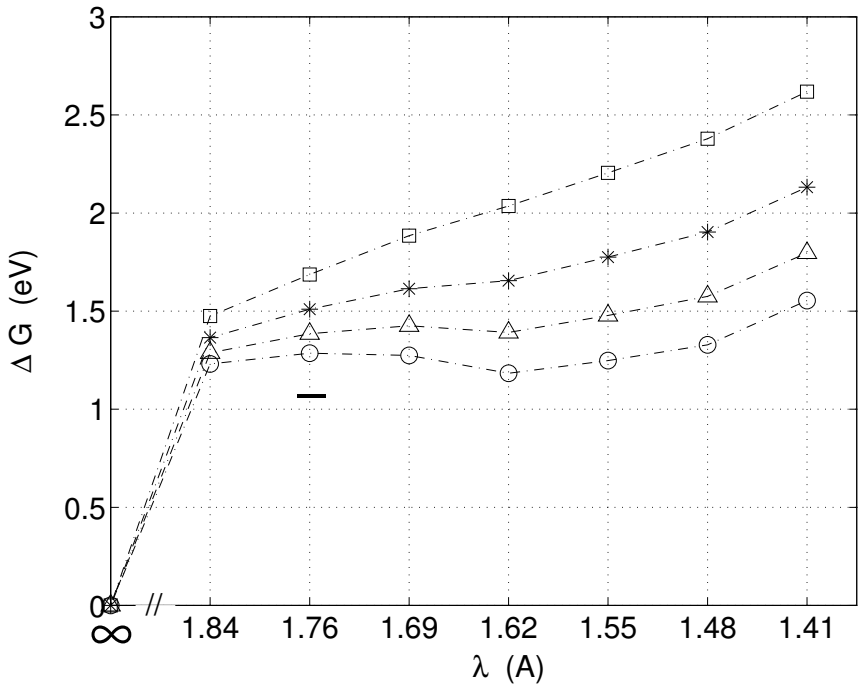

Figure 3-5: Free energies changes ΔG of the reacting complex versus reaction coordinate λ for various values of the bulk water dielectric constant ϵ_b. Circles are $\epsilon_b = 80$, triangles $\epsilon_b = 5$, stars $\epsilon_b = 2$, and squares $\epsilon_b = 1$. Free energy changes are relative to the reactants at infinite separation $\lambda = \infty$. The solid dash represents the experimental value of the reaction barrier at $T = 388K$ [29] where $\epsilon_b = 52$ [94].

This solvation energy is added to G_0^i to give the total free energy of the configuration G^i. Figure 3-5 shows the resulting free energies of the reacting complex as a function of reaction coordinate λ referenced to the $\lambda = \infty$ configuration for various values of ϵ_b. The curve with $\epsilon_b = 1$ shows the raw *ab initio* energies of each configuration in vacuum G_0^i.

The results in Figure 3-5 show the following trends:

(a) For large values of the dielectric constant ϵ_b, the free energy curve has a local maximum at $\lambda \approx 1.7$ Å followed by a shallow minimum at smaller λ. The height of this maximum is the activation free energy ΔG^* of Eq. (3.3) for our reaction.

(b) As ϵ_b decreases, the height of the maximum and therefore the activation barrier increase. This is because a weaker dielectric will lead to a weaker induced charge in the solvent, to a smaller solvation energy, and to less stabilization of the reacting

complex.

(c) As ϵ_b decreases, the maximum and minimum move closer and finally coalesce. The free energy curve then becomes rather featureless and rises monotonically for decreasing λ.

The trends (b) and (c) explain the qualitative behavior of the hydrolysis reaction of Eq. (3.2). As the temperature T increases, the dielectric constant of water drops sharply from its ambient value of $\epsilon_b = 80$ to $\epsilon_b \approx 1$. As ϵ_b decreases, the height of the barrier ΔG^* rises which drastically reduces the reaction rate.

As for quantitative comparison with experimental values, the work of reference [29] found an activation barrier of $\Delta G^* = 1.11 \pm 0.02$ eV at $T = 388K$, which corresponds to $\epsilon_b = 52$ [94]. This experimental value is marked as the solid dash in Figure 3-5. The *ab initio* results predict a barrier $\Delta G^* = 1.3$ eV. The local-density approximation used in the *ab initio* calculations is generally considered to be accurate to within 0.1-0.2 eV for such a system. In addition, the other approximations listed above together with their appropriate uncertainties can introduce a further uncertainty of 0.06 eV. Therefore, our *ab initio* prediction of the barrier compares favorably with the experimental value.

Trend (c) raises the question of the nature of the reaction of Eq. (3.2) for high enough temperatures or low enough dielectric ϵ_b. A simple interpretation of the data in Figure 3-5 based on transition state theory would imply that there is no stable product and that the reaction comes to a halt. However, since we have not explored the entire possible phase space for the two step reaction of Eq. (3.1), it would be excessive to claim that the reaction stops completely. More probably, for this high temperature regime, some other reaction path or mechanism will begin to compete with the one we examine here.

In summary, the results of applying our *ab initio* method explain the behavior of the hydrolysis reaction Eq. (3.1) as a function of temperature. Furthermore, the quantitative comparison to available experimental data is favorable given the approximations involved.

3.3.2 Kirkwood theory

The idea of modeling a solvent by a dielectric continuum dates as far back as the works of Born [12] and Bell [8]. Kirkwood [51] developed a picture for the interaction of a dielectric continuum with a quantum mechanical solute. Within Kirkwood's formulation, the solvent-excluded cavity is a sphere of radius R. The dielectric inside is $\epsilon = 1$, and outside the sphere the dielectric takes its bulk value $\epsilon = \epsilon_b$. One then performs a multipole expansion of the solute charge density. Solving for the induced surface charge density on the spherical boundary is then a standard textbook problem. The resulting expression for the electrostatic solvation free energy G^i_{solv} is

$$G^i_{solv} = \frac{Q_i^2 e^2}{2R}\left(\frac{1}{\epsilon_b} - 1\right) + \frac{p_i^2}{R^3}\left(\frac{1-\epsilon_b}{1+2\epsilon_b}\right) + ... \tag{3.21}$$

where $Q_i e$ is the total charge of the solute, $\vec{p}_i$ is its electrical dipole moment, and contributions of higher moments of the solute charge distribution are not shown. Generally, the leading term of the series dominates and higher order terms can be neglected. In our case, we have neutral but polar reactants CH_2Cl_2 and H_2O, so that the series begins with the dipolar term, the only term retained in the the Kirkwood model below.

The main problem with the Kirkwood formalism involves how one should to choose the cavity radius R. We choose R such that the volume of the Kirkwood sphere and our dielectric cavity are the same. We compute the dipole moment $\vec{p}_i$ of the reactant configuration directly from the *ab initio* charge density. The Kirkwood solvation energy is then the dipolar term of Eq. (3.21).

Figure 3-6 is a correlation plot of the Kirkwood solvation free energies versus solvation free energies computed with the *ab initio* dielectric cavity for the configurations used in this study. The most striking feature of the plot is its linearity. The fact that there is an offset (of approximately 0.77 eV) between the two is not relevant since we concentrate on *differences* of free energies. The linear correlation argues that the simplified Kirkwood model is correct in its qualitative description of solvation effects. Most likely, the Kirkwood free energies are too large because the radius R is too

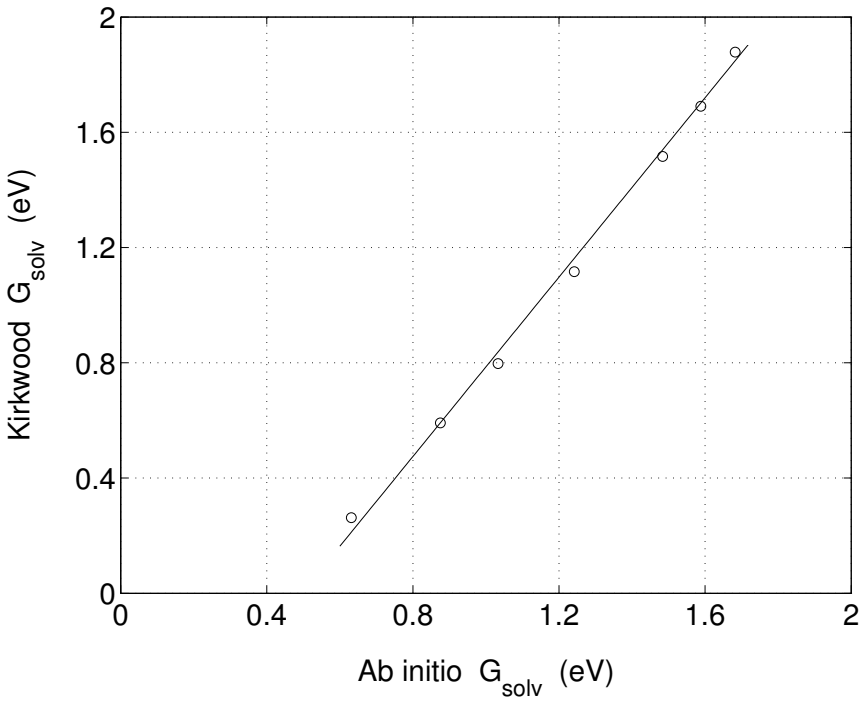

Figure 3-6: Correlation plot of the absolute electrostatic solvation free energies G^i_{solv} as predicted by the *ab initio* model (horizontal) and the Kirkwood model (vertical) for the configurations considered in this work (circles). The line is a least-squares fit with equation $y = 1.56x - 0.77$ eV.

small.

Given the simple form of the dipole term in Eq. (3.21), we can adjust R to modify the scale of the of the solvation energies and thereby create good *a posteriori* agreement with the *ab initio* results. Of course, we do not expect such detailed adjustments are to be transferable to other chemical reactions. Therefore, the conclusion we reach from the strong correlation is that the Kirkwood solvation free energy is a very good *qualitative* indicator of the true solvation free energy and that it can help establish trends in solvation behavior.

3.3.3 Comparison of dielectric models

We now examine the simplifications used to arrive at the Kirkwood model and study each one separately. The two key simplifications are (a) the replacement of the

(I)	(II)
$\epsilon = ab\ initio$ $\rho_r = ab\ initio$ $\phi_s = ab\ initio$ (SC) ΔG	$\epsilon = ab\ initio$ $\rho_r =$ dipole $\phi_s = ab\ initio$ (NSC) $1.15\Delta G$
$1.56\Delta G$ $\epsilon =$ sphere $\rho_r =$ dipole $\phi_s =$ dipole (SC)	$2.28\Delta G$ $\epsilon = ab\ initio$ $\rho_r =$ dipole $\phi_s =$ dipole (SC)
(IV)	(III)

Table 3.2: Comparison of the four dielectric models. ϵ indicates the shape of the dielectric cavity, ρ_r indicates the reactant charge density, and ϕ_s indicates the induced solvent potential. SC means that ϕ_s is the self-consistent solution of the Poisson equation with the ρ_r and ϵ for that model, and NSC means the contrary. Given a free energy difference ΔG between two configurations as predicted by model (I), the free energy differences for the same to configurations as calculated in the other models are also shown.

detailed reactant charge distribution by a dipole, and (b) the replacement of the arbitrarily shaped dielectric cavity by a spherical one.

Our calculations provide us with the reactant charge density ρ_r, the dielectric function ϵ, and the induced solvent potential ϕ_s for each reactant configuration. Therefore, we are poised to study these various approximations to the reactant charge density and the dielectric function which are not unique to the Kirkwood model but are used in many other solvation models in the literature [92]. As we shall see, in going from the *ab initio* to the Kirkwood model, there are two other intermediate models to consider. We label the four possible models (I), (II), (III), and (IV) and describe them in detail below. The four possibilities are outlined in Table 3.2.

(I) This is our *ab initio* model: we use the *ab initio* reactant charge density and dielectric cavity and we solve the Poisson equation to obtain the solvation free energy. In addition, we extract the electrostatic potential ϕ_s created by solvent which we will

use below.

(II) Here, we replace the reactants' charge density by its dipole moment while holding the dielectric cavity and solvent potential ϕ_s fixed at their *ab initio* values of model (I). We calculate the dipole moment from the *ab initio* charge density, and we replace the charge density by this dipole. The dipole is placed at the geometric center of mass of the cavity. We use ϕ_s from (I) and its interaction with the dipole to compute the solvation free energy. Hence, in going from model (I) to (II), we gauge the effect of replacing the charge density by its dipole moment.

(III) Next, we start with the dielectric and dipole of model (II), but we now solve the Poisson equation to self-consistency. That is, we allow ϕ_s to change and become the appropriate self-consistent potential for the diploe in the center of the dielectric cavity. We then evaluate the solvation free energy from Eq. (3.14). Thus, in going from (II) to (III), we study the importance of self-consistency.

(IV) The final possible change involves the cavity shape. Here, we replace the dielectric cavity of (III) by a sphere of equal volume and arrive at the Kirkwood model. The solvent field ϕ_s is the self-consistent solution for this spherical problem. In going from model (III) to (IV), we gauge the effect of cavity shape.

Upon comparing the solvation free energies predicted by models (I)-(IV), we find the same strong linear correlations we found earlier when comparing the *ab initio* and Kirkwood models (c.f. Figure 3-6 in the previous section). Since we are only interested in free energy differences, the slopes of the lines of best fit are the relevant parameters for comparing the models.

In Table 3.2, we present the free energy differences predicted by each model: given a free energy difference of ΔG between two configurations as calculated in model (I), we list the free energy difference between the same configurations as calculated by models (II)-(IV). The prefactor of ΔG is the slope of the line of best fit when we perform a correlation plot of free energy changes as exemplified by Figure 3-6.

Upon examining the table, we observe that: (a) the effect of replacing the detailed *ab initio* charge density by its dipole moment (from (I) to (II)) is small: the error is only 15%; (b) the effect of the shape of the dielectric cavity (from (III) to (IV))

is larger and the changes are $\approx 50\%$; (c) the most important effect is that of self-consistency (from (II) to (III)): the changes are $\approx 100\%$; and (d) the increase from (II) to (III) is canceled by the decrease from (III) to (IV).

These observations lead us the following conclusions. Observations (a) and (c) show that if we replace the *ab initio* charge density by its dipole moment and place this dipole in a dielectric cavity which was originally chosen to mold specifically to the *ab initio* charge distribution, we will have rather large errors in free energies. In addition, the largest part of the error is not due to the fact that the charge density has been simplified but rather that the cavity was designed for use with a different charge density.

A further sign of the inappropriateness of the cavity to a dipole charge is evinced by the fact that changing the cavity to a sphere of equivalent volume reduces the errors in free energies. This argues that the cavity surface must have inward-bulging regions that reflect the non-spherical shape of the *ab initio* charge density. When we use the dielectric cavity with the dipole, the inward-bulging regions must approach the localized dipole too closely and hence lead to a unphysically large interactions between dipole and induced charges. Thus, when we reduce the extended *ab initio* charge distribution to a localized dipole, we are better off choosing a spherical cavity which is equidistant from the localized charge. In summary, simplifying the cavity shape and reactant charge densities can provide reasonable free energies only if the dielectric cavity is used in conjunction with the charge density it was designed to mold to.

3.4 Conclusions

We have presented an *a priori* approach to the problem of the calculation of solvation free energies using continuum dielectric models coupled to a quantum mechanical description of solvated molecules. We provide a derivation of the dielectric treatment based on a coarse-graining of the molecular description of the solvent. Our method for creating the dielectric cavity does not rely on empirically scaled van der Waals

radii but rather uses the electron density of the reactants as the physical variable describing the cavity. In addition, the precise choice of cavity is made by ensuring that the dielectric reproduces the correct linear response of the solvent to electrostatic perturbations.

As a model application, we study the hydrolysis of methylene chloride, of interest in super critical oxidation experiments, and which has shown surprising solvation effects close to criticality. Using our *ab initio* methodology, we find results in good agreement with available experimental reaction barriers. We then study, in a controlled manner, the relative importance of further approximations that are routinely performed in the literature including the use of spherical cavities, the replacement of the reactants by a dipole, or the neglect of self consistency in solving the electrostatic equation.

Chapter 4

Ab Initio Study of Screw Dislocations in Mo and Ta

The microscopic origins of plasticity are far more complex and less well understood in bcc metals than in their fcc and hcp counterparts. For example, slip planes in fcc and hcp metals are almost invariably close-packed, whereas in bcc materials many slip systems can be active. Moreover, bcc metals violate the Schmid law that the resistance to plastic flow is constant and independent of slip system and applied stress [87].

Detailed, microscopic observations have established that in bcc metals at low temperatures, long, low-mobility $\langle 111 \rangle$ screw dislocations control the plasticity [96, 26]. Over the last four decades, the dominant microscopic picture of bcc plasticity involves a complex core structure for these dislocations. The key ingredient of this intricate picture is an extended, non-planar sessile core which must contract before it moves. The first such proposed structure respected the symmetry of the underlying lattice and extended over many lattice constants [41]. More recent and currently accepted theories, based on interatomic potentials, predict extension over several lattice constants and spontaneously broken lattice symmetry [96, 26, 101, 69]. While these models can explain the overall non-Schmid behavior, their predicted magnitude for the critical stress required to move dislocations (*Peierls stress*) is uniformly too large by a factor of about three when compared to experimental yield stresses extrapolated

to zero-temperature [26, 7].

We take the first *ab initio* look at dislocation core structure in bcc transition metals. Although we study two metals with quite different mechanical behavior, molybdenum and tantalum, a consistent pattern emerges from our results which, should it withstand the test of time, will require rethinking the presently accepted picture. Specifically, we find screw dislocation cores with compact structures, without broken symmetry, and with energy scales which appear to be in much better accord with experimental Peierls barriers.

4.1 *Ab initio* methodology

Our *ab initio* calculations for Mo and Ta are carried out within the total-energy plane-wave density functional pseudopotential approach [77], using the Perdew-Zunger [78] parameterization of the Ceperley-Alder [15] exchange-correlation energy. Non-local pseudopotentials of the Kleinman-Bylander form [53] are used with s, p, and d channels. The Mo potential is optimized according to [80] and the Ta potential is from [100]. We use plane wave basis sets with energy cutoffs of 45 Ryd for Mo and 40 Ryd for Ta to expand the wave functions of the valence (outermost s and d) electrons. Calculations in bulk show these cutoffs to give total system energies to within 0.01 eV/atom. We carry out electronic minimizations using the analytically continued approach [4] within a new, object-oriented approach to calculations within density-functional theory, the DFT++ formalism [49].

To gauge the reliability of the pseudopotentials, Table 4.1 displays our *ab initio* results for the materials' lattice constants and those elastic moduli most relevant for the study of $\langle 111 \rangle$ screw dislocations. The tabulated moduli describe the long-range elastic fields of the dislocations (K), the coupling of displacement gradients along the dislocation axis z to core-size changes in the orthogonal x, y plane ($c_{xx,zz} = (c_{11} + 5c_{12} - 2c_{44})/6$), and the coupling of core-size changes to themselves in the plane ($c_{xx,xx} = (c_{11} + c_{12} + 2c_{44})/2$ and $c_{xx,yy} = (c_{11} + 2c_{12} - 2c_{44})/3$). These results indicate that our predicted core energy differences should be reliable to within better than

	Mo			Ta		
	AI	Expt	Error	AI	Expt	Error
a	3.10	3.15	-1.6%	3.25	3.30	-1.5%
K	1.60	1.36	18%	0.65	0.62	5%
$c_{xx,zz}$	2.17	1.91	14%	1.72	1.39	24%
$c_{xx,xx}$	5.48	4.25	29%	3.02	2.98	1.3%
$c_{xx,yy}$	2.21	1.77	25%	1.72	1.49	15%

Table 4.1: Lattice constants a (Å) and elastic moduli (Mbar) for Mo and Ta based on *ab initio* pseudopotentials (AI) and experiments (Expt) [100].

$\sim 30\%$, which suffices for the purposes of our study.

4.2 Preparation of dislocation cells

The cell we use for dislocation studies has lattice vectors $\vec{a}_1 = 5a[1,-1,0]$, $\vec{a}_2 = 3a[1,1,-2]$, and $\vec{a}_3 = a[1,1,1]/2$, where a is the lattice constant. We call this ninety-atom cell the "5×3" cell in reference to the lengths of $\vec{a}_1$ and $\vec{a}_2$, and the Burgers vectors of all of the dislocations in our work are along $\vec{a}_3$. Eight k-points $k_1 = k_2 = \frac{1}{4}, k_3 \in \pm\{\frac{1}{16}, \frac{3}{16}, \frac{5}{16}, \frac{7}{16}\}$ sample the Brillouin zone in conjunction with a non-zero electronic temperature of $k_B T = 0.1$ eV, which facilitates the sampling of the Fermi surface. These choices give total energies to within 0.01 eV/atom.

Given the relatively small cell size, we wish to minimize the overall strain and the effects of periodic images. We therefore follow [11] and employ a quadrupolar arrangement of dislocations (a rectangular checkerboard pattern in the $\vec{a}_1, \vec{a}_2$ plane). This ensures that dislocation interactions enter only at the quadrupolar level and that the net force on each core is zero by symmetry, thereby minimizing perturbations of core structure due to the images. As was found in [11] and as we explore in detail below, we find very limited impact of finite-size effects on the cores when following this approach.

In bcc structures, screw dislocations are known to have two inequivalent core configurations, termed "easy" and "hard" [96, 101, 69]. These cores can be obtained

from one another by reversing the Burgers vector of a dislocation line while holding the line at a fixed position. We produce cells with either only easy or only hard cores in this way. To create atomic structures for the cores, we proceed in three stages. First, we begin with atomic positions determined from isotropic elasticity theory for our periodic array of dislocations. Next, we relax this structure to the closest local energy minimum within the interatomic MGPT model for Mo [101]. Since we do not have an interatomic potential for Ta and expect similar structures in Ta and Mo [69], we create suitable Ta cells by scaling the optimized MGPT Mo structures by the ratio of the materials' lattice constants. Finally, we perform standard *ab initio* atomic relaxations on the resulting MGPT structures until all ionic forces in all axial directions are less than 0.06 eV/Å.

4.3 Extraction of core energies

The energy of a long, straight dislocation line with Burgers vector $\vec{b}$ is $E = E_c(r_c) + Kb^3 \ln(L/r_c)$ per b along the line [42], where L is a large-length cutoff, and K is an elastic modulus (see Table 4.1) computable within anisotropic elasticity theory [38, 39]. The core radius r_c is a short-length cutoff inside of which the continuum description fails and the discrete lattice and electronic structure of the core become important. $E_c(r_c)$ measures the associated "core energy", which, due to severe distortions in the core, is most reliably calculated by *ab initio* methods.

The energy of our periodic cell contains both the energy of four dislocation cores and the energy stored outside the core radii in the long-range elastic fields. To separate these contributions, we start with the fact that two straight dislocations at a distance d with equal and opposite Burgers vectors have an anisotropic elastic energy per b given by $E = 2E_c(r_c) + 2Kb^3 \ln(d/r_c)$. Next, by regularizing the infinite sum of this logarithmically divergent pair interaction, we find that the energy per dislocation per b in our cell is given by

$$E = E_c(r_c) + Kb^3 \left[\ln\left(\frac{|\vec{a}_1|/2}{r_c} \right) + A\left(\frac{|\vec{a}_1|}{|\vec{a}_2|} \right) \right]. \tag{4.1}$$

E_c (eV/b)	5×3	9×5	Cylindrical [101, 102]
hard	2.62	2.57	2.66
easy	2.35	2.31	2.42
Δ	0.27	0.26	0.24

Table 4.2: Core energies for $r_c = 2b$ as predicted by the MGPT model. "easy" and "hard" refer to different core configurations. Δ is the hard–easy core energy difference.

The function $A(x)$ contains all the effects of the infinite Ewald-like sums of dislocation interactions and has the value $A = -0.598\,846\,386$ for our cell. Subtracting the long-range elastic contribution (the second term of (4.1)) from the total energy, we arrive at the core energy E_c.

To test the feasibility of this approach, we compare $E_c(r_c)$ for the MGPT potential as extracted with the above procedure from cells of two different sizes: the 5×3 cell and the corresponding 9×5 cell. (The MGPT is *fit* to reproduce experimental elastic moduli, so K is given in Table 4.1.) With the choice $r_c = 2b$, Table 4.2 shows that our results, even for the 5×3 cell, compare quite favorably with those of [101, 102], especially given that our 5×3 and 9×5 cells contain only ninety and 270 atoms respectively, whereas the cited works used cylindrical cells with a single dislocation and *two thousand* atoms or more. Given the suitability of the 5×3 cell, all *ab initio* results reported below are carried out in this cell.

4.4 *Ab initio* core energies

Except for the Mo hard core, all the core structures relax quite readily from their MGPT configurations to their equilibrium *ab initio* structures. The Mo hard-core configuration, however, spontaneously relaxes into easy cores, strongly indicating that the hard core, while meta-stable within MGPT by only 0.02 eV/b [102], is not stable in density functional theory. We do not believe that this instability is due to finite-size effects, which appear to be quite small for the reasons outlined previously.

Table 4.3 compares our *ab initio* results to available MGPT results for core energies

	Mo			Ta	
E_c (eV/b)	MGPT	AI*	AI	MGPT [69]	AI
hard	2.57	2.94	–	–	0.91
easy	2.35	2.86	2.64	–	0.86
Δ	0.22	0.08	–	0.14	0.05

Table 4.3: Core energies for $r_c = 2b$ for fully relaxed *ab initio* cores (AI) and interatomic (MGPT) cores in the 5×3 cell. AI* refers to *ab initio* core energies computed based on relaxed MGPT configurations as the *ab initio* Mo hard core is unstable. Ref. [69] only reported Δ for Ta.

in Mo and Ta. To make comparison with the MGPT, for the unstable Mo hard core we evaluate the *ab initio* core energy at the optimal MGPT atomic configuration (column AI* in Table 4.3). Note that, in computing hard–easy core energy differences, the long-range elastic contributions cancel so that these differences are much better converged than the absolute core energies.

Table 4.3 shows that the MGPT hard–easy core energy differences are much larger than the corresponding *ab initio* values by approximately a factor of three. The accuracy of the elastic moduli of Table 4.1, combined with the high transferability of the local-density pseudopotential approach, indicates that this factor of three is not an artifact of our approximations. We believe that the reason for this discrepancy is that the MGPT is less transferable. Having been forced to fit *bulk* elastic moduli and thus long-range distortions, the MGPT may not describe the short wavelength distortions in the cores with high accuracy. An examination of Mo phonons along [100] provides poignant evidence: the MGPT frequencies away from the zone center are too large when compared to experimental and band-theoretic values [101] and translate into spring constants that are up to approximately three times too large.

The magnitude of the core energy difference has important implications for the magnitude of the Peierls energy barrier and Peierls stress for the motion of screw dislocations in Mo and Ta. In a recent Mo MGPT study [102], the most likely path for dislocation motion was identified to be the $\langle 112 \rangle$ direction: the moving dislocation core changes from easy to hard and back to easy as it shifts along $\langle 112 \rangle$.

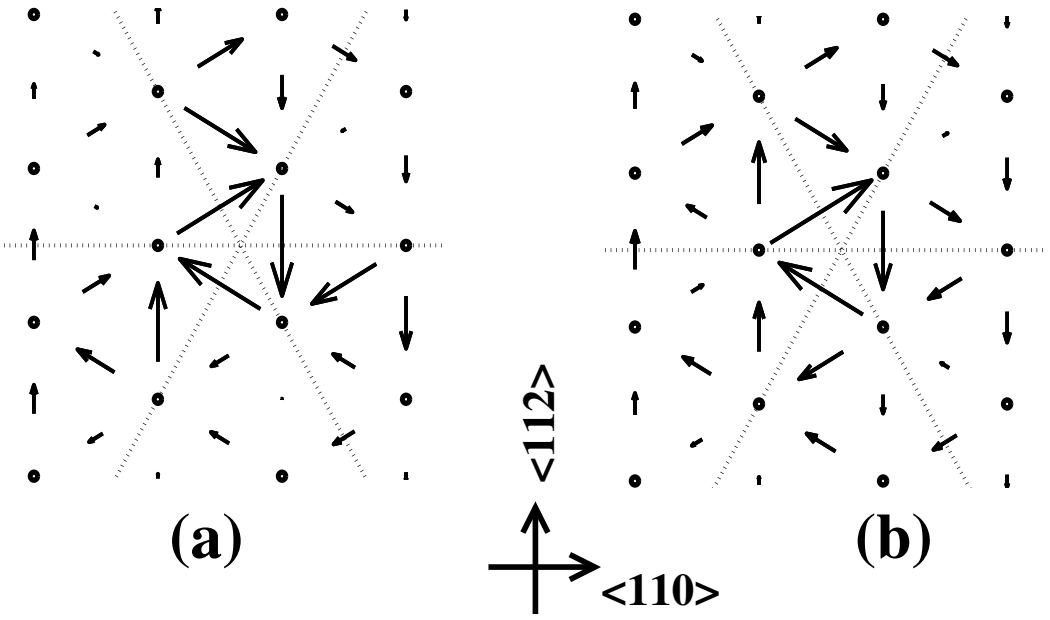

Figure 4-1: DD maps as found in the MGPT model for (a) easy, and (b) hard Mo dislocation cores. Dotted lines indicate axes of C_2 symmetry of the D_3 symmetry group.

The energy barrier was found to be 0.26 eV/b, very close to the MGPT hard–easy energy difference itself. The fact that the *ab initio* hard–easy energy differences in Mo and Ta are smaller by about a factor of three than the respective interatomic values suggests that the *ab initio* energy landscape for the process has a correspondingly smaller scale. If so, the Peierls stress in Mo and Ta should also be correspondingly smaller and in much better agreement with the values suggested by experiments.

4.5 Dislocation core structures

Figure 4-1 shows differential displacement (DD) maps [96] of the core structures we find in our ninety-atom supercell when working with the interatomic MGPT potential for Mo. Our DD maps show the atomic structure projected onto the (111) plane. The vector between a pair of atomic columns is proportional to the change in the [111] separation of the columns due to the presence of the dislocations. The maps show that both easy and hard cores have approximate 3-fold rotational (C_3) point-

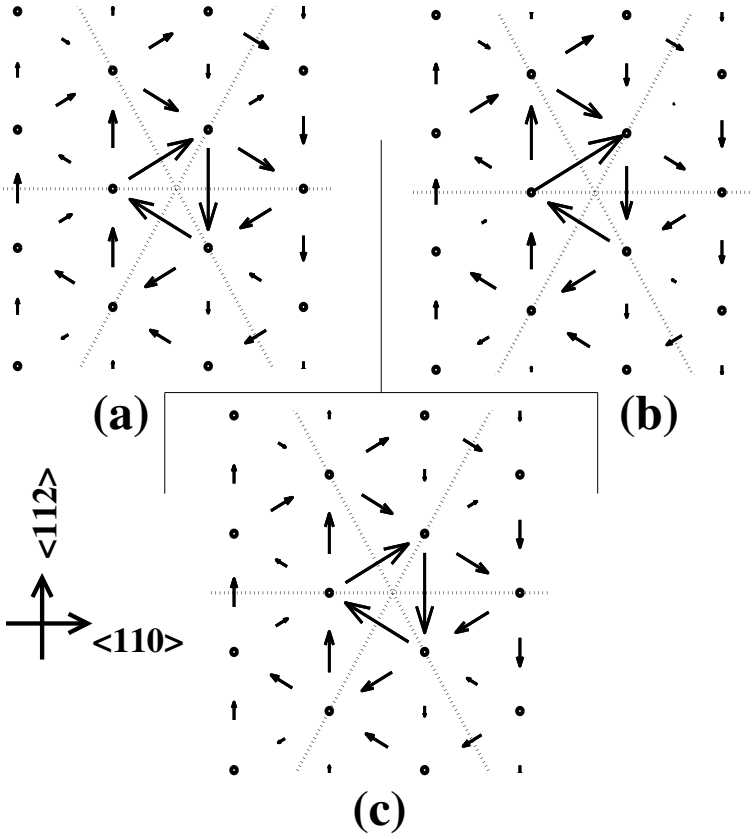

Figure 4-2: DD maps of the *ab initio* dislocation cores: (a) Ta easy, (b) Ta hard; and (c) Mo easy. Dotted lines indicate axes of C_2 symmetry of the D_3 symmetry group.

group symmetry about the out-of-page [111] axis through the center of each map. The small deviations from this symmetry reflect the weakness of finite-size effects in our quadrupolar cell. The hard core has three additional 2-fold rotational (C_2) symmetries about the three $\langle 110 \rangle$ axes marked in the maps, increasing its point-group symmetry to the dihedral group D_3 which is shared by the underlying crystal. The easy core, however, shows a strong spontaneous breaking of this symmetry: its core spreads along only three out of the six possible $\langle 112 \rangle$ directions. Our results reproduce those of [101, 102] who employed much larger cylindrical cells with open boundaries, underscoring the suitability of our cell for determining core structure. This symmetry-breaking core extension is that which has been theorized to explain the relative immobility of screw dislocations and violation of the Schmid law in bcc metals.

Figure 4-2 displays DD maps of our *ab initio* core structures. Contrary to the atomistic results, we find that the low-energy easy cores in Mo and Ta have full D_3 symmetry and do not spread along the $\langle 112 \rangle$ directions. Combining this with the above results concerning core energetics, we have two examples for which our pseudopotentials are sufficiently accurate to disprove the conventional wisdom that generic bcc metallic systems *require* broken symmetry in the core to explain the observed immobility of screw dislocations.

Turning to the hard core structures, the *ab initio* resuts for Ta show a significant distortion when compared to the atomistic core (contrast Figure 4-1b and Figure 4-2b). As the *ab initio* Mo hard core was unstable, we believe that this distortion of the Ta hard core suggests that this core is much less stable within density functional theory than in the atomistic potentials.

To complete the specification of the three-dimensional *ab initio* structure of easy cores in Mo and Ta, Figure 4-3 presents maps of the atomic displacement in the (111) plane. The small atomic shifts, which are due entirely to anisotropic effects, are shown as in-plane vectors centered on the bulk atomic positions and magnified by a factor of *fifty*. To reduce noise in the figure, before plotting we perform C_3 symmetrization of the atomic positions about the [111] axis passing through the center of the figure. As all the dislocation cores in our study have a minimum of C_3 symmetry, this procedure does not hinder the identification of possible spontaneous breaking of the larger D_3 symmetry group. Our maps indicate that the easy cores in both Mo and Ta have full D_3 symmetry.

Recent high-resolution electron microscopy explorations of the symmetry of dislocations in Mo have focused on the small shifts in the (111) plane of columns of atoms along [111] [88]. This pioneering work reports in-plane displacements extending over a range much greater than the corresponding MGPT results and also much greater than what we find *ab initio*. In [88] this is attributed to possible stresses from thickness variations and foil bending. We believe this makes study of the internal structure of the core difficult, and that cleaner experimental results are required to resolve the nature of the symmetry of the core and its extension.

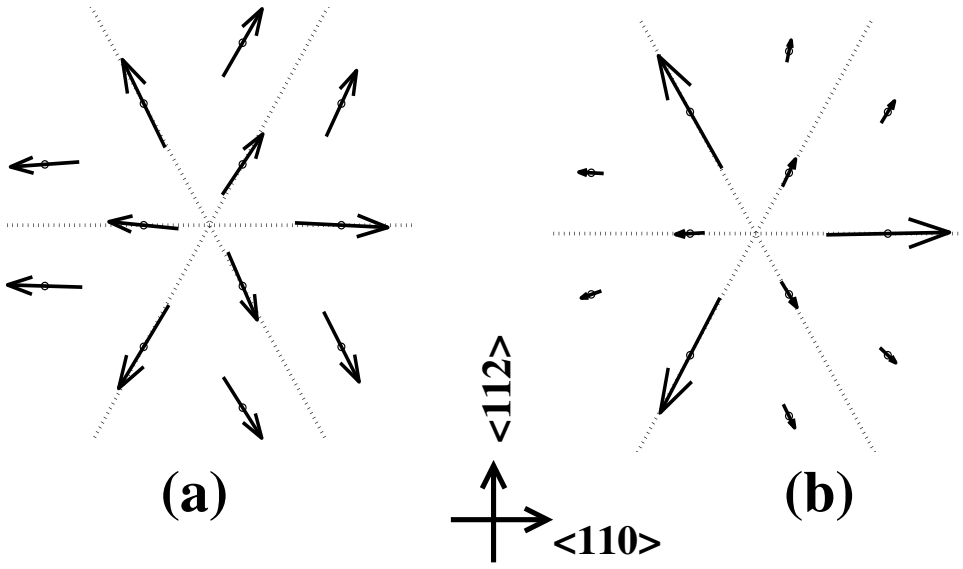

Figure 4-3: Planar displacement maps of the *ab initio* (a) Mo easy, and (b) Ta easy cores. Vectors show in-plane (111) atomic shifts and have been magnified by a factor of fifty. Dotted lines indicate the C_2 axes of the D_3 symmetry group.

4.6 Conclusions

In conclusion, our first principles results show no preferential spreading or symmetry breaking of the dislocation cores and exhibit an energy landscape with the proper scales to explain the observed immobility of dislocations. Atomistic models which demonstrate core spreading and symmetry breaking, both of which tend to reduce the mobility of the dislocations, are well-known to over-predict the Peierls stress. The combination of these two sets of observations argues strongly in favor of much more compact and symmetric bcc screw dislocation cores than presently believed.

Chapter 5

New Algebraic Formulation of *Ab Initio* Calculation

This chapter gives a self-contained description of how to build a highly flexible, portable density-functional production code which attains significant fractions of peak performance on scalar cached architectures, shared-memory processors (SMP), and distributed-memory processors (DMP). More importantly, however, this chapter introduces a new formalism, DFT++, for the development, implementation, and dissemination of new *ab initio* generalized functional theoretic techniques among researchers. The most well-known and widely used generalized functional theory (GFT) is density-functional theory, where the energy of the system is parametrized as a functional of the electron density. Although the formalism presented here is applicable to other single-particle GFTs, such as self-interaction correction or Hartree-Fock theory, for concreteness we concentrate primarily on density functional theory (DFT).

This formalism is of particular interest to those on the forefront of exploring new *ab initio* techniques and novel applications of such in the physical sciences. It allows practitioners to quickly introduce new physics and techniques without expenditure of significant effort in debugging and optimizing or in developing entirely new software packages. It does so by providing a new, compact, and explicit matrix-based language for expressing GFT calculations, which allows new codes to be "derived" through straightforward formal manipulations. It also provides a high degree of modularity,

a great aid in maintaining high computational performance.

This language may be thought of as being for GFT what the Dirac notation is for quantum mechanics: a fully explicit notation free of burdensome details which permits the ready performance of complex manipulations with focus on physical content. Direct application of the Dirac notation to GFT is particularly cumbersome because in single-particle theories, the quantum state of the system is not represented by a single ket but rather a collection of kets, necessitating a great deal of indexing. Previous attempts to work with the Dirac notation while eliminating this indexing have included construction of column vectors whose entries were kets [4] but such constructions have proved awkward because, ultimately, kets are members of an abstract Hilbert space and are not the fundamental objects of an actual calculation.

The foundation of the new DFT++ formalism is the observation that all the necessary computations in an *ab initio* calculation can be expressed explicitly as standard linear-algebraic operations upon the actual computational representation of the quantum state without reference to complicated indexing or to the underlying basis set. With traditional approaches, differentiating the energy functional, which is required for self-consistent solution for the single-particle orbitals, is a frequent source of difficulty. Issues arise such as the distinction between wave functions and their duals, covariant versus contravariant quantities, establishing a consistent set of normalization conventions, and translation from continuum functional derivatives to their discrete computational representations. However, by expressing the energy explicitly in our formalism, all these difficulties are automatically avoided by straightforward differentiation of a well-defined linear-algebraic expression.

This new formalism allows not only for ease of formal manipulations but also for direct transcription of the resulting expressions into software, i.e. literal typing of physical expressions in their matrix form into lines of computer code. Literal transcription of operations such as matrix addition and multiplication is possible through the use of any of the modern, high-level computer languages which allows for the definition of new object types (e.g. vectors and matrices) *and* the action of the standard operators such as "+", "-", or "*" upon them. Once the basic operators

have been implemented, the task of developing and debugging is simplified to checking the formulae which have been entered into the software. This allows the researcher to modify or extend the software and explore entirely new physical ideas rapidly. Finally, a very important practical benefit of using matrix operations wherever possible is that the theory of attaining peak performance on modern computers is well developed for matrix-matrix multiplication.

The high level of modularity which naturally emerges within the DFT++ formalism compartmentalizes and isolates from one another the primary areas of research in electronic structure calculation: (i) derivation of new physical approaches, (ii) development of effective numerical techniques for reaching self-consistency, and (iii) optimization and parallelization of the underlying computational kernels. This compartmentalization brings the significant advantage that researchers with specialized skills can explore effectively the areas which pertain to them, without concerning themselves with the areas with which they are less familiar.

A few anecdotes from our own experience serve to illustrate the efficacy of this approach. The extension of our production software to include electron spin through the local spin-density approximation (LSDA) required a student with no prior familiarity with our software only one-half week to gain that familiarity, three days to redefine the software objects to include spin, and less than one day to implement and debug the new physics. The time it took another student to develop, implement, explore, and fine-tune the new numerical technique of Section 5.5.2 was less than a week. Finally, our experience with parallelization and optimization has been similarly successful. To parallelize our software for use with an SMP (using threads) required a student starting with no prior knowledge of parallelization two weeks to develop a code which *sustains* an average per processor FLOP rate of 80% of the processor clock speed. (See Section 5.6.4 for details.) Finally, for massively parallel applications, the development of an efficient DMP code (based on MPI), a task which often requires a year or more, required two students working together approximately two months to complete.

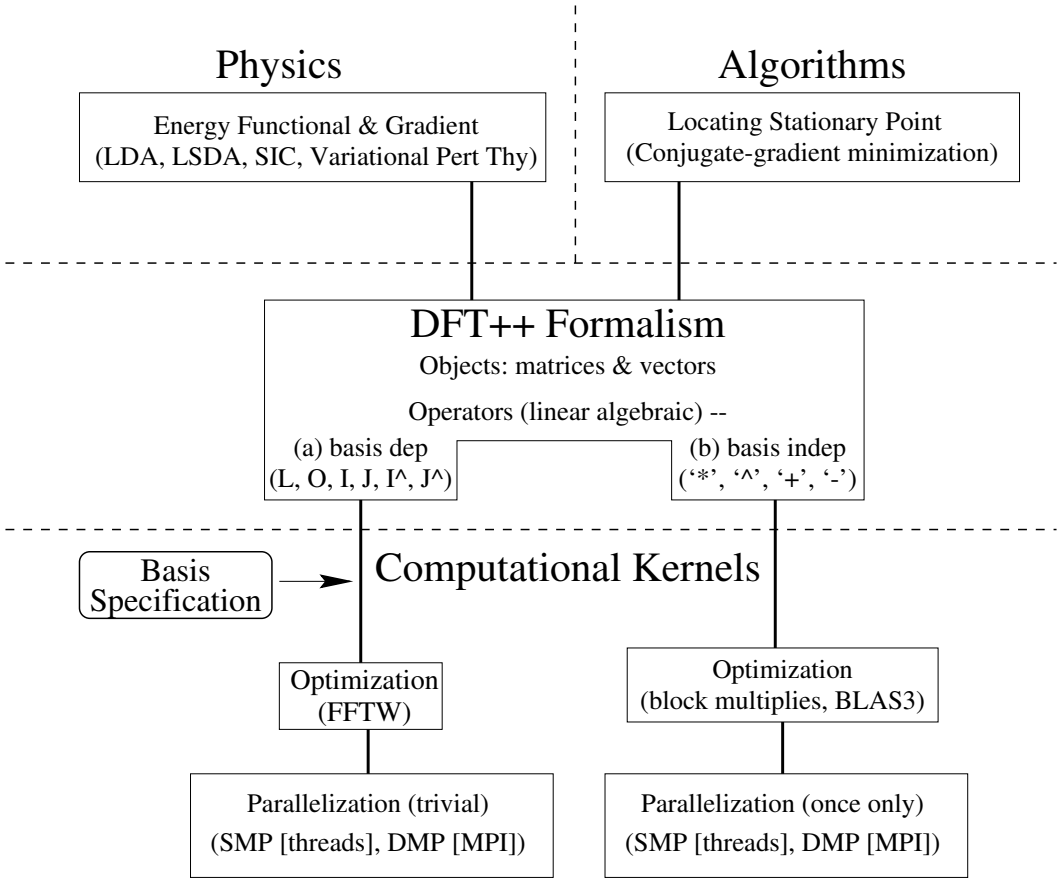

Figure 5-1: Overview of DFT++ formalism

5.1 Overview

Figure 5-1 both illustrates the interconnections among the primary areas of active research in modern electronic structure calculations and serves as a road-map for the content of this chapter. The figure emphasizes how the DFT++ formalism forms an effective central bridge connecting these areas.

Reduction to practice of new physical approaches generally requires expressions for an energy functional and the derivatives of that functional, as indicated in the upper-left portion of the figure. Our discussion begins in Section 5.2 with an exposition of the mathematical framework which we employ throughout this work, a Lagrangian formulation of generalized density functional theories. In Section 5.3 we introduce our matrix-based formalism using density-functional theory (DFT) within the local-

density approximation (LDA) [58] as a case study, deriving the requisite expressions for the energy functional and its gradient.

In Section 5.4, we go on to consider several examples of other functionals for physical calculations, including the local spin-density approximation (LSDA), self-interaction correction (SIC), density-functional variational perturbation theory, and band-structure calculations. We derive the requisite expressions for the corresponding functionals and their derivatives in the space of a few pages and thereby show the power and compactness of our formulation for the treatment of a wide range of single-particle quantum mechanical problems.

As mentioned in the introduction, our matrix-based formalism allows the relevant formulae to be literally typed into the computer. Because these formulae are self-contained, we can make, as illustrated in the upper-right portion of the figure, a clear distinction between the expression of the physics itself and the algorithms which search for the stationary point of the energy functional to achieve self-consistency. For concreteness, in Section 5.5 we provide full specification for both a preconditioned conjugate-gradient minimization algorithm and a new algorithm for accelerating convergence when working with metallic systems.

Due to our matrix-based formulation, the expressions for the objective function and its derivatives are built from linear-algebraic operations involving matrices. As the lower portion of Figure 5-1 illustrates, the DFT++ formalism clearly isolates the software which contains the actual computational kernels. These kernels therefore may be optimized and parallelized independently from all other considerations.

Section 5.6 describes these computational considerations in detail. In Section 5.6.1 we discuss the use of object-oriented languages for linking the underlying computational kernels with higher level physical expressions. Section 5.6.2 discusses the scaling with physical system size of the burden for the most time consuming computational kernels. There are in fact two distinct types of computational kernels, both of which appear in the lower portion of the figure.

The first type are kernels which implement those few operators in our formalism that depend on the choice of basis set (L, $\mathcal{O}$, $\mathcal{I}$, $\mathcal{J}$, $\mathcal{I}^\dagger$, $\mathcal{J}^\dagger$, defined in Section 5.3.1).

These kernels represent the only entrance of basis-set details into the overall framework. (Appendix A provides the requisite details for plane-wave calculations.) This allows for coding of new physics and algorithms without reference to the basis and for a single higher-level code to be used with "plug-ins" for a variety of different basis sets. The application of the basis-dependent operators can be optimized as discussed in Section 5.6.3 by calling standard packages such as FFTW [30]. Parallelization for the basis-dependent operators is trivial because they act in parallel on all of the electronic wave functions at once. Section 5.6.4 discusses such parallelization for SMP and DMP architectures.

Finally, the second class of kernels are basic linear-algebraic operations (e.g. matrix multiplication '*', addition '+', subtraction '-', and Hermitian-conjugated multiplication '^') which do not in any way depend on the basis set used for the calculation. As such, the work of optimization and parallelization for these kernels need only be performed once. Section 5.6.3 presents the two strategies we employ for these optimizations: blocking of matrix multiplication and calling optimized linear-algebra packages such as BLAS3. Parallelization of these operations is not trivial because data-sharing or communication is required between processors. We detail high performance strategies for dealing with this issue in Sections 5.6.4 for both SMP and DMP architectures.

5.2 Lagrangian formalism

The traditional equations of density-functional theory are the Kohn-Sham equations [58] for a set of effective single-particle electronic states $\{\psi_i(r)\}$. Below, when we refer to "electrons", we are in fact always referring to these effective electronic degrees of freedom. The electrostatic or Hartree field $\phi(r)$ caused by the electrons is traditionally found from solving the Poisson equation with the electron density derived from these wave functions as the source term. The ground-state energy of the system is then found by minimizing the traditional energy functional, which ensures the self-consistent solution of the Kohn-Sham equations. A great advantage of this

variational principle is that first-order errors in the wave functions lead to only second order errors in the energy. However, although not frequently emphasized, errors in solving the Poisson equation due to the incompleteness of the basis set used in a calculation may produce a non-variational (i.e. first-order) error in the energy.

We now consider a new variational principle which ultimately leads to identical results for complete basis sets, but which places $\{\psi_i\}$ and ϕ on an equal footing and has several advantages in practice. The central quantity in this principle is the Lagrangian $\mathcal{L}_{LDA}$ introduced in [62], which within the local-density approximation (LDA), is

$$\begin{aligned}\mathcal{L}_{LDA}\left(\{\psi_i(r)\},\phi(r)\right) &= -\frac{1}{2}\sum_i f_i \int d^3r\ \psi_i^*(r)\nabla^2\psi_i(r) \\ &\quad + \int d^3r\ V_{ion}(r)\, n(r) + \int d^3r\ \epsilon_{xc}(n(r))\, n(r) \\ &\quad - \int d^3r\ \phi(r)\ (n(r)-n_0) - \frac{1}{8\pi}\int d^3r\ \|\vec{\nabla}\phi(r)\|^2 , \quad (5.1)\end{aligned}$$

where the electron density $n(r)$ is defined in terms of the electronic states and the Fermi-Dirac fillings f_i as

$$n(r) = \sum_i f_i \|\psi_i(r)\|^2 . \quad (5.2)$$

Here and throughout this chapter we work in atomic units and therefore have set $\hbar = m_e = e = 1$, where m_e is the electron mass and e is the charge of the proton. Finally, the Kohn-Sham electronic states $\{\psi_i\}$ must satisfy the orthonormality constraints

$$\int d^3r\ \psi_i^*(r)\,\psi_j(r) = \delta_{ij} . \quad (5.3)$$

Above, $\epsilon_{xc}(n)$ is the exchange-correlation energy per electron of a uniform electron gas of electron density n, and $V_{ion}(r)$ is the potential each electron feels due to the ions. The constant n_0 is used in calculations in periodic systems as a uniform positive background that neutralizes the electronic charge density. The effect of this background on the total energy is properly compensated when the Ewald summation is used to compute the interionic interactions.

The following equations, subject to the constraints of Eq. (5.3), specify the stationary point of $\mathcal{L}_{LDA}$,

$$\frac{1}{f_i}\frac{\delta \mathcal{L}_{LDA}}{\delta \psi_i^*(r)} = 0 = \left[-\frac{1}{2}\nabla^2 + V_{ion}(r) - \phi(r) + V_{xc}(r)\right]\psi_i(r) - \epsilon_i \psi_i(r)\ , \quad (5.4)$$

$$\frac{\delta \mathcal{L}_{LDA}}{\delta \phi(r)} = 0 = -(n(r) - n_0) + \frac{1}{4\pi}\nabla^2\phi(r)\,. \quad (5.5)$$

These are seen to be the standard Kohn-Sham eigenvalue equations for $\psi_i(r)$ and the Poisson equation for the Hartree potential $\phi(r)$ where the negative sign in the second equation properly accounts for the negative charge of the electrons.

The behavior of $\mathcal{L}_{LDA}$ is in fact quite similar to that of the traditional LDA energy functional,

$$E_{LDA}\left(\{\psi_i(r)\}\right) = -\frac{1}{2}\sum_i f_i \int d^3r\ \psi_i^*(r)\nabla^2\psi_i(r) + \int d^3r\ V_{ion}(r)\, n(r) + \int d^3r\ \epsilon_{xc}(n(r))\, n(r) + \frac{1}{2}\int d^3r \int d^3r'\ \frac{n(r)n(r')}{\|r - r'\|}\,. \quad (5.6)$$

First, as shown in [2], evaluation of $\mathcal{L}_{LDA}\left(\{\psi_i(r)\}, \tilde{\phi}(r)\right)$, where $\tilde{\phi}$ is the solution of the Poisson equation, recovers the value of the traditional energy functional. Moreover, the derivatives of $\mathcal{L}_{LDA}$ and E_{LDA} are also equal at $\tilde{\phi}$. This result follows by considering a variation of the equality $E_{LDA}\left(\{\psi_i(r)\}\right) = \mathcal{L}_{LDA}\left(\{\psi_i(r)\}, \tilde{\phi}(r)\right)$, which expands into

$$\sum_i \int d^3r \left(\frac{\delta E_{LDA}}{\delta \psi_i(r)}\,\delta\psi_i(r) + \frac{\delta E_{LDA}}{\delta \psi_i^*(r)}\,\delta\psi_i^*(r)\right) = \sum_i \int d^3r \left(\frac{\delta \mathcal{L}_{LDA}}{\delta \psi_i(r)}\,\delta\psi_i(r) + \frac{\delta \mathcal{L}_{LDA}}{\delta \phi_i^*(r)}\,\delta\psi_i^*(r)\right) + \int d^3r\, \frac{\delta \mathcal{L}_{LDA}}{\delta \phi(r)}\,\delta\tilde{\phi}(r)\,.$$

Because Poisson's equation (Eq. (5.5)) is the condition that the functional derivative of $\mathcal{L}_{LDA}$ with respect to ϕ vanishes, the last term on the right-hand side is zero when $\phi = \tilde{\phi}$. Therefore, the functional derivatives of E_{LDA} and $\mathcal{L}_{LDA}$ with respect to the electronic states $\{\psi_i\}$ are equal when evaluated at $\tilde{\phi}(r)$. Finally, the critical points of $\mathcal{L}_{LDA}$ are in one-to-one correspondence with the minima of E_{LDA}. The reason is

that (i) for each fixed $\{\psi_i\}$, there is a unique $\tilde{\phi}$ (up to a choice of arbitrary constant) which solves the Poisson equation because $\mathcal{L}_{LDA}$ as a function of ϕ is a negative-definite quadratic form, and (ii) as we have just seen, at such points the derivatives with respect to $\{\psi_i\}$ of E_{LDA} and $\mathcal{L}_{LDA}$ are identical.

One advantage of placing ϕ and $\{\psi_i\}$ on an equal footing is that now errors in the Lagrangian are second-order in the errors of both the wave functions *and* the Hartree field. Additionally, as a practical matter, one has greater flexibility in locating the stationary point. Rather than solving for the optimal ϕ at each value of $\{\psi_i\}$, as is done in traditional DFT methods, one has the option of exploring in both $\{\psi_i\}$ and ϕ simultaneously. However, some care in doing this is needed, because the stationary points of $\mathcal{L}_{LDA}$ are not extrema but saddle points. (Note that the first term of Eq. (5.1), the kinetic energy, is unbounded above, whereas the last term, the Hartree self-energy, is unbounded below.) This saddle has a particularly simple structure, and a method to exploit this is outlined in [3].

Finally, by allowing ϕ to be a free variable, we have rendered local all interactions among the fields. One great formal advantage is that the subtle mathematical issues in periodic systems arising from the long-range nature of the Coulomb interaction no longer require special treatment. For example, the choice of the neutralizing background n_0 in periodic systems is straightforward and is treated in detail in Section 5.3.3. Because of this and the aforementioned advantages, we will work in the Lagrangian framework throughout this chapter.

5.3 Basis-set independent matrix formulation

Our basis-set independent matrix formalism allows us to express the structure of any single-particle quantum theory in a compact and explicit way. In this section, we apply it to the Lagrangian of Eq. (5.1) which contains energetic terms and non-linear couplings that are common to all such theories.

To make progress, we first must deal with the fact that the Lagrangian is a function of continuous fields. When we perform a computation, we are forced to represent these

fields in terms of expansions within a finite basis set. Denoting our basis functions as $\{b_\alpha(r)\}$, where Greek letters index basis functions, we expand the wave functions and Hartree potential in terms of expansion coefficients $C_{\alpha i}$ and ϕ_α through

$$\psi_i(r) = \sum_\alpha b_\alpha(r)\, C_{\alpha i} \quad , \quad \phi(r) = \sum_\alpha b_\alpha(r)\, \phi_\alpha \ . \tag{5.7}$$

Typical and popular choices of basis functions are plane waves (i.e. Fourier modes) [77], finite-element functions [98, 16], multiresolution analyses [2], or Gaussian orbitals [20, 99]. Once a basis set has been chosen, $\mathcal{L}_{LDA}$ becomes an explicit function of the finite set of variables $C_{\alpha i}$ and ϕ_α.

In addition to the basis set itself, we require a grid of points p in real space covering the simulation cell. This grid is necessary for a number of operations, such as for computing the values of the wave functions or the electron density in real-space and for computing the exchange-correlation energy density $\epsilon_{xc}(n(r))$ of Eq. (5.1), which is a non-algebraic function of the electron density $n(r)$ and can only be computed point by point on the real-space grid.

Our aim is to find a compact, matrix-based notation that works in the space of expansion coefficients $C_{\alpha i}$ and ϕ_α and is thus applicable to any calculation within any basis set. In the course of doing so, we will be able to clearly identify which parts of our formalism require information about the particular basis that is chosen and which parts are completely general and independent of this choice. In addition, when we have arrived at a matrix-based notation, it will be clear that only a few fundamental types of computational kernels are needed to perform the calculation, so that parallelization and optimization need only address themselves to these few kernels. This provides a great boon for portability, ease of programming, and extensibility to new physical scenarios.

In the discussion below, we describe our formalism only for the case of local ionic potentials. The use of non-local potentials (e.g. for the important case of pseudopotential calculations) results in only minor changes that are addressed in Appendix B. Furthermore, for periodic systems, we will be working with only a single

k-point at $k = 0$, as is evident from the choice of expansion in Eq. (5.7). We work at $k = 0$ in order to keep the mathematical expressions as transparent as possible. The minor extensions required to accommodate non-zero and multiple k-points are straightforward and are dealt with in Appendix C.

5.3.1 Basis-dependent operators

In this section we describe all the operators in our formalism that depend on the basis set chosen for the calculation. We will see that there are a small number of such operations, and that we can easily separate their role from the rest of the formalism.

The first two operators involve matrix elements of the identity and the Laplacian between pairs of basis functions. Specifically, we define

$$\mathcal{O}_{\alpha,\beta} \equiv \int d^3r \; b^*_\alpha(r)\, b_\beta(r)\,, \tag{5.8}$$

$$L_{\alpha,\beta} \equiv \int d^3r \; b^*_\alpha(r)\, \nabla^2\, b_\beta(r)\,. \tag{5.9}$$

We call these operators the overlap and Laplacian respectively. Note that for orthonormal bases, we have $\mathcal{O} = I$ where I is the identity matrix.

The integrals of the basis functions are the components of the column vector s,

$$s_\alpha \equiv \int d^3r \; b^*_\alpha(r)\,. \tag{5.10}$$

For periodic systems, we use the vector s to define a new operator $\bar{\mathcal{O}}$ through

$$\bar{\mathcal{O}} \equiv \mathcal{O} - \frac{ss^\dagger}{\Omega}\,, \tag{5.11}$$

where Ω is the volume of the periodic supercell. For calculations in systems without boundaries, the volume Ω is infinite so that $\mathcal{O} = \bar{\mathcal{O}}$. The chief use of $\bar{\mathcal{O}}$ is for solving the Poisson equation in periodic systems where divergences due to the long-range Coulomb interaction must be avoided. The automatic avoidance of such divergences and the proper choice of n_0 in Eq. (5.1) both follow directly from the nature of the

Lagrangian as will be discussed in Section 5.3.3.

The next four operators involve the values of the basis functions on the points p of the real-space grid introduced above. The *forward transform* operator $\mathcal{I}$ allows for changing representation from the space of expansion coefficients to the space of function values on the real-space grid. Specifically, given a basis function α and a grid point p, we define

$$\mathcal{I}_{p\alpha} = b_\alpha(p)\,. \tag{5.12}$$

Thus the αth column of $\mathcal{I}$ consists of the values of the αth basis function on the points of the real-space grid.

Next, it is at times necessary to find the expansion coefficients for a function given its values on the real-space grid. We denote this linear *inverse transform* by $\mathcal{J}$. In implementations where the number of grid points p is equal to the number of basis functions, the natural choice is to take $\mathcal{J} \equiv \mathcal{I}^{-1}$ (e.g., plane-wave basis sets). However, this is not necessary: in some applications, one may choose to use a very dense real-space grid which has more points than the number of basis functions. Hence, we keep the formal distinction between $\mathcal{J}$ and $\mathcal{I}^{-1}$. We will also require two *conjugate* transforms, which are the Hermitian conjugates $\mathcal{I}^\dagger$ and $\mathcal{J}^\dagger$.

The final mathematical object that depends on the basis set involves the ionic potential $V_{ion}(r)$. We define a column vector V_{ion} whose components are the integrals

$$(V_{ion})_\alpha \equiv \int d^3r\; b^*_\alpha(r)\, V_{ion}(r)\,, \tag{5.13}$$

which encodes overlaps of the ionic potential with the basis functions. We will use V_{ion} when evaluating the electron-ion interaction energy in Section 5.3.3.

5.3.2 Identities satisfied by the basis-dependent operators

Although the operators $\mathcal{O}$, $\bar{\mathcal{O}}$, L, $\mathcal{I}$, $\mathcal{J}$, $\mathcal{I}^\dagger$, and $\mathcal{J}^\dagger$ depend on the choice of basis, they satisfy various identities which will prove important below. In addition to their formal properties, these identities allow for verification of the implementation of these

operators.

The most important identity involves the constant function. To represent the constant function on the grid, we define the column vector $\mathbf{1}$ as having the value of unity on each grid point p: $\mathbf{1}_p = 1$. Many basis sets can represent this function exactly (e.g. plane waves or finite-element sets). For such bases, for all points r in the simulation cell, we must have the identity

$$\sum_\alpha (\mathcal{J}\mathbf{1})_\alpha \, b_\alpha(r) = 1 \, . \tag{5.14}$$

Evaluating this identity on the real-space grid yields

$$\mathcal{I}\mathcal{J}\mathbf{1} = \mathbf{1} \, . \tag{5.15}$$

For basis sets that can not represent the constant function exactly, the identity of Eq. (5.15) and the ones below should hold approximately in the regions described by the basis.

According to Eq. (5.14), the vector $\mathcal{J}\mathbf{1}$ specifies the coefficients of the expansion of the constant function. Using the integrals s of Eq. (5.10), we can see that

$$\begin{aligned} s_\alpha &= \int d^3r \; b_\alpha^*(r) \\ &= \int d^3r \; b_\alpha^*(r) \left(\sum_\beta (\mathcal{J}\mathbf{1})_\beta b_\beta(r) \right) \\ &= (\mathcal{O}\mathcal{J}\mathbf{1})_\alpha \, . \end{aligned}$$

Thus we have that $s = \mathcal{O}\mathcal{J}\mathbf{1}$. We can also derive the normalization condition

$$s^\dagger \mathcal{J}\mathbf{1} = \int d^3r \; \sum_\alpha b_\alpha(r) (\mathcal{J}\mathbf{1})_\alpha = \int d^3r \; 1 = \Omega \, , \tag{5.16}$$

where Ω is the volume in which the calculation is performed.

When solving Poisson's equation for the electrostatic potential (Eq. (5.5)), we must take special care regarding the null space of the Laplacian operator L, which is

the space of constant functions. Integrating the identity $\nabla^2 1 = 0$ against the complex conjugate of each basis function yields

$$L\mathcal{J}\mathbf{1} = 0\,. \tag{5.17}$$

We use this identity when dealing with the Poisson equation in periodic systems to avoid divergences due to the long-range nature of the Coulomb interaction.

5.3.3 Basis-independent expression for the Lagrangian

We now use the above operators to express the Lagrangian of Eq. (5.1) in a matrix-based, basis-independent manner. We begin by introducing two operators, "diag" and "Diag". The operator diag converts a square matrix M into a column vector containing the diagonal elements of the matrix. The operator Diag converts a vector v into a diagonal matrix with the components of v on its diagonal. In terms of components, we have that

$$(\text{diag } M)_\alpha = M_{\alpha\alpha} \quad , \quad (\text{Diag } v)_{\alpha\beta} = v_\alpha \delta_{\alpha\beta}\,, \tag{5.18}$$

where δ is the Kronecker delta. Thus, diag Diag $v = v$ for any vector v whereas Diag diag $M = M$ if and only if M is a diagonal matrix. Two useful identities involving these operators are

$$(\text{diag } M)^\dagger\, v = \text{Tr}(M^\dagger\, \text{Diag } v) \quad , \quad v^\dagger\, (\text{diag } M) = \text{Tr}(\,(\text{Diag } v)^\dagger\, M)\,, \tag{5.19}$$

where † indicates Hermitian or complex-conjugated transposition.

Next, if we regard the expansion coefficients $C_{\alpha i}$ as a matrix whose ith column contains the expansion coefficients of the ith wave function (Eq. (5.7)), and we also define the diagonal matrix of Fermi fillings $F_{ij} = f_i \delta_{ij}$, it is easy to see that

$$P = CFC^\dagger \tag{5.20}$$

is the representation of the single-particle density matrix in the space of basis functions.

Before considering the Lagrangian itself, we will also need expressions for the electron density $n(r)$ which appears in most of the terms of the Lagrangian of Eq. (5.1). We define a vector n whose components are the values of the electron density on the points p of the real-space grid. Specifically,

$$\begin{aligned} n_p \equiv n(p) &= \sum_i f_i \|\psi_i(p)\|^2 = \sum_i f_i \| \left(\mathcal{I}C\right)_{pi} \|^2 \\ &= \sum_i \left(\mathcal{I}C\right)^*_{pi} f_i \left(\mathcal{I}C\right)_{pi} = \left((\mathcal{I}C) F (\mathcal{I}C)^\dagger \right)_{pp} , \end{aligned}$$

whence we arrive at the identity defining the vector n

$$n = \mathrm{diag}\left(\mathcal{I}P\mathcal{I}^\dagger\right) . \tag{5.21}$$

Given the values of the electron density on the real-space grid, we use the inverse transform $\mathcal{J}$ to find the expansion coefficients of $n(r)$ in terms of the basis functions. This vector of coefficients is just $\mathcal{J}n$.

Armed with these few tools, we now proceed to write the various energetic terms of the Lagrangian in the matrix language developed above.

Kinetic energy

The kinetic energy T can be transformed into the matrix language by using the expansion coefficients C of Eq. (5.7) and the definition of the Laplacian L of Eq. (5.9):

$$\begin{aligned} T &\equiv -\frac{1}{2} \sum_i f_i \int d^3r \; \psi_i^*(r) \nabla^2 \psi_i(r) = -\frac{1}{2} \sum_i f_i \sum_{\alpha,\beta} C^*_{\alpha i} L_{\alpha\beta} C_{\beta i} \\ &= -\frac{1}{2} \mathrm{Tr}\left(F C^\dagger L C\right) = -\frac{1}{2} \mathrm{Tr}\left(LP\right) , \end{aligned} \tag{5.22}$$

where the last two equivalent expressions are related by the cyclic property of the trace. Thus, we are able to write the kinetic energy explicitly as a function of the density matrix P of Eq. (5.20).

Electron-ion interaction

Since the electron density $n(r)$ is real, we may write the electron-ion interaction as

$$\begin{aligned} E_{e-i} &\equiv \int d^3r\ n^*(r)\, V_{ion}(r) = \sum_\alpha \int d^3r (\mathcal{J}n)^*_\alpha b^*_\alpha(r)\, V_{ion}(r) \\ &= (\mathcal{J}n)^\dagger V_{ion} = \mathrm{Tr}\left(\mathcal{I}^\dagger \left[\mathrm{Diag}\ \mathcal{J}^\dagger V_{ion}\right] \mathcal{I}P\right), \end{aligned} \tag{5.23}$$

where we have used the definition of V_{ion} from Eq. (5.13) and have used Eqs. (5.21) and (5.19) to rewrite this interaction in terms of P.

Exchange-correlation energy

Given the vector n of electron-density values on the grid, we can evaluate the exchange-correlation energy per particle at each grid point p through $\epsilon_{xc}(n(p))$. We collect these values into a vector $\epsilon_{xc}(n)$. We then inverse transform this vector and the electron density vector, and we use the overlap operator to arrive at

$$\begin{aligned} E_{xc} &\equiv \int d^3r\ n^*(r)\, \epsilon_{xc}(n(r)) \\ &= (\mathcal{J}n)^\dagger\, \mathcal{O}\, (\mathcal{J}\epsilon_{xc}(n)) = \mathrm{Tr}\left(\mathcal{I}^\dagger \left[\mathrm{Diag}\ \mathcal{J}^\dagger \mathcal{O} \mathcal{J} \epsilon_{xc}(n)\right] \mathcal{I}P\right), \end{aligned} \tag{5.24}$$

where we again have conjugated the electron density for ease of formal manipulations. The derivation of the final expression in terms of P uses Eq. (5.21).

Hartree self-energy

The self-energy of the Hartree field can be written as

$$E_{H-H} \equiv -\frac{1}{8\pi} \int d^3r\ \|\vec{\nabla}\phi(r)\|^2 = \frac{1}{8\pi} \int d^3r\ \phi^*(r) \nabla^2 \phi(r) = \frac{1}{8\pi} \phi^\dagger L \phi\,, \tag{5.25}$$

where we have first integrated by parts and then substituted the expansion coefficients ϕ of Eq. (5.7). The complex conjugation of the real-valued function $\phi(r)$ allows for the simplicity of the final expression.

Electron-Hartree interaction

The interaction of the electron density $n(r)$ and Hartree potential $\phi(r)$ can be written as

$$E_{e-H} \equiv -\int d^3r \;\; (n(r) - n_0)^* \, \phi(r) = -\left[\mathcal{J}(n - n_0\mathbf{1})\right]^\dagger \mathcal{O}\phi \,. \tag{5.26}$$

The proper choice of n_0 for periodic systems can be found by noting that the Hartree self-energy E_{H-H} of Eq. (5.25) has no dependence on the projection of ϕ onto the null space of L which, as we saw in Section 5.3.2, lies along the vector $\mathcal{J}\mathbf{1}$. Thus, for the Lagrangian of Eq. (5.1) to have a saddle-point, there can be no coupling of $n(r) - n_0$ with the projection of ϕ along $\mathcal{J}\mathbf{1}$. That is, we must have $\left[\mathcal{J}(n - n_0\mathbf{1})\right]^\dagger \mathcal{O} \cdot \mathcal{J}\mathbf{1} = 0$. The identities of Section 5.3.2 then lead to the choice $n_0 = (\mathcal{J}n)^\dagger s/\Omega$. Our final expression for E_{e-H} is thus given by

$$E_{e-H} = -(\mathcal{J}n)^\dagger \left(\mathcal{O} - \frac{ss^\dagger}{\Omega}\right)\phi = -(\mathcal{J}n)^\dagger \bar{\mathcal{O}}\phi = -\mathrm{Tr}\left(\mathcal{I}^\dagger \left[\mathrm{Diag}\; \mathcal{J}^\dagger \bar{\mathcal{O}}\phi\right] \mathcal{I}P\right) \,. \tag{5.27}$$

Complete Lagrangian

Summing all the contributions above, we arrive at two equivalent expressions for the Lagrangian $\mathcal{L}_{LDA}$,

$$\begin{aligned} \mathcal{L}_{LDA} &= -\frac{1}{2}\mathrm{Tr}\left(FC^\dagger LC\right) + (\mathcal{J}n)^\dagger \left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n) - \bar{\mathcal{O}}\phi\right] + \frac{\phi^\dagger L\phi}{8\pi} & (5.28) \\ &= -\frac{1}{2}\mathrm{Tr}\left(LP\right) + \frac{1}{8\pi}\phi^\dagger L\phi \\ &\quad + \mathrm{Tr}\left(\mathcal{I}^\dagger \mathrm{Diag}\left[\mathcal{J}^\dagger V_{ion} + \mathcal{J}^\dagger \mathcal{O}\mathcal{J}\epsilon_{xc}(n) - \mathcal{J}^\dagger \bar{\mathcal{O}}\phi\right] \mathcal{I}P\right) \,. & (5.29) \end{aligned}$$

The first, compact form is computationally efficient for evaluating the Lagrangian as a function of C and ϕ. The second form, written as a function of the density matrix P, finds its best use in the formal manipulations required to find the gradient of the Lagrangian.

5.3.4 Orthonormality constraints

The orthonormality constraints of Eq. (5.3) are equivalent to the matrix equation

$$C^\dagger \mathcal{O} C = I\,. \tag{5.30}$$

If we wish to compute gradients of the Lagrangian with respect to C in order to arrive at the Kohn-Sham equations, we must do so while always obeying these constraints.

The analytically-continued functional approach [4] deals with these constraints by introducing a set of expansion coefficients Y which are *unconstrained* and which can have any overlap U,

$$U = Y^\dagger \mathcal{O} Y. \tag{5.31}$$

We also allow for the possibility of subspace rotation, which is a unitary transformation mapping the subspace of occupied states $\{\psi_i\}$ onto itself. Such a transformation is affected by a unitary matrix V, and we parameterize V as the exponential of a Hermitian matrix B through

$$V \equiv e^{iB} \quad \text{where} \quad B^\dagger = B\,. \tag{5.32}$$

The coefficients C are defined as dependent variables through the mapping

$$C = Y U^{-1/2} V^\dagger, \tag{5.33}$$

which ensures that Eq. (5.30) is automatically obeyed, as is easy to verify by direct substitution. The density matrix P takes the following form in terms of Y and V,

$$P = CFC^\dagger = YU^{-1/2}V^\dagger F V U^{-1/2} Y^\dagger\,. \tag{5.34}$$

In most cases, we simply set $V = I$. In fact, for the case of constant fillings, $F = fI$, the unitary matrix V drops out of P completely. The subspace rotations find their primary use in the study of metallic or high-temperature systems where the

Fermi-Dirac fillings are not constant, and the rotations allow for greatly improved convergence rates when searching for the saddle point of the Lagrangian. This point is explained in more detail in Section 5.5.

5.3.5 Derivatives of the Lagrangian

Since the most effective modern methods that search for stationary points require knowledge of the derivative of the objective function, we will now find the derivative of the Lagrangian of Eq. (5.28) or (5.29) with respect to the variables Y and ϕ (and B if subspace rotations are used). Differentiation with respect to Y is far more complex due to the orthonormality constraints, and we begin with this immediately.

Derivative with respect to the electronic states

Computing the derivative of the Lagrangian with respect to Y is intricate, and we break the problem into smaller pieces by first finding the derivative with respect to the density matrix P. Once the derivative with respect to P is found, we can use the relation between P and Y (Eq. (5.34)) to find the derivative with respect to Y.

We begin by noting that except for the exchange-correlation energy, the entire expression of Eq. (5.29) is linear in the density matrix P. The exchange-correlation energy is a function of the electron density n, which, through Eq. (5.21), is also a function of P. Thus if we consider a differential change dP of the density matrix, the only term in $d\mathcal{L}_{LDA}$ that needs to be considered carefully is

$$\begin{aligned} n^\dagger \mathcal{J}^\dagger \mathcal{O} \mathcal{J} d[\epsilon_{xc}(n)] &= n^\dagger \mathcal{J}^\dagger \mathcal{O} \mathcal{J} \left[\text{Diag } \epsilon'_{xc}(n)\right] dn \\ &= \text{Tr}\left\{ \mathcal{I}^\dagger \text{Diag}\left(\left[\text{Diag } \epsilon'_{xc}(n)\right] \mathcal{J}^\dagger \mathcal{O} \mathcal{J} n \right) \mathcal{I}\, dP \right\}. \end{aligned}$$

In the above derivation, we have used Eq. (5.21) to relate dn to dP as well as the identities of Eq. (5.19). The vector $\epsilon'_{xc}(n)$ is given by its values on the real-space grid point p via $(\epsilon'_{xc}(n))_p \equiv \epsilon'_{xc}(n(p))$.

We can now write the differential of the Lagrangian of Eq. (5.29) with respect to

P as

$$\begin{aligned} d\mathcal{L}_{LDA} &= \mathrm{Tr}\left\{-\frac{1}{2}L\,dP + \mathcal{I}^\dagger \mathrm{Diag}\left[\mathcal{J}^\dagger V_{ion} + \mathcal{J}^\dagger \mathcal{O}\mathcal{J}\epsilon_{xc}(n)\right.\right. \\ &\qquad \left.\left. +\ [\mathrm{Diag}\ \epsilon'_{xc}(n)]\,\mathcal{J}^\dagger\mathcal{O}\mathcal{J}n - \mathcal{J}^\dagger\bar{\mathcal{O}}\phi\right]\mathcal{I}\,dP\right\} \\ &\equiv \mathrm{Tr}\,(H\,dP)\ , \end{aligned} \tag{5.35}$$

where the single-particle Kohn-Sham Hamiltonian operator H is given by

$$\begin{aligned} H &= -\frac{1}{2}L + \mathcal{I}^\dagger[\mathrm{Diag}\ V_{sp}]\mathcal{I}\,, \quad \text{where} \\ V_{sp} &= \mathcal{J}^\dagger V_{ion} + \mathcal{J}^\dagger\mathcal{O}\mathcal{J}\epsilon_{xc}(n) + [\mathrm{Diag}\ \epsilon'_{xc}(n)]\mathcal{J}^\dagger\mathcal{O}\mathcal{J}n - \mathcal{J}^\dagger\bar{\mathcal{O}}\phi\,. \end{aligned} \tag{5.36}$$

The single-particle Hamiltonian is the sum of a kinetic operator and a local single-particle potential V_{sp} (a vector of numbers on the real-space grid specifying the values of the potential).

Eq. (5.35) has conveniently separated out the physical description of the system, the Hamiltonian H, from the variation dP which we now compute. Differentiating the relation $U^{-1/2}U^{1/2} = I$, we find that

$$d[U^{-1/2}] = -U^{-1/2}d[U^{1/2}]U^{-1/2},$$

and we use this to express the variation of the density matrix of Eq. (5.34) as

$$\begin{aligned} dP &= (dY)U^{-1/2}V^\dagger FVU^{-1/2}Y^\dagger + YU^{-1/2}V^\dagger FVU^{-1/2}(dY^\dagger) \\ &\quad - YU^{-1/2}\left(d[U^{1/2}]U^{-1/2}V^\dagger FV + V^\dagger FVU^{-1/2}d[U^{1/2}]\right)U^{-1/2}Y^\dagger. \end{aligned}$$

We now substitute this expression for dP into Eq. (5.35). We use the definition of the operator Q (Eq. (E.6) of Appendix E), its relation to $d[U^{1/2}]$ (Eq. (E.5)), and the identities which Q satisfies (Eqs. (E.7)). After some manipulations involving the

cyclicity of the trace, we arrive at

$$\begin{aligned} d\mathcal{L}_{LDA} &= \mathrm{Tr}\left[dY^\dagger \left(\frac{\partial \mathcal{L}_{LDA}}{\partial Y^\dagger}\right) + \left(\frac{\partial \mathcal{L}_{LDA}}{\partial Y^\dagger}\right)^\dagger dY\right], \text{ where} \\ \left(\frac{\partial \mathcal{L}_{LDA}}{\partial Y^\dagger}\right) &\equiv \left(I - \mathcal{O}CC^\dagger\right) HCFVU^{-1/2} + \mathcal{O}CVQ\left(V^\dagger[\tilde{H}, F]V\right), \text{ and} \\ \tilde{H} &\equiv C^\dagger HC, \end{aligned} \tag{5.37}$$

where $\tilde{H}$ is the subspace Hamiltonian and contains matrix elements of the Hamiltonian H among the wave functions $\{\psi_i(r)\}$. Square brackets denote the commutator, $[a, b] \equiv ab - ba$. Physical interpretation of the terms in Eq. (5.37) is provided in Section 5.3.6.

Finally, since Y and $Y^\dagger$ are not independent, we can simplify the expression for the differential of $\mathcal{L}_{LDA}$ to

$$d\mathcal{L}_{LDA} = 2 \text{ Re Tr}\left[dY^\dagger \left(\frac{\partial \mathcal{L}_{LDA}}{\partial Y^\dagger}\right)\right],$$

where Re denotes the real part of its argument.

Derivative with respect to the Hartree field

Since the Lagrangian in Eq. (5.28) is quadratic in ϕ, its derivative with respect to ϕ may be readily calculated. However, to arrive at a symmetric expression for the derivative in terms of ϕ and $\phi^\dagger$, we note that the linear dependence on ϕ, given by $z \equiv (\mathcal{J}n)^\dagger \bar{\mathcal{O}}\phi$, is a real number because both $n(r)$ and $\phi(r)$ are real in Eq. (5.26). For convenience, we rewrite this as $(z+z^*)/2$, which is an equivalent expression symmetric in ϕ and $\phi^\dagger$. By using this, we compute the variation of $\mathcal{L}_{LDA}$ and arrive at

$$\begin{aligned} d\mathcal{L}_{LDA} &= d\phi^\dagger \left(\frac{\partial \mathcal{L}_{LDA}}{\partial \phi^\dagger}\right) + \left(\frac{\partial \mathcal{L}_{LDA}}{\partial \phi^\dagger}\right)^\dagger d\phi, \text{ where} \\ \left(\frac{\partial \mathcal{L}_{LDA}}{\partial \phi^\dagger}\right) &\equiv -\frac{1}{2}\bar{\mathcal{O}}\mathcal{J}n + \frac{1}{8\pi}L\phi. \end{aligned} \tag{5.38}$$

Again, since ϕ and $\phi^\dagger$ are not independent, we can express the variation as

$$d\mathcal{L}_{LDA} = 2\,\mathrm{Re}\left[d\phi^\dagger \left(\frac{\partial \mathcal{L}_{LDA}}{\partial \phi^\dagger}\right)\right] .$$

Derivative with respect to subspace rotations

We have parameterized the unitary matrix V of Eq. (5.33) by the Hermitian matrix B of Eq. (5.32) as $V = e^{iB}$. We will now find the derivative of $\mathcal{L}_{LDA}$ with respect to B. First, we consider the variation of $\mathcal{L}_{LDA}$ with respect to those of V and $V^\dagger$ by using Eq. (5.35) as our starting point. Using the definition of P in Eq. (5.34), we have that

$$\begin{aligned} d\mathcal{L}_{LDA} &= \mathrm{Tr}\{H\,dP\} \\ &= \mathrm{Tr}\left\{HYU^{-1/2}[dV^\dagger FV + V^\dagger F dV]U^{-1/2}Y^\dagger\right\} \\ &= \mathrm{Tr}\left\{\tilde{H}'[dV^\dagger FV + V^\dagger F dV]\right\} , \end{aligned}$$

where $\tilde{H}' = U^{-1/2}Y^\dagger HYU^{-1/2}$. Differentiating the identity $V^\dagger V = I$ leads to $dV^\dagger = -V^\dagger dV V^\dagger$ which allows us to write

$$d\mathcal{L}_{LDA} = \mathrm{Tr}\left\{[\tilde{H}, F]dVV^\dagger\right\} ,$$

where again $\tilde{H} = C^\dagger HC$ is the subspace Hamiltonian.

We place the eigenvalues of B on the diagonal of a diagonal matrix β and place the eigenvectors of B in the columns of a unitary matrix Z. Thus $B = Z\beta Z^\dagger$ and $Z^\dagger Z = ZZ^\dagger = I$. We now use the result of Eq. (E.4) of Appendix E applied to the case $V = f(B) = e^{iB}$ to arrive at the following result relating dV to dB:

$$(Z^\dagger dV Z)_{nm} = (Z^\dagger dB Z)_{nm} \cdot \begin{cases} ie^{i\beta_n} & \text{if } m = n \\ \left[\frac{e^{i\beta_m} - e^{i\beta_n}}{\beta_m - \beta_n}\right] & \text{if } m \neq n \end{cases} .$$

Using this and the fact that $V^\dagger = Ze^{-i\beta}Z^\dagger$, we have that

$$\begin{aligned} d\mathcal{L}_{LDA} &= \mathrm{Tr}\left\{[\tilde{H},F]Z(Z^\dagger dVZ)Z^\dagger V^\dagger\right\} \\ &= \mathrm{Tr}\left\{Z^\dagger[\tilde{H},F]Z(Z^\dagger dVZ)e^{-i\beta}\right\} \\ &= \sum_{n,m}(Z^\dagger[\tilde{H},F]Z)_{nm}(Z^\dagger dBZ)_{mn}\cdot\begin{cases} i & \text{if } m=n \\ \left[\frac{e^{i\beta_m - i\beta_n}-1}{\beta_m-\beta_n}\right] & \text{if } m\neq n \end{cases}. \end{aligned}$$

We define the operator $R(A)$ acting on a general matrix A via

$$(Z^\dagger R(A)Z)_{nm} \equiv (Z^\dagger AZ)_{nm}\cdot\begin{cases} i & \text{if } m=n \\ \left[\frac{e^{i\beta_m - i\beta_n}-1}{\beta_m-\beta_n}\right] & \text{if } m\neq n \end{cases}.$$

This allows us to write the variation of $\mathcal{L}_{LDA}$ as

$$d\mathcal{L}_{LDA} = \mathrm{Tr}\left\{Z^\dagger R\left([\tilde{H},F]\right)ZZ^\dagger dBZ\right\} = \mathrm{Tr}\left\{R\left([\tilde{H},F]\right)dB\right\},$$

so that the derivative of $\mathcal{L}_{LDA}$ with respect to B is

$$\frac{\partial\mathcal{L}_{LDA}}{\partial B} = R\left([\tilde{H},F]\right). \tag{5.39}$$

5.3.6 Kohn-Sham and Poisson equations

The Kohn-Sham and Poisson equations (Eqs. (5.4) and (5.5)) are obtained by setting the derivative of the Lagrangian with respect to Y and ϕ to zero. This results in the two equations

$$\left(\frac{\partial\mathcal{L}_{LDA}}{\partial Y^\dagger}\right) = 0 = \left(I-\mathcal{O}CC^\dagger\right)HCFVU^{-1/2} + \mathcal{O}CVQ(V^\dagger[\tilde{H},F]V), \tag{5.40}$$

$$\left(\frac{\partial\mathcal{L}_{LDA}}{\partial\phi^\dagger}\right) = 0 = -\frac{1}{2}\bar{\mathcal{O}}\mathcal{J}n + \frac{1}{8\pi}L\phi. \tag{5.41}$$

Eq. (5.40) states the stationarity of the Lagrangian with respect to variations of the wave-function coefficients Y, and we examine it first.

We define the projection operator $\rho = \mathcal{O}CC^\dagger$ which satisfies $\rho^2 = \rho$ and which

projects onto the subspace of occupied states $\{\psi_i\}$ used in the calculation. Its complement $\bar{\rho} = I - \rho$ projects onto the orthogonal subspace spanned by the unoccupied states. By multiplying Eq. (5.40) on the left by $\bar{\rho}$ and assuming that none of the Fermi fillings are zero, we find that

$$\bar{\rho} H C = 0 \,.$$

This reproduces the well known condition that at the stationary point, the Hamiltonian must map the occupied subspace onto itself.

Conversely, we can project Eq. (5.40) onto the occupied subspace by multiplying on the left by $C^\dagger$. This, combined with the fact that Q is an invertible linear operator, leads to the condition

$$[\tilde{H}, F] = 0 \,.$$

Given that F is a diagonal matrix, for arbitrary fillings, the subspace Hamiltonian $\tilde{H}$ also must be diagonal: the states $\{\psi_i\}$ must be eigenstates of H with eigenvalues ϵ_i, as we have explicitly written in Eq. (5.4). However, if a pair of states ψ_i and ψ_j have degenerate fillings, $f_i = f_j$, then $\tilde{H}_{ij}$ need not be zero. Converting such degenerate cases into the conventional diagonal representation requires a further unitary rotation, which, however, is not required for stationarity.

The second condition for stationarity, Eq. (5.41), rearranges into

$$L\phi = 4\pi \bar{\mathcal{O}} \mathcal{J} n \,. \tag{5.42}$$

We have arrived at the Poisson equation for the Hartree potential ϕ generated by the electron density n (the negative electronic charge is reflected by the positive coefficient of the right-hand side). Since we have explicitly projected out the null-space of L from the right-hand side, we may invert L and find the solution to the Poisson equation

$$\phi = 4\pi L^{-1} \bar{\mathcal{O}} \mathcal{J} n \,. \tag{5.43}$$

Finally, if we substitute the result of Eq. (5.43) for ϕ into our Lagrangian, we find the LDA energy functional (cf. Eq. (5.6)):

$$E_{LDA} = -\frac{1}{2}\mathrm{Tr}\left(FC^\dagger LC\right) + (\mathcal{J}n)^\dagger \left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n) - \frac{1}{2}\bar{\mathcal{O}}\left(4\pi L^{-1}\bar{\mathcal{O}}\mathcal{J}n\right)\right]. \quad (5.44)$$

5.3.7 Expressions for Lagrangian and derivatives: summary

We now collect the expressions for the LDA Lagrangian and its derivatives in one place. As we will see in Section 5.6, formulae in the DFT++ notation translate directly into lines of computer code, so that we will also be specifying the computational implementation of the Lagrangian. In addition, given the Lagrangian and its derivatives, we can apply any suitable algorithm to find the stationary point (Section 5.5).

The key expressions are

$$\begin{aligned}
\mathcal{L}_{LDA}(Y,\phi,B) &= -\frac{1}{2}\mathrm{Tr}\left(FC^\dagger LC\right) + (\mathcal{J}n)^\dagger\left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n) - \bar{\mathcal{O}}\phi\right] + \frac{1}{8\pi}\phi^\dagger L\phi\,, \\
\frac{\partial\mathcal{L}_{LDA}}{\partial Y^\dagger} &= \left(I - \mathcal{O}CC^\dagger\right)HCFVU^{-1/2} + \mathcal{O}CVQ\left(V^\dagger[\tilde{H},F]V\right)\,, \\
\frac{\partial\mathcal{L}_{LDA}}{\partial\phi^\dagger} &= -\frac{1}{2}\bar{\mathcal{O}}\mathcal{J}n + \frac{1}{8\pi}L\phi \quad , \quad \frac{\partial\mathcal{L}_{LDA}}{\partial B} = R\left([\tilde{H},F]\right)\,, \\
H &= -\frac{1}{2}L + \mathcal{I}^\dagger\left[\mathrm{Diag}\;V_{sp}\right]\mathcal{I} \quad , \quad \tilde{H} = C^\dagger HC\,.
\end{aligned}$$

As discussed in Section 5.2, the value of $\mathcal{L}_{LDA}$ and its Y and B derivatives are equal to the value and respective derivatives of the energy E_{LDA} of Eq. (5.44) when we evaluate the Lagrangian-based quantities at the solution of the Poisson equation. Therefore, the above expressions can also be used to find the derivatives of E_{LDA}, a fact that we will exploit in Section 5.5.

5.4 DFT++ specification for various *ab initio* techniques

In the previous section, we presented a detailed derivation of the expression for the LDA Lagrangian and its derivatives in the DFT++ formalism. We believe that the community of physicists and chemists using this and other general-functional methods should use this formalism for the communication of new energy functionals and comparisons among them.

From a physicist's or chemist's viewpoint, which we adopt in this section, linear algebra and matrices are the settings in which quantum mechanical computations must be performed. Therefore, they are the fundamental objects in the new formalism. This is in contrast with the Dirac notation, which while an excellent conceptual tool for studying quantum problems, can never be used to carry out an actual calculation: matrix elements of bras and kets must first be found before an actual computation can proceed.

Once an expression for an energy functional is found, its derivative is found by straightforwardly differentiating it with respect to the matrices of independent variables. Armed with expressions for the functional and its derivative, the methods discussed in Section 5.5 can then be applied to achieve self-consistency.

In this spirit, we now present a few examples of energy functionals. Some are extensions of the LDA, while others are similar to the LDA only in that they involve the study of single-particle systems. In all cases, our aim will be to show how quickly and easily we can find the requisite expressions for the appropriate functional and its derivative.

5.4.1 Local spin-density approximation (LSDA)

The most straightforward generalization of the LDA approximation is to allow for spin-up and spin-down electrons to have different wave functions but to still treat exchange-correlation energies in a local approximation. Specifically, the exchange-

correlation energy per particle at position r is now a function of both the spin-up and spin-down electron densities, $n_\uparrow(r)$ and $n_\downarrow(r)$, and the total LSDA exchange-correlation energy is given by

$$E_{xc} = \int d^3r \; n(r)\, \epsilon_{xc}(n_\uparrow(r), n_\downarrow(r)) \; ,$$

where $n(r) = n_\uparrow(r) + n_\downarrow(r)$ is the total electron density. The LSDA has been found to show substantial improvements over the LDA for atomic and molecular properties [37, 76] since the spin of the electrons is explicitly dealt with.

The changes required in the expressions of the Lagrangian and its derivatives in order to incorporate the LSDA are straightforward and easy to implement. We label spin states by σ, which can take the value $\uparrow$ or $\downarrow$. We have spin-dependent expansion coefficient matrices C_σ for the wave functions (cf. Eq. (5.7)). Each spin channel has its own fillings F_σ and density matrix P_σ,

$$P_\sigma = C_\sigma F_\sigma C_\sigma^\dagger \, .$$

The electron densities n_σ and the total electron density n are given by (cf. Eq. (5.21))

$$n_\sigma = \mathrm{diag}(\mathcal{I} P_\sigma \mathcal{I}^\dagger) \quad \text{and} \quad n = \sum_\sigma n_\sigma \; .$$

The LSDA Lagrangian is given by

$$\mathcal{L}_{LSDA}(C_\uparrow, C_\downarrow, \phi) = -\frac{1}{2} \sum_\sigma \mathrm{Tr}\left(F_\sigma C_\sigma^\dagger L C_\sigma\right) + (\mathcal{J} n)^\dagger \left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n_\uparrow, n_\downarrow) - \bar{\mathcal{O}}\phi\right] + \frac{1}{8\pi}\phi^\dagger L \phi .$$

The orthonormal expansion coefficients C_σ are found from unconstrained variables Y_σ via

$$C_\sigma = Y_\sigma U_\sigma^{-1/2}, \text{ where } U_\sigma = Y_\sigma^\dagger \mathcal{O} Y_\sigma \, ,$$

where, for simplicity, we have set subspace rotations to unity, $V_\sigma = I$. Finding the derivatives of the Lagrangian with respect to the coefficients Y_σ follows the analysis

of Section 5.3.5. Each spin channel has a single-particle Hamiltonian H_σ given by

$$\begin{aligned} H_\sigma &= -\frac{1}{2}L + \mathcal{I}^\dagger[\text{Diag } V_\sigma]\mathcal{I} \ , \text{ where} \\ V_\sigma &= \mathcal{J}^\dagger V_{ion} + \mathcal{J}^\dagger \mathcal{O}\mathcal{J}\epsilon_{xc}(n_\uparrow, n_\downarrow) + \text{Diag}\left[\frac{\partial \epsilon_{xc}(n_\uparrow, n_\downarrow)}{\partial n_\sigma}\right]\mathcal{J}^\dagger\mathcal{O}\mathcal{J}n - \mathcal{J}^\dagger\bar{\mathcal{O}}\phi. \end{aligned}$$

The derivative of the Lagrangian with respect to Y_σ (cf. Eq. (5.37)) is given by

$$\frac{\partial \mathcal{L}_{LSDA}}{\partial Y_\sigma^\dagger} = \left(I - \mathcal{O}C_\sigma C_\sigma^\dagger\right) H_\sigma C_\sigma F_\sigma U_\sigma^{-1/2} + \mathcal{O}C_\sigma Q([\tilde{H}_\sigma, F_\sigma]), \text{ where } \tilde{H}_\sigma \equiv C_\sigma^\dagger H_\sigma C_\sigma.$$

In summary, we have the following expressions for the LSDA Lagrangian and derivatives

$$\begin{aligned} \mathcal{L}_{LSDA}(Y_\uparrow, Y_\downarrow, \phi) &= -\frac{1}{2}\sum_\sigma \text{Tr}\left(F_\sigma C_\sigma^\dagger L C_\sigma\right) + \frac{1}{8\pi}\phi^\dagger L\phi \\ &\quad + (\mathcal{J}n)^\dagger\left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n_\uparrow, n_\downarrow) - \bar{\mathcal{O}}\phi\right] \\ \frac{\partial \mathcal{L}_{LSDA}}{\partial Y_\sigma^\dagger} &= \left(I - \mathcal{O}C_\sigma C_\sigma^\dagger\right) H_\sigma C_\sigma F_\sigma U_\sigma^{-1/2} + \mathcal{O}C_\sigma Q([\tilde{H}_\sigma, F_\sigma]) \\ \frac{\partial \mathcal{L}_{LSDA}}{\partial \phi^\dagger} &= -\frac{1}{2}\bar{\mathcal{O}}\mathcal{J}n + \frac{1}{8\pi}L\phi \, . \end{aligned}$$

5.4.2 Self-interaction correction

The LDA and LSDA exchange-correlation functionals suffer from self-interaction errors: the functionals do not correctly subtract away the interaction of an electron with its own Hartree field when the electron density is not uniform. Perdew and Zunger [78] proposed a scheme to correct for these errors (the SIC-LDA which we simply refer to as SIC below).

The idea is to subtract the individual electrostatic and exchange-correlation contribution due to the density $n_i(r) = \|\psi_i(r)\|^2$ of each quantum state $\psi_i(r)$ from the LDA functional. This procedure has the virtue of yielding the correct result for a one-electron system as well as correcting for the Hartree self-interaction exactly. In terms of our formalism, we define the density matrix P_i and electron density n_i for

the state i and relate them to the total density matrix P and total electron density n through

$$P_i = Ce_i f_i e_i^\dagger C^\dagger \ , \ P = \sum_i P_i \ \ ; \ \ n_i = \mathrm{diag}(\mathcal{I} P_i \mathcal{I}^\dagger) \ , \ n = \sum_i n_i \ ,$$

where e_i is the column vector with unity in the ith entry and zero elsewhere. In addition to the total Hartree field ϕ, we also introduce Hartree fields ϕ_i for each state i, and the SIC Lagrangian takes the form

$$\begin{aligned} \mathcal{L}_{SIC}(C, \{\phi_i\}) &= -\frac{1}{2}\mathrm{Tr}\left(FC^\dagger LC\right) + (\mathcal{J}n)^\dagger \left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n) - \bar{\mathcal{O}}\phi\right] + \frac{1}{8\pi}\phi^\dagger L\phi \\ &\quad - \sum_i (\mathcal{J}n_i)^\dagger \left[\mathcal{O}\mathcal{J}\epsilon_{xc}(n_i) - \bar{\mathcal{O}}\phi_i\right] - \frac{1}{8\pi}\sum_i \phi_i^\dagger L\phi_i \, . \end{aligned}$$

Setting the variation with respect to ϕ_i and ϕ to zero (cf. Section 5.3.6) results in the Poisson equations

$$L\phi = 4\pi\bar{\mathcal{O}}\mathcal{J}n \ , \ L\phi_i = 4\pi\bar{\mathcal{O}}\mathcal{J}n_i \, .$$

Substituting these solutions into the SIC Lagrangian yields the familiar SIC energy

$$\begin{aligned} E_{SIC}(C) &= -\frac{1}{2}\mathrm{Tr}\left(FC^\dagger LC\right) + (\mathcal{J}n)^\dagger \left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n) - \frac{1}{2}\bar{\mathcal{O}}(4\pi L^{-1}\bar{\mathcal{O}}\mathcal{J}n)\right] \\ &\quad - \sum_i (\mathcal{J}n_i)^\dagger \left[\mathcal{O}\mathcal{J}\epsilon_{xc}(n_i) - \frac{1}{2}\bar{\mathcal{O}}(4\pi L^{-1}\bar{\mathcal{O}}\mathcal{J}n_i)\right] . \end{aligned}$$

The derivatives of the SIC Lagrangian with respect to the density matrices P_i generate state-dependent Hamiltonians H_i and state-dependent potentials V_i given by

$$\begin{aligned} H_i &= -\frac{1}{2}L + \mathcal{I}^\dagger \left[\mathrm{Diag}(V_{sp} - V_i)\right]\mathcal{I} \ , \\ V_i &= \mathcal{J}^\dagger\mathcal{O}\mathcal{J}\epsilon_{xc}(n_i) + \left[\mathrm{Diag}\ \epsilon'_{xc}(n_i)\right]\mathcal{J}^\dagger\mathcal{O}\mathcal{J}n_i - \mathcal{J}^\dagger\bar{\mathcal{O}}\phi_i \ , \end{aligned}$$

where the state-independent potential V_{sp} is that of Eq. (5.36). The derivation of the expression for the derivative of the Lagrangian with respect to Y follows precisely the

same steps as in Section 5.3.5, and the final form is

$$\begin{aligned}\left(\frac{\partial \mathcal{L}_{SIC}}{\partial Y^\dagger}\right) &= \left(I - \mathcal{O}CC^\dagger\right)\left(\sum_i H_i C e_i f_i e_i^\dagger\right) U^{-1/2} + \mathcal{O}CQ\left(\sum_i \left[\tilde{H}_i, e_i f_i e_i^\dagger\right]\right), \\ \tilde{H}_i &= C^\dagger H_i C\,.\end{aligned}$$

An examination of this form shows that to compute the derivative, each Hamiltonian H_i need only be applied to the ith column of C (as the product Ce_i occurs in all places), so that computation of the derivative is only slightly more demanding than the corresponding LDA derivative.

The above results for the derivative are a generalization of those in [35]. Those authors, however, work in the traditional real-space representation (where necessarily all the sums over indices appear) and, at each step of the minimization, orthonormalize their wave functions, so that their expressions are a special case of ours when $U = I$.

The summary of the SIC Lagrangian and derivatives follows:

$$\begin{aligned}\mathcal{L}_{SIC}(Y, \phi, \{\phi_i\}) &= -\frac{1}{2}\mathrm{Tr}\left(FC^\dagger LC\right) + (\mathcal{J}n)^\dagger\left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n) - \bar{\mathcal{O}}\phi\right], \\ &\quad - \sum_i (\mathcal{J}n_i)^\dagger\left[\mathcal{O}\mathcal{J}\epsilon_{xc}(n_i) - \bar{\mathcal{O}}\phi_i\right] + \frac{1}{8\pi}\phi^\dagger L\phi - \frac{1}{8\pi}\sum_i \phi_i^\dagger L\phi_i\,, \\ \frac{\partial \mathcal{L}_{SIC}}{\partial Y^\dagger} &= \left(I - \mathcal{O}CC^\dagger\right)\left(\sum_i H_i C e_i f_i e_i^\dagger\right) U^{-1/2} + \mathcal{O}CQ\left(\sum_i \left[\tilde{H}_i, e_i f_i e_i^\dagger\right]\right), \\ \frac{\partial \mathcal{L}_{SIC}}{\partial \phi^\dagger} &= -\frac{1}{2}\bar{\mathcal{O}}\mathcal{J}n + \frac{1}{8\pi}L\phi \quad , \quad \frac{\partial \mathcal{L}_{SIC}}{\partial \phi_i^\dagger} = \frac{1}{2}\bar{\mathcal{O}}\mathcal{J}n_i - \frac{1}{8\pi}L\phi_i\,.\end{aligned}$$

5.4.3 Band-structure and fixed Hamiltonian calculations

Very often, we aim to find a set of quantum states $\{\psi_i\}$ that are the lowest energy eigenstates of a fixed Hamiltonian. One case where this occurs is in the calculation of band structures for solids within DFT, where one has already found the stationary point of the Lagrangian and the optimal electron density $n(r)$. One then aims to explore the band structure for various values of k-vectors. (See Appendix C for a

full discussion of k-points.) This requires finding the lowest energy eigenstates of the Hamiltonian. The problem is the same as a tight-binding calculation in the sense that the Hamiltonian is fixed and the electronic energy of the system is sought after, i.e. the minimum of the expectation value of the Hamiltonian among an orthonormal set of states. In both cases, the approach described below is most useful when the number of basis functions is much larger than the number of states $\{\psi_i\}$ so that direct diagonalization of the Hamiltonian is computationally prohibitive.

In such cases, we have a Hamiltonian H, and we expand our wave functions as shown in Eq. (5.7). We must minimize the energy E

$$E = \mathrm{Tr}(C^\dagger H C)\,.$$

We introduce unconstrained variables Y in the same way as before (Eq. (5.31) and onwards). The variation of the energy is given by

$$dE = \mathrm{Tr}(H\, d[Y U^{-1} Y^\dagger]) = 2\,\mathrm{Re}\,\mathrm{Tr}\left[dY^\dagger \left(\frac{\partial E}{\partial Y^\dagger}\right)\right]\,,$$

where the derivative of E is

$$\left(\frac{\partial E}{\partial Y^\dagger}\right) = (I - \mathcal{O}CC^\dagger)HCU^{-1/2}\,.$$

When we are at the minimum of E, we have an orthonormal set C that spans the subspace of the lowest-energy eigenstates of H. The minimum value of E is the electronic energy for the case of a tight-binding Hamiltonian. If the energy eigenvalues and eigenvectors are desired, we diagonalize the subspace Hamiltonian $\tilde{H} = C^\dagger H C$ to obtain the eigenvalues ϵ. We then use the unitary matrix S which diagonalizes $\tilde{H}$, $\tilde{H} = S(\mathrm{Diag}\ \epsilon)S^\dagger$, to find the expansion coefficients for the eigenstates, given by the product CS. The summary of key equations follows.

$$
\begin{aligned}
E(Y) &= \mathrm{Tr}(C^\dagger H C)\,, \\
\frac{\partial E}{\partial Y^\dagger} &= (I - \mathcal{O}CC^\dagger)HCU^{-1/2}\,.
\end{aligned}
$$

5.4.4 Unoccupied states

A slightly more complex variant of the previous problem arises when we have converged a calculation, found the orthonormal states C spanning the occupied subspace, and are interested in finding the eigenvalues and eigenstates for the low-lying unoccupied states. For example, let us say that we have converged a calculation in an insulator or semi-conductor, where the occupied space specifies the valence band. We wish to find the low-lying conduction states in order to study the band structure and band-gap of the material.

Thus, we start with a fixed Hamiltonian H and a fixed set of occupied states C. We aim to find a set of orthonormal unoccupied states D that are orthogonal to C and which also minimize the expectation of the Hamiltonian. Specifically, we wish to minimize $E = \mathrm{Tr}(D^\dagger H D)$ under the orthogonality constraint $D^\dagger \mathcal{O} C = 0$.

We introduce a set of unconstrained states Z. We project out the part of Z lying in the occupied subspace by using the projection operator $\bar{\rho}^\dagger = I - CC^\dagger\mathcal{O}$ of Section 5.3.6,

$$
D = \bar{\rho}^\dagger Z X^{-1/2} \quad \text{where} \quad X = (\bar{\rho}^\dagger Z)^\dagger \mathcal{O} (\bar{\rho}^\dagger Z)\,.
$$

Then, following the results of the previous section, the differential of E is given by

$$
dE = \mathrm{Tr}(H\, d[\bar{\rho}^\dagger Z X^{-1} \bar{\rho} Z^\dagger]) = 2\,\mathrm{Re}\,\mathrm{Tr}\left[dZ^\dagger \left(\frac{\partial E}{\partial Z^\dagger}\right)\right]\,,
$$

where the derivative of E with respect to Z is given by

$$
\left(\frac{\partial E}{\partial Z^\dagger}\right) = (I - \mathcal{O}CC^\dagger)(I - \mathcal{O}DD^\dagger)HDX^{-1/2}.
$$

As expected, the derivative has two projection operators: $\bar{\rho} = I - \mathcal{O}CC^\dagger$, which

projects out the component lying in the occupied subspace, and $(I - \mathcal{O}DD^\dagger)$, which projects out the component lying in the portion of the unoccupied subspace under consideration. Minimization of E leads to a set of states D that span the lowest-lying unoccupied states. At the minimum, the resulting unoccupied subspace Hamiltonian $\bar{H} = D^\dagger H D$ can be diagonalized to obtain the desired eigenvalues and eigenstates.

The energy and its gradient are summarized by

$$\begin{aligned} E(Z) &= \mathrm{Tr}(D^\dagger H D)\,, \\ \frac{\partial E}{\partial Z^\dagger} &= (I - \mathcal{O}CC^\dagger)(I - \mathcal{O}DD^\dagger)HDX^{-1/2}\,, \\ D &= \bar{\rho}^\dagger Z X^{-1/2} \quad , \quad X = (\bar{\rho}^\dagger Z)^\dagger \mathcal{O}(\bar{\rho}^\dagger Z) \quad , \quad \bar{\rho} = I - \mathcal{O}CC^\dagger\,. \end{aligned}$$

5.4.5 Variational density-functional perturbation theory

In this final application, we consider perturbation theory within a single-particle formalism, which is required to compute response functions. Specifically, we consider the important case of linear response, which was first dealt with in [36]. We imagine that we have converged the calculation of the zeroth-order (i.e. unperturbed) configuration and have found the zeroth-order wave functions C_0 for our problem. We now wish to find the first-order changes of the wave functions, C_1, due to an external perturbation to the system. Depending on the type of perturbation applied, the variation C_1 allows for the calculation of the corresponding response functions. For example, the displacement of atoms along a phonon mode allows for the computation of the dynamical matrix for that mode whereas perturbations due to an external electrostatic field allow for calculation of the dielectric tensor.

Regardless of the physical situation, all perturbations enter as a change in the ionic (or external) potential V_{ion} which drives the electronic system. Letting λ be the perturbation parameter, we expand any physical quantity A in powers of λ and let A_n be the coefficient of λ^n in the expansion. A few examples follow

$$V_{ion} = V_{ion,0} + \lambda V_{ion,1} + \lambda^2 V_{ion,2} + \cdots$$

$$
\begin{aligned}
C &= C_0 + \lambda C_1 + \lambda^2 C_2 + \cdots \\
n &= n_0 + \lambda n_1 + \lambda^2 n_2 + \cdots
\end{aligned}
$$

As is well known from perturbation theory, the first order change $V_{ion,1}$ determines the first order shift of the wave functions C_1.

The work of [36] shows that C_1 can be obtained via the constrained minimization of an auxiliary quadratic functional of C_1. In our matrix-based notation, for the case of constant fillings (taken to be unity) and the LDA approximation, this quadratic functional is given by

$$
\begin{aligned}
E(C_1) &= \mathrm{Tr}\left\{C_1^\dagger H_0 C_1 - C_1^\dagger \mathcal{O} C_1 [\mathrm{Diag}\ \epsilon_0]\right\} \\
&\quad + (\mathcal{J} n_1)^\dagger \left\{V_{ion,1} + \mathcal{O}\mathcal{J}\left[\mathrm{Diag}\ a(n_0)\right] n_1 - \frac{1}{2}\bar{\mathcal{O}}\phi_1\right\} + E_{nonvar}\ .
\end{aligned}
$$

The energy E_{nonvar} contains terms that depend only on C_0 or the Ewald sum over atomic positions and need not concern us any further. The zeroth-order Hamiltonian $H_0 = -\frac{1}{2}L + \mathcal{I}^\dagger[\mathrm{Diag}\ V_0]\mathcal{I}$ is the same as that of Eq. (5.36) where we have simply renamed the zeroth-order single-particle potential to V_0. The diagonal matrix Diag ϵ_0 holds the eigenvalues of the zeroth-order Hamiltonian. The vector $a(n_0)$ is found by evaluating the function $a(n) = \frac{d^2}{dn^2}\left(n\epsilon_{xc}(n)\right)$ on the zeroth-order electron density n_0 over the real-space grid. The vector n_1, the first order shift of the electron density, is given by

$$
n_1 = 2\ \mathrm{Re\ diag}\left(\mathcal{I} C_0 C_1^\dagger \mathcal{I}^\dagger\right)\ .
$$

The first order change of the Hartree potential ϕ_1 is the solution of the Poisson equation $\phi_1 = 4\pi L^{-1}\bar{\mathcal{O}}\mathcal{J}n_1$.

Given the quadratic nature of $E(C_1)$, its differential with respect to C_1 follows immediately and is given by

$$
dE = 2\ \mathrm{Re\ Tr}\left\{dC_1^\dagger\left(H_0 C_1 - \mathcal{O}C_1[\mathrm{Diag}\ \epsilon_0] + \mathcal{I}^\dagger\left[\mathrm{Diag}\ V_1\right]\mathcal{I}C_0\right)\right\}\ ,
$$

where the first-order single-particle potential V_1 is given by

$$V_1 = \mathcal{J}^\dagger V_{ion,1} + \mathcal{J}^\dagger \mathcal{O} \mathcal{J} \left[\text{Diag } a(n_0)\right] n_1 + \left[\text{Diag } a(n_0)\right] \mathcal{J}^\dagger \mathcal{O} \mathcal{J} n_1 - \mathcal{J}^\dagger \bar{\mathcal{O}} \phi_1 \,.$$

The constraint to be obeyed during the minimization is that the first-order shifts C_1 be orthogonal to the zeroth-order wave functions C_0,

$$C_1^\dagger \mathcal{O} C_0 = 0 \,.$$

This constraint is easily handled in the manner of the previous section by using a projection operator. We introduce an unconstrained set of wave functions Y_1 from which we project out the part laying in the space spanned by C_0,

$$C_1 = (I - C_0 C_0^\dagger \mathcal{O}) Y_1 \,.$$

Based on this relation, we find the gradient of E with respect to Y_1

$$\begin{aligned} dE &= 2 \text{ Re Tr} \left\{ dY_1^\dagger \left(\frac{\partial E}{\partial Y_1^\dagger} \right) \right\} \text{ where} \\ \left(\frac{\partial E}{\partial Y_1^\dagger} \right) &= (I - \mathcal{O} C_0 C_0^\dagger) \left\{ H_0 C_1 - \mathcal{O} C_1 [\text{Diag } \epsilon_0] + \mathcal{I}^\dagger \left[\text{Diag } V_1\right] \mathcal{I} C_0 \right\} \,. \end{aligned}$$

Finally, we can convert the energy function into a Lagrangian by letting ϕ_1 be a free variable and by adding the appropriate Hartree self-energy and coupling to n_1. We arrive at the summarized expressions

$$\begin{aligned} \mathcal{L}(Y_1, \phi_1) &= \text{Tr} \left\{ C_1^\dagger H_0 C_1 - C_1^\dagger \mathcal{O} C_1 [\text{Diag } \epsilon_0] \right\} + E_{nonvar} \\ &\quad + (\mathcal{J} n_1)^\dagger \left\{ V_{ion,1} + \mathcal{O} \mathcal{J} \left[\text{Diag } a(n_0)\right] n_1 - \bar{\mathcal{O}} \phi_1 \right\} + \frac{1}{8\pi} \phi_1^\dagger L \phi_1 \,, \\ \frac{\partial \mathcal{L}}{\partial Y_1^\dagger} &= (I - \mathcal{O} C_0 C_0^\dagger) \left\{ H_0 C_1 - \mathcal{O} C_1 [\text{Diag } \epsilon_0] + \mathcal{I}^\dagger \left[\text{Diag } V_1\right] \mathcal{I} C_0 \right\} \,, \\ \frac{\partial \mathcal{L}}{\partial \phi_1^\dagger} &= -\frac{1}{2} \bar{\mathcal{O}} \mathcal{J} n_1 + \frac{1}{8\pi} L \phi_1 \,. \end{aligned}$$

calc-energy(Y): calculates U, $U^{-1/2}$, C, n, ϕ, and returns the LDA energy

- Calculate the overlap $U = Y^\dagger \mathcal{O} Y$
- Diagonalize $U = W \mu W^\dagger$ and calculate $U^{-1/2} = W \mu^{-1/2} W^\dagger$
- Calculate the orthonormal states $C = Y U^{-1/2}$
- Calculate the electron density $n = f \cdot \mathrm{diag}\left\{(\mathcal{I}C)(\mathcal{I}C)^\dagger\right\}$
- Solve the Poisson equation $\phi = 4\pi L^{-1} \bar{\mathcal{O}} \mathcal{J} n$
- Return the LDA energy E_{LDA}:

$$E_{LDA} = -\frac{f}{2}\mathrm{Tr}(C^\dagger L C) + (\mathcal{J}n)^\dagger \left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n) - \frac{1}{2}\bar{\mathcal{O}}\phi\right]$$

Figure 5-2: LDA energy routine (DFT++ formalism)

5.5 Minimization algorithms

In this section, we show how the DFT++ formalism can succinctly specify the algorithm which finds the stationary point of the Lagrangian or energy function (derived in the previous sections). Such an algorithm only requires the value and derivative of the objective function, which is the reason that we have repeatedly emphasized the importance of these two quantities in our analysis above. Once we choose a minimization algorithm, we need only "plug in" the objective function and its derivative into the appropriate slots. Furthermore, since the DFT++ formalism is compact and at the same time explicit, once we specify the operations that must be performed for a given algorithm in our notation, the transition to coding on a computer is trivial: the formulae translate directly into computer code (as shown in Section 5.6).

Specifically, we aim to find the stationary point of the Lagrangian of Eq. (5.28) with respect to Y and ϕ and possibly B. A direct search for the stationary point is possible, and at the saddle point, both the Kohn-Sham and Poisson equations hold true simultaneously. This highly effective strategy has been followed in [3]. Alternatively, other approaches to reach a solution of these equations through self-consistent iteration and use of Broyden-like schemes [19] may be considered.

However, in order to make direct contact with DFT calculations within the traditional minimization context [77], and to keep our presentation as simple as possible, we choose instead to solve the Poisson equation (Eq. (5.42)) for the optimal ϕ at each iteration of the minimization algorithm. For cases where L^{-1} is easy to compute (e.g. the plane-wave basis where L is diagonal), we may compute the solution ϕ directly from Eq. (5.43). Otherwise, the straightforward approach of maximizing the quadratic functional $\mathcal{G}(\phi) = (1/8\pi)\phi^\dagger L\phi - (\mathcal{J}n)^\dagger \bar{\mathcal{O}}\phi$ via conjugate gradients (or some other method) achieves the same goal. For certain basis sets, multigrid methods or other specialized techniques are possibilities as well [3, 97, 73]. Once the Poisson equation has been solved, the remaining free variable is the matrix of coefficients Y, and the aim of the calculation is to minimize the energy E_{LDA} of Eq. (5.44) with respect to Y.

As shown in Section 5.2, the value and Y-derivatives of the Lagrangian $\mathcal{L}_{LDA}$ and energy E_{LDA} are identical if we evaluate the Lagrangian-based expressions using the Hartree potential ϕ which is the solution of Poisson's equation. Therefore, in our algorithms below, we can use expressions derived for derivatives of the Lagrangian when dealing with the energy.

5.5.1 Semiconducting and insulating systems

Consider the case of a semiconducting system with a large unit cell. The fillings are constant, $F = fI$, and we will use a single k-point at $k = 0$ (as appropriate for a large cell). The simplest algorithm for minimizing the energy is the steepest descent method: we update Y by shifting along the negative gradient of the energy with respect to Y, scaled by a fixed multiplicative factor. As a first organizational step, we introduce the function calc-energy(Y), whose code we display in Figure 5-2. Given Y, this function computes the overlaps U and $U^{-1/2}$, the orthonormalized C, the electron density n, the solution to Poisson's equation ϕ, and returns the LDA energy. Figure 5-3 displays the steepest descent algorithm as it appears in the DFT++ language.

We would like to emphasize a number of points regarding this code. First, the algorithm optimizes all the wave functions (i.e. the entire matrix Y) at once, leading

- Calculate the ionic potential V_{ion} (Eq. (5.13))
- Randomize Y and choose the stepsize λ
- Iterate until E_{LDA} is sufficiently converged:
  - Calculate the energy: $E_{LDA} = \text{calc-energy}(Y)$
  - Calculate the single-particle potential V_{sp}:

$$V_{sp} = \mathcal{J}^\dagger \left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n) - \bar{\mathcal{O}}\phi \right] + [\text{Diag } \epsilon'_{xc}(n)]\mathcal{J}^\dagger\mathcal{O}\mathcal{J}n$$

  - Calculate the gradient of E_{LDA}:

$$G = \frac{\partial E_{LDA}}{\partial Y^\dagger} = f(I - \mathcal{O}CC^\dagger)HCU^{-1/2}$$

  - Take a step down the gradient: $Y - \lambda G \to Y$

Figure 5-3: Steepest descent algorithm

to a very effective minimization and a complete avoidance of charge creep [4] or other instabilities. Second, the code is independent of the basis set used: the basis-dependent operators L, $\mathcal{O}$, etc., are coded as low-level functions that need to be written only once. The choice of physical system and minimization algorithm is a high-level matter that is completely decoupled form such details. Third, the figure shows *all* the operations required for the entire minimization loop. That this is possible is grace to the succinct matrix formalism.

The only part of the algorithm of Figure 5-3 that is specific to the steepest descent method is the last operation where Y is updated. To generalize to a preconditioned conjugate-gradient algorithm is quite straightforward, and we specify the necessary changes below.

Conjugate-gradient algorithms require line minimization of the objective function along a specified direction, i.e. an algorithm is needed that minimizes $E(\lambda) \equiv E_{LDA}(Y + \lambda X)$ with respect to the real number λ for a fixed search vector X. The subject of line minimization is rich, and an abundance of algorithms with varying degrees of complexity exist in the literature. (For a brief introduction see [79].) However, for typical DFT calculations, most of the effort for the calculation is spent in

quadratic-line-min(Y, X, E, G): quadratic minimization of E_{LDA}
X is the search direction for the minimization
E is the energy E_{LDA} at Y
G is the gradient of E_{LDA} at Y

- Compute the directional derivative of E_{LDA} at Y: $\Delta = 2\,\mathrm{Re}\ \mathrm{Tr}(X^\dagger G)$
- Compute the energy at trial position $E_t = \text{calc-energy}(Y + \lambda_t X)$
- Compute the curvature of the energy function $c = [E_t - (E + \lambda_t \Delta)]/\lambda_t^2$
- Compute the minimizing shift $\tilde{\lambda} = -\Delta/(2c)$
- Shift to the minimum: $Y + \tilde{\lambda} X \to Y$

Figure 5-4: Quadratic line minimizer

the quadratic basin close to the minimum. Thus, a simple and efficient line-minimizer that exploits this quadratic behavior should be sufficient, and we have found this to be the case in our work.

Our line-minimizer takes the current value of the energy and its derivative as well as the value of the energy at the shifted trial configuration $Y + \lambda_t X$ (where λ_t is a trial step-size) to achieve the quadratic fit $E(\lambda) \approx E + \Delta\lambda + c\lambda^2$. Here, Δ is the directional derivative of E with respect to λ, and c is the curvature of E with respect to λ. The line-minimizer then moves to the minimum of this quadratic fit located at $\tilde{\lambda} = -\Delta/(2c)$. Figure 5-4 shows the code for this line-minimizer.

Using this line minimizer, we present the entire preconditioned conjugate-gradient algorithm in Figure 5-5. Note that we have omitted some of the formulae which are identical to those of Figure 5-3. A simple diagonal preconditioning, as described in [77], is quite effective for plane-wave basis sets, and the operator K is the preconditioner in the algorithm of the figure.

5.5.2 Metallic and high-temperature systems

While the degrees of freedom in the variable Y are sufficient to describe any electronic system, the convergence of minimization algorithms can be hampered by ill-

- Calculate the ionic potential V_{ion}
- Randomize Y_0
- For j=0, 1, 2, ...
  - Let $E_j = \text{calc-energy}(Y_j)$
  - Calculate the single-particle potential V_{sp}
  - Calculate the gradient: $G_j = (\partial E_{LDA}/\partial Y^\dagger)|_{Y_j}$
  - Apply the preconditioner: $\Gamma_j = K(G_j)$
  - If $j > 0$ then set $\alpha_j = \text{Tr}(G_j^\dagger \Gamma_j)/\text{Tr}(G_{j-1}^\dagger \Gamma_{j-1})$; otherwise $\alpha_j = 0$.
  - Compute the search direction $X_j = \Gamma_j + \alpha_j X_{j-1}$
  - Call quadratic-line-min(Y_j, X_j, E_j, G_j)

Figure 5-5: Preconditioned conjugate-gradient algorithm

conditioning of the physical system. A standard case of such a problem is when the Fermi-Dirac fillings f_i are not constant and some fillings are very small, a situation encountered when studying metals or high-temperature insulators.

One type of ill-conditioning that arises due to states with small fillings, $f_i \ll 1$, stems from the fact that changes in such states do not affect the value of the energy E_{LDA} very much when compared to the states with large fillings, $f_i \sim 1$. Thus modes associated with the small-filling states are "soft" and we have an ill-conditioned problem. The solution to this problem, however, is rather straightforward and involves scaling the derivative of $\mathcal{L}_{LDA}$ with respect to the state ψ_i by $1/f_i$.

Much harder to deal with are soft modes due to subspace rotations which were introduced in Section 5.3.4. As we saw there, the unitary transformations V, which generate the rotations, cancel out completely from the expression for the density matrix P in the case of constant Fermi fillings, $F = fI$. Since the entire energy can be computed from P alone, the energy will not depend on V. Hence we have found that the unitary transformations V are an exact symmetry of a system with constant fillings.

However, once we introduce variations in the fillings, the symmetry is broken. Now both the density matrix and the energy change when V mixes states with different fillings. If the difference in fillings between the mixed states is small, a case typically

encountered in a real system, the mixing produces small changes in the energy. Again, we have soft modes, this time due to the breaking of the unitary subspace-rotation symmetry.

The idea of using subspace rotations was first suggested in [4]. Its use as a cure for the ill-conditioning described above was discussed and demonstrated convincingly in [65]. The strategy is first to minimize the objective function over B (since B parameterizes the rotation V) and only then perform minimization on the wave functions Y. By ensuring that we are always at the minimum with respect to B, we automatically set the derivative of E_{LDA} with respect to subspace-rotations to zero. When this is true, changing Y can not (to first order) give rise to contributions due to the soft modes, and we have eliminated the ill-conditioning.

In practice, we have found it unnecessary for our calculations to be at the absolute minimum with respect to B: being close to the minimum is sufficiently beneficial computationally. In our algorithm, we take steps along both Y and B simultaneously but ensure that our step-size in B is much larger than that in Y. As the minimization proceeds, this choice automatically drives the system to stay close to the minimum along B at all times.

Our simpler procedure has been found as effective as the original strategy of [65] and translates into using the following search direction X in the space (Y, B)

$$X = \left(\frac{\partial \mathcal{L}_{LDA}}{\partial Y^\dagger} \ , \ \eta \cdot \frac{\partial \mathcal{L}_{LDA}}{\partial B} \right)$$

where η is a numerical scaling factor. We have found $\eta \sim 30$ to be a good choice for efficient minimization while avoiding the ill-conditioning mentioned above.

As a practical showcase of the improvement in convergence in a metallic system, we study the convergence rate for the conjugate-gradient minimization of the energy of bulk molybdenum. We study the bcc cubic unit cell containing two Mo atoms. We use a plane-wave basis set (details of implementation in Appendix A) with an electronic cutoff of 22.5 Hartree (45 Ryd), and a 4×4×4 cubic k-point mesh to sample to Brillouin zone. The electronic temperature used is $k_B T$=0.0037 Hartree (0.1 eV),

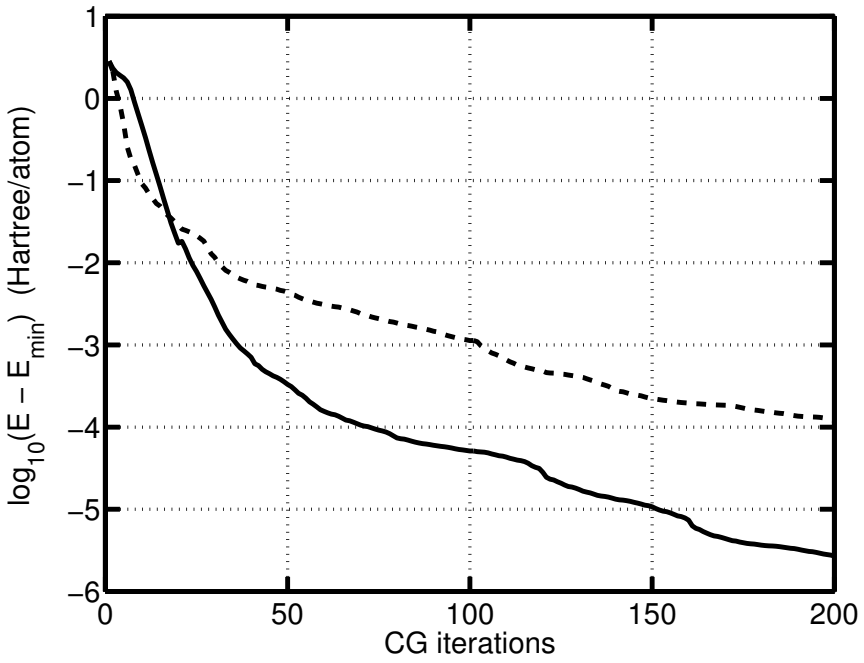

Figure 5-6: Effect of subspace rotation on convergence – Convergence of conjugate-gradient minimization with (solid) and without (dashed) the use of subspace rotations for the case of a metallic system. Both minimizations use the same random wave functions as their starting points. The horizontal axis is the number of conjugate-gradient iterations and the vertical axis is the energy per atom above the minimum energy in Hartree per atom in logarithmic units. See the text for further details.

the Fermi fillings are recomputed every twenty conjugate-gradient steps based on the eigenvalues of the subspace Hamiltonian from the previous iteration, and a value of $\eta = 30$ is used to calculate the search direction. Non-local pseudopotentials of the Kleinmann-Bylander form [53] are used with p and d projectors. Figure 5-6 presents a plot of the convergence of the energy per atom to its minimum value (as determined by a run with many more iterations than shown in the figure). We compare minimization with and without the use of subspace rotation variables, and both minimizations are started with the same initial random wave functions. The extra cost required for the use of subspace rotations was very small in this case, the increase in the time spent per iteration being less than two percent. As we can see, the use of subspace rotations dramatically improves the convergence rate.

5.6 Implementation, optimization, and parallelization

In this final section we address how the DFT++ formalism can be easily and effectively implemented on a computer, and what steps must be taken to ensure efficient optimization and parallelization of the computations. As is clear from the previous sections, the DFT++ formalism is firmly based on linear algebra. The objects appearing in the formalism are vectors and matrices. The computations performed on these objects are matrix addition and multiplication and the application of linear operators. An important benefit is that linear-algebraic products involve matrix-matrix multiplications (i.e. BLAS3 operations), which are precisely those operations that achieve the highest performance.

We use an object-oriented approach and the C++ programming language to render the implementation and coding as easy as possible. In addition, object-oriented programming introduces modularity and localization of computational kernels allowing for effective optimization and parallelization. In the sections below, we present outlines of our implementation, optimization, and parallelization strategies.

5.6.1 Object-oriented implementation in C++

In our work, we have found that an object-oriented language such as C++ is ideal for implementing the required vectors and matrices and for defining the operations on them in a manner that follows the DFT++ formalism as closely as possible. The object-oriented approach presents a number of advantages.

First, the programming task becomes highly modular: we identify the object types needed and the operations that must be performed on them, and we create a separate module for each object that can be tested independently. For example, we define the class of matrices and the operations on them (e.g. multiplication, addition, diagonalization, etc.), and we can test and debug this matrix module separate from any other considerations. Second, we gain transparency: the internal structure or

functioning of an object can be modified for improved performance without requiring any changes to higher-level functions that use the object. This gives us a key feature in that the high-level programmer creating new algorithms or testing new energy functionals does not need to know about or interact with the lower-level details of how the objects actually are represented or how they function. Third, this separation of high-level function from low-level implementation allows for a centralization and reduction in the number of computational kernels in the code: there can be a large variety of high-level objects for the convenience of the programmer, but as all the operations defined on them are similar linear-algebraic ones, only a few actual routines must be written. Fourth, modularity produces shorter and more legible code. This, combined with the object-oriented approach, implies that the high-level computer code will read the same as the equations of the DFT++ formalism so that debugging will amount to checking formulae, without any interference of cumbersome loops and indices.

To give a concrete example of what this means, consider the simplest object in the formalism, a column vector such as the electron density n. In C++, we define an object class `vector` which will contain an array of `complex` numbers (itself a lower-level object). A `vector v` has a member `v.size` specifying the number of rows it contains as well as a pointer to the array containing the data. We define the action of the parenthesis so as to allow convenient access of the `i`th element of `v` via the construction `v(i)`.

A very common operation between two vectors v and w is the scalar product $v^\dagger w$. We implement this by defining (in C++ parlance overloading) an operator `^` that takes two `vectors` and returns a `complex` result. The code accomplishing this is

```
friend complex operator^(vector &v, vector &w)
{
    complex result = 0;

    for(int i=0; i < v.size; i++)
       result += conjugate(v(i))*w(i);
```

```
    return result;
}
```

An example of an operator acting on vectors is the inverse transform $\mathcal{J}$. This operator can be coded as a function `J` that takes a `vector v` as its argument and returns the `vector` result `J(v)`. Of course, the details of what $\mathcal{J}$ does internally are basis-dependent.

Based on this definition, the evaluation of the electron-ion interaction energy of Section 5.3.3, given by the expression $E_{e-i} = (\mathcal{J}n)^\dagger V_{ion}$, is coded by

```
E_ei = J(n)^V_ion;
```

where `V_ion` is the vector V_{ion} of Eq. (5.13).

Following the same idea, we define a `matrix` class to handle the matrices such as U, W, $\tilde{H}$, etc. that occur in the formalism. Physically, expansion coefficients such as Y and C are collections of column vectors (a column of coefficients $C_{\alpha i}$ for each wave function ψ_i), and we define the class `column_bundle` to handle these objects. Although mathematically `column_bundle`s such as Y and C are matrices, the choice not to use the `matrix` class for representing them is deliberate, as Y and C have a distinct use and a special physical meaning that an arbitrary matrix will not have. In this way, we can tailor our objects to reflect the physical content of the information they contain. Of course, when a matrix multiplication is performed, such as when we evaluate the expression $C = YU^{-1/2}$, the `column_bundle` class and `matrix` class call a single, low-level, optimized multiplication routine. We thus gain flexibility and legibility of codes on the higher levels while avoiding an accumulation of specialized functions on the lower levels.

As a concrete example of the power and ease of this style of programming, consider writing the code for the function calc-energy(Y) of Figure 5-2. In order to do so, we will need a few more operators. Following the example of the `vector`s, we

```
complex calc_energy(column_bundle &Y, column_bundle &C,
                    matrix &U, matrix &Umhalf,
                    vector &n, vector &phi,
                    double f, vector &V_ion)
{
  complex E_LDA;

  // calc_Umhalf(U) returns U^{-1/2} given U
  U = Y^O(Y);
  Umhalf = calc_Umhalf(U);
  C = Y*Umhalf;

  // diag_of_self_product(X) returns diag(X*adjoint(X))
  n = f*diag_of_self_product(I(C));

  phi = (4.0*PI)*invL(Obar(n));

  E_LDA = -0.5*f*Tr(C^L(C)) +
          J(n)^( V_ion + O(J(exc(n))) - 0.5*Obar(phi) );

  return E_LDA;
}
```

Figure 5-7: LDA energy routine – C++ code for the calc-energy(Y) function outlined in Figure 5-2.

define the `^` operator between two `column_bundle`s to handle Hermitian-conjugated multiplications such as $Y_1^\dagger Y_2$, where the result of the product is of type `matrix`. Next, we define the `*` operator between a `column_bundle` and `matrix` to handle multiplications of the type $YU^{-1/2}$, where the result is a `column_bundle`. We code the action of the basis-dependent operators such as $\mathcal{O}$ or $\mathcal{I}$ on a `vector` ϕ or a `column_bundle` C as function calls `O(phi)` or `I(C)`, which return the same data type as their argument. Finally, we implement multiplication by scalars in the obvious way. Figure 5-7 presents the C++ code for the energy calculation routine. The code is almost exactly the same as the corresponding expressions in the DFT++ formalism, making the translation from mathematical derivation to actual coding trivial.

As a performance issue, when computing `n`, we do not use the straightforward code `diag(X*adjoint(X))` which computes the entire matrix `X*adjoint(X)` before extracting its diagonal and is thus computationally wasteful. Rather, we have written a separate function `diag_of_self_product(X)` that takes a `column_bundle X` and computes only the required diagonal portion of `X*adjoint(X)`.

5.6.2 Scalings for dominant DFT++ operations

Before we describe our approach to optimization and parallelization, we will work out the scalings of the floating-point operation counts as a function of system size for the various computational operations in the DFT++ formalism. Thus it will be clear which optimizations and parallelizations will increase the overall performance most efficaciously. Let n be the number of wave functions $\{\psi_i\}$ and let N be the number of basis functions $\{b_\alpha(r)\}$ in the calculation. A `vector` contains N complex numbers, a `matrix` is $n \times n$, and a `column_bundle` is $N \times n$ and is the largest data structure in the computation. Both n and N are proportional to the number of atoms n_a in the simulation cell, or equivalently, to the volume of the cell. Typically, for accurate DFT calculations, the number of basis functions required is much larger than the number of quantum states, $N \gg n$.

In Table 5.1 we present the floating-point operation counts for the different operations required in the DFT++ formalism. We note that for very large systems, the basis-set independent matrix products dominate the overall computational workload. However, for medium-sized or slightly larger problems, the application of the basis-dependent operators can play an important role as well.

For most basis sets, there are techniques available in the literature that allow for efficient application of the basis-dependent operators to a column vector in $O(N)$ or $O(N \ln N)$ operations. For example, when working with plane waves, the Fast Fourier transform [17, 27] is the algorithm of choice for implementing the operators $\mathcal{I}$, $\mathcal{J}$, $\mathcal{I}^\dagger$, and $\mathcal{J}^\dagger$ and allows us to affect these operators in $O(N \ln N)$ operations. (See Appendix A for the details of a plane-wave implementation.) The operators L and $\mathcal{O}$ are already diagonal in a plane-wave basis and are thus trivial to implement. For

Operation	Examples	FLOP count
`column_bundle` multiplications	$M = Y_1^\dagger Y_2$ or $C = YU^{-1/2}$	$O(n^2N)$
`matrix` multiplications and diagonalizations	$U = W\mu W^\dagger$ or $U^{-1/2} = W\mu^{-1/2}W^\dagger$	$O(n^3)$
Applying basis-dependent operators	$\mathcal{O}Y$, LC, or $\mathcal{I}C$	$O(nN)$ or $O(nN \ln N)$
Calculating n given $\mathcal{I}C$ or calculating the kinetic energy given LC	$n = \mathrm{diag}\{(\mathcal{I}C)F(\mathcal{I}C)^\dagger\}$ or $\mathrm{Tr}\{FC^\dagger(LC)\}$	$O(nN)$
`vector` operations	$\epsilon_{xc}(n)$, $(\mathcal{J}n)^\dagger V_{ion}$, or $\bar{\mathcal{O}}\phi$	$O(N)$

Table 5.1: Floating-point operation count for various common operations in the DFT++ formalism. The size of the basis set is N and the number of wave functions is n. Thus a `column_bundle` is $N \times n$, a `matrix` is $n \times n$, and a `vector` is N long.

real-space, grid-based computations, multigrid methods [97] are highly effective for inverting L and solving the Poisson equation. Special techniques exist for multiresolution [62, 3, 10] and Gaussian [73] basis sets that allow for the application of the basis-dependent operators in $O(N)$ operations as well.

5.6.3 Optimization of computational kernels

Due to the modularity of our object-oriented approach, the physical nature of the problem under study is a high-level issue that is completely independent of the functioning of the few, low-level computational kernels which handle most of the calculations in the code. The aim now is to optimize these kernels to obtain maximum computational performance. By optimization we mean increasing performance on a single processor. Parallelization, by which we mean distribution of the computational task among several processors, will be addressed in the next section, but good parallel performance is only possible when each processor is working most effectively.

The computationally intensive operations involved in the DFT++ formalism fall

into two overall classes. First are the application of the basis-dependent operators such as L, $\mathcal{O}$, $\mathcal{I}$, etc., whose operation counts scale quadratically in the system size (see Table 5.1). Given their basis-dependent nature, no general recipe can be given for their optimization. However, for many widely used basis sets, highly efficient methods exist in the literature which allow for the application of these operators, and we refer the reader to the references cited in the preceding section. For the case of plane waves, we have used the FFTW package [30] to affect the Fourier transformations. This highly portable, freely obtainable software library provides excellent per processor performance across many platforms.

The second class of operations are the basis-independent matrix products, and we will now consider the optimization of these operations. As a case study, we examine the Hermitian-conjugated matrix product between two `column_bundle`s, which occurs in an expression such as $Y_1^\dagger Y_2$ and which is coded using the `column_bundle^column_bundle` operator as `Y_1^Y_2`. The most "naive" and straightforward implementation of this operator in C++ is given by

```
friend matrix operator^(column_bundle &Y_1, column_bundle &Y_2)
{
   // A Y_1.n_columns x Y_2.n_columns holds the result
   matrix M(Y_1.n_columns,Y_2.n_columns);
   int i,j,k;

   for(i=0; i < Y_1.n_columns; i++)
      for(j=0; j < Y_2.n_columns; j++) {
         M(i,j) = 0;
         for(k=0; k < Y_1.n_rows; k++)
            M(i,j) += conjugate(Y_1(k,i))*Y_2(k,j);
   }
   return M;
}
```

While easy to follow, this implementation is quite inefficient on modern computer architectures for large matrix sizes because the algorithm does not take any advantage of caching. The access pattern to memory is not localized, and the processor must continually read and write data to the slower-speed main memory instead of to the much faster (but smaller) cache memory.

One solution to this problem is to resort to vendor-specific linear algebra packages. For example, one can use a version of LAPACK optimized for the computational platform at hand, and this generally results in very good performance. An alternative choice is to perform the optimizations by hand. While this second choice may sacrifice some performance, it does ensure highly portable code, and this is the strategy that we have followed here.

Our basic approach to increasing performance is to use blocking: we partition each of the input and output matrices into blocks of relatively small dimensions, and the matrix multiplication is rewritten as a set of products and sums over these smaller blocks. Provided that the block sizes are small enough, say 32×32 or 64×64 for todays' typical processor and memory architectures, both the input and output blocks will reside in the high-speed cache memory and fast read/write access to the cache will dramatically improve performance. Figure 5-8 shows a schematic diagram of how the product $M = Y_1^\dagger Y_2$ would be carried out in a blocked manner.

Please note that due to blocking, the task of optimization is now also modularized. We need only write a single, low-level, block-block matrix multiplication routine that should be highly optimized. All higher-level matrix multiplication functions will then loop over blocks of the input and output data and copy the contents to small, in-cache matrices which are then multiplied by calling the low-level multiplier.

Applying these ideas to our $M = Y_1^\dagger Y_2$ example, the rewritten code for the `column_bundle^column_bundle` operator takes the following form:

```
friend matrix operator^(column_bundle &Y_1, column_bundle &Y_2)
{
   matrix M(Y_1.n_columns,Y_2.n_columns); // M=Y_1^Y_2 holds the result
```

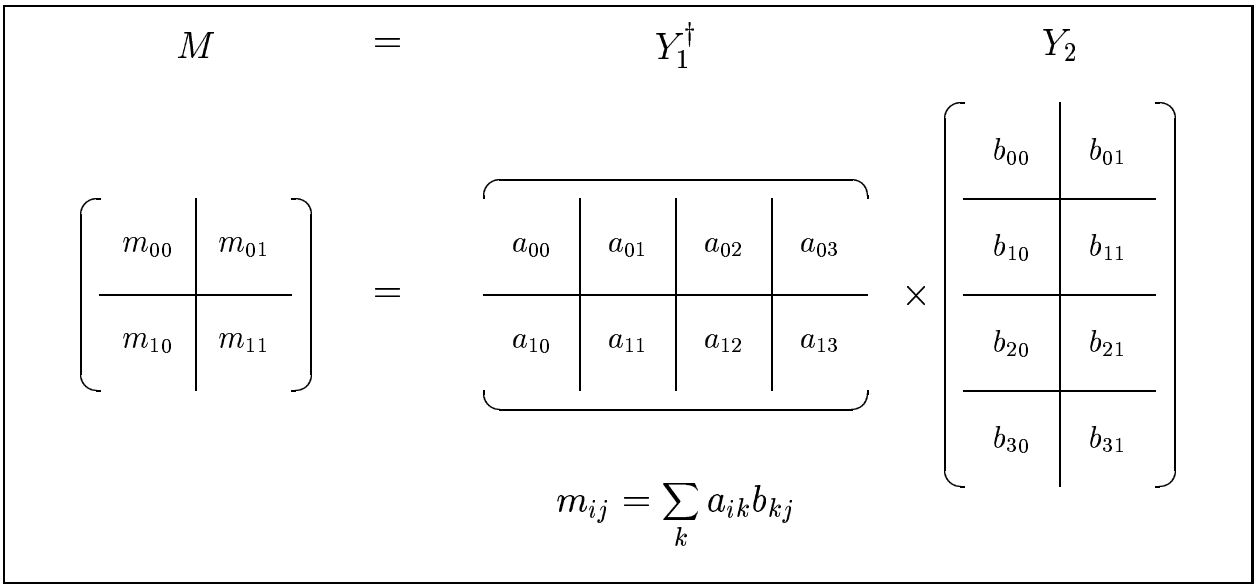

Figure 5-8: Blocked matrix multiplication – A schematic of the blocked matrix multiplication method applied to computing the product $M = Y_1^\dagger Y_2$. The blocks m_{ij}, a_{ij}, and b_{ij} are matrices of small size (e.g. 32×32). Our schematic shows how each of the matrices is partitioned into blocks and how the result blocks m_{ij} are to be computed.

```
int B = 32; // Pick a reasonable block size

matrix in1(B,B),in2(B,B),out(B,B); // input and output blocks

int ib,jb,kb,i,j,k;

// loop over blocks of the output
for(ib=0; ib < Y_1.n_columns; ib+=B)
   for(jb=0; jb < Y_2.n_columns; jb+=B) {

      // zero the output block
      for(i=0; i < B; i++)
         for(j=0; j < B; j++)
            out(i,j) = 0;

      // loop over blocks to be multiplied
```

```
            for(kb=0; kb < Y_1.nrows; kb+=B) {

               // load data into input blocks
               for(i=0; i < B; i++)
                  for(k=0; k < B; k++) {
                     in1(i,k) = conjugate(Y_1(kb+k,ib+i));
                     in2(i,k) = Y_2(kb+k,jb+i);
               }

               // perform the block multiplication
               low_level_mutliply(in1, in2, out, B);
            }

            // write the result to memory
            for(i=0; i < B; i++)
               for(j=0; j < B; j++)
                  M(ib+i,jb+j) = out(i,j);
      }

      return M;
   }
```

The function `low_level_multiply` performs the block-block multiplication of the input blocks and accumulates into the output block. Not only the `column_bundle^column_bundle` operator but *all* matrix multiplications can be blocked in a similar way and will thus call the low-level multiplier.

The simplest implementation for `low_level_multiply` is the straightforward one:

```
void low_level_multiply(matrix &in1, matrix &in2,
                        matrix &out, int B)
{
```

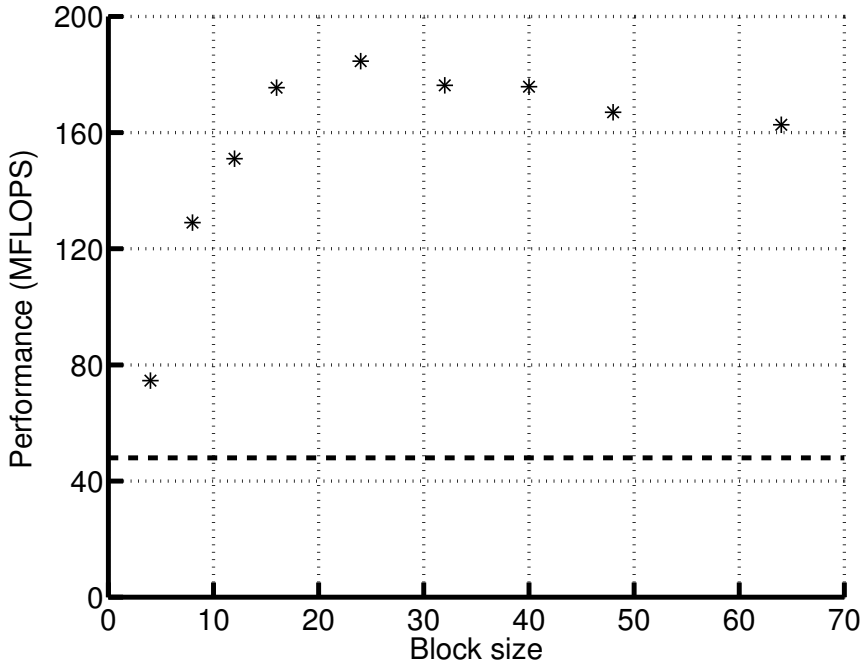

Figure 5-9: Matrix-multiplication FLOP rates (single processor) – Effect of blocking on computational performance for the matrix product $M = Y_1^\dagger Y_2$, where Y_1 and Y_2 are 10,000×200 complex-valued matrices. The horizontal axis shows the size of the square blocks used for the block-multiplication scheme described in the text. The vertical axis is the performance in mega floating-point operations per second (MFLOPS). The horizontal dashed line shows the rate for a non-blocked "naive" multiplication routine (see text).

```
    int i,j,k;
    for(i=0; i < B; i++)
       for(j=0; j < B; j++) {
          complex z = 0;
          for(k=0; k < B; k++)
             z += in1(i,k)*in2(j,k);
          out(i,j) += z;
       }
}
```

The use of this simple low-level multiplier combined with blocking can provide

a tremendous improvement. Figure 5-9 shows a plot of the performance of the $M = Y_1^\dagger Y_2$ blocked-multiplication as a function of the block size when run on a 333 MHz SUN Ultrasparc 5 microprocessor. For comparison, the performance of the "naive" code with no blocking, which was presented above, is also indicated in the figure. Initially, the performance increases dramatically with increasing block size due to more effective caching. However, there is an optimal size above which performance decreases because the blocks become too large to fit effectively into the cache. On most computational platforms that we have had experience with, this simple blocked multiplier runs at at least half the peak theoretical rate of the processor, as exemplified by the results in the figure. Further speedups can be found by rewriting the `low_level_multiply` routine so as to use register variables (as we have done and found up to 40% performance enhancements) or by using a hierarchical access pattern to memory which can provide better performance if the memory system has multiple levels of caching.

5.6.4 Parallelization

Once the computer code has been optimized to perform well on a single processor, the computation can be divided among multiple processors. We now discuss how this division can be achieved effectively.

The architectures of modern parallel supercomputers can generally be divided into two categories: shared-memory (SMP) versus distributed-memory (DMP) processors. In the SMP case, a set of identical processors share access to a very large repository of memory. The main advantages of shared memory are that the processors can access whatever data they need, and that, with judicious choice of algorithms, very little inter-processor communication is required. In addition, only small changes are required to parallelize a serial code: the computational task is divided among the processors, and each processor executes the original serial code on the portion of the data allotted to it. However, as the number of processors increases, the traffic for accessing the main memory banks increases dramatically and the performance will stop to scale well with the number of processors utilized. Nevertheless, many mid-

sized problems are well suited to SMPs and can take full advantage of the key features of SMP systems. Examples of such calculations can be found in [47, 18].

The largest of today's supercomputers have distributed memory: each processor has a private memory bank of moderate size to which it has exclusive access, and the processors communicate with each other by some message passing mechanism. The main advantages of DMP are scalability and heterogeneity, as the processors need not all be identical nor located in close physical proximity. However, the price paid is the necessity of an inter-processor communication mechanism and protocol. In addition, computer algorithms may have to be redesigned in order to minimize the required communication. Furthermore, a slow communication network between processors can adversely affect the overall performance.

We will describe, in outline, the strategies we employ for both SMP and DMP architectures, and we will continue to examine the case of the `column_bundle^column_bundle` matrix multiplication as a specific example. As our results for actual calculations will demonstrate, we only need to parallelize two main operations in the entire code, (a) the application of basis-dependent operators such as $\mathcal{I}$ or L to `column_bundle`s (which is trivial) and (b) the matrix products involving `column_bundle`s such as $Y_1^\dagger Y_2$ or $YU^{-1/2}$, to obtain excellent or highly satisfactory performance on SMP and DMP systems.

Shared-memory (SMP) architectures

Since all processors in an SMP computer have access to all the data in the computation, the parallelization of the basis-dependent operators is trivial. For example, the operation $\mathcal{I}C$ involves the application of $\mathcal{I}$ to each column of C separately, and the columns can be divided equally among the processors. This leads to near perfect parallelization as none of the processors read from or write to the same memory locations.

Based on the discussion of Section 5.6.2, for large problems, the most significant gains for parallelization involve the basis-independent matrix-products. Below, we focus on the `column_bundle^column_bundle` operation as a case study. For this

operation, it is impossible to distribute the task so that all processors always work on different memory segments. However, we divide up the work so that no two processors ever write to the same memory location: not only does this choice avoid possible errors due to the synchronizations of simultaneous memory writes, it also avoids performance degradation due to cache-flushing when memory is written to.

Consider the matrix product $M = Y_1^\dagger Y_2$, which we have implemented as a blocked matrix multiplication. Parallelization is achieved simply by assigning each processor to compute a subset of the output blocks. The main program spawns a set of subordinate tasks (termed threads) whose sole purpose is to compute their assigned output blocks and to write the results to memory. The main program waits for all threads to terminate before continuing onwards. Referring to Figure 5-8, this strategy corresponds to distributing the computation of the blocks m_{ij} among the processors, and since memory is shared, all processors have access to all of the data describing Y_1 and Y_2 at all times. For large problem sizes, the overhead in spawning and terminating the threads is far smaller than the work needed to compute the results, so the simplicity of this strategy does not sacrifice performance.

We now present the code which accomplishes this parallelized matrix product in order to highlight the ease with which such parallelizations can be performed. In the interest of brevity, we assume that the number of columns of Y_1 is a multiple of the number of threads launched. Parallelization is affected by assigning different threads to compute different rows of the result $M = Y_1^\dagger Y_2$.

```
friend matrix operator^(column_bundle &Y_1, column_bundle &Y_2)
{
    matrix M(Y_1.n_columns,Y_2.n_columns); // M=Y_1^Y_2 holds the result

    int n_threads = 8; // a reasonable number of threads

    int thread_id[n_threads];
    int i, start, n;
```

```
    for (i=0; i < n_threads; i++) {

        // The set of rows of M that this thread should compute
        n = Y_1.n_columns/n_threads;
        start = i*n;

        // Launch a thread that runs the routine calc_rows_of_M
        // and pass it the remaining arguments.
        thread_id[i] = launch_thread(calc_rows_of_M, Y_1, Y_2, M,
                                     start, n);
    }

    // Wait for the threads to terminate
    for (i=0; i < n_threads; i++)
        wait_for_thread(thread_id[i]);

    return M;
}
```

The routine `calc_rows_of_M(Y_1,Y_2,M,start,n)` computes rows `start` through `start+n-1` of `M`, where `M=Y_1^Y_2`. The routine's algorithm is identical to that of the blocked multiplier presented in the previous section. The only change required is to have the `ib` loop index start at `start` and end at `start+n-1`.

We parallelize other matrix multiplications involving `column_bundle`s analogously. In addition, we parallelize the application of the basis-dependent operators as discussed above. For a realistic study of performance and scaling, we consider a system of bulk silicon with 128 atoms in the unit cell. We use a plane-wave cutoff of 6 Hartrees (12 Ryd) which leads to a basis of size $N = 11797$. We use Kleinmann-Bylander [53] non-local pseudopotentials with p and d non-local projectors to eliminate the core states, and we have $n = 256$ bands of valence electrons. We sample the Brillouin

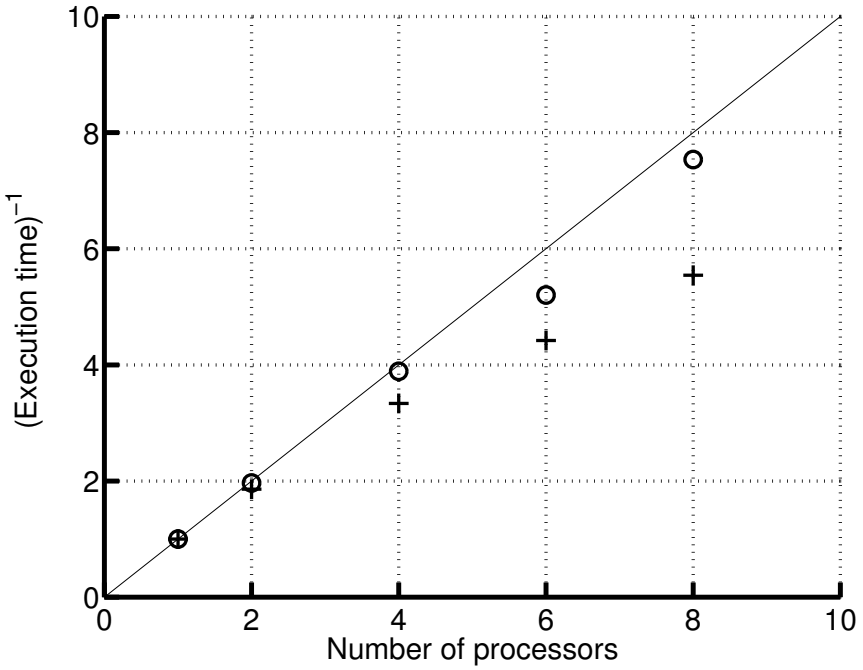

Figure 5-10: Scaling for SMP parallelization – Performance of our SMP parallelized plane-wave code for a 128 atom silicon system. We show the performance of the parallelized $M = Y_1^\dagger Y_2$ multiplications (circles) and the overall code (pluses) as a function of the number of processors . In both cases, performance has been normalized to the respective performance with a single processor. The straight line with slope of unity represents ideal scaling for perfect parallelization.

zone at $k = 0$. In Figure 5-10, we show a plot of the performance of the parallelized $M = Y_1^\dagger Y_2$ multiplication as well as the overall performance of the code for a single conjugate-gradient step as a function of the number of processors employed. The calculation was run on a Sun Ultra HPC 5000 with eight 167 MHz Ultrasparc 4 microprocessors and 512 MB of memory.

As the figure shows, the parallelized $M = Y_1^\dagger Y_2$ matrix multiplication shows near ideal scaling. There are a number of reasons for this behavior: (1) since different processors always write to different segments of memory, the algorithm does not suffer from memory write-contentions, (2) the inputs Y_1 and Y_2 are never modified so that memory reads are cached effectively, and (3) the problem size is large enough so that each processor's workload is much larger than the overhead required to distribute the work among the processors. This scaling is all the more impressive because when

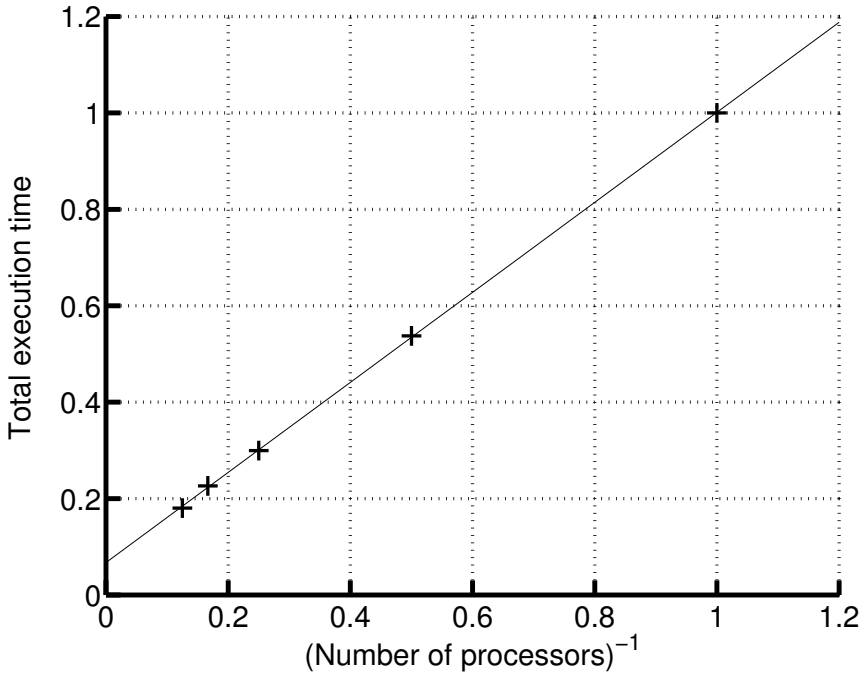

Figure 5-11: Amdahl's analysis of SMP scaling – Total execution time of the SMP parallelized plane-wave code for a 128 silicon atom system as a function of the reciprocal of the number of processors used. The line of least-squares fit to the data points is also shown. Execution times are normalized in units of the execution time for a single processor.

using eight processors, each processor performs the multiplications at 140 MFLOPS or at 83% of the processor clock rate.

The figure also displays the total performance of the code, which shows evidence of saturation. To understand this behavior in more detail, we model the overall execution time in accordance with Amdahl's law. We assume that there exists an intrinsic serial computation time T_0 which must be spent regardless of the number of processors available. In addition, there is an analogous parallel computation time $\tilde{T}$ which, however, is divided equally among all the processors. Thus the total execution time will be given by $T = T_0 + \tilde{T}/p$, where p is the number of processors. We show a plot of the total execution time versus $1/p$ in Figure 5-11, and the model shows an excellent fit to the available data. The vertical asymptote of the least-squares fit straight line is the extrapolated value of T_0, in this case approximately 10% of the

single processor or serial execution time. Thus the few operations that we have chosen to parallelize do in fact constitute the largest share of the computational burden and our parallelization strategy is quite effective.

When we reach the data point with eight processors, the total execution time is already within a factor of two of T_0, so that the total serial and parallel components have become comparable. Indeed, timing various portions of the code confirms this hypothesis: for example, with eight processors, the time needed to perform a parallel multiplication $M = Y_1^\dagger Y_2$ is equal to the time needed by the Sun high-performance LAPACK library to diagonalize the overlap matrix U (cf. Eq. (5.31)). With eight processors, the code has each processor *sustaining* an average of 134 MFLOPS or 80% of the processor clock rate. We are quite satisfied with this level of performance, but if more improvements are desired, the remaining serial portions of the code can be further optimized and parallelized.

Distributed-memory (DMP) architectures

Parallelization on DMP architectures requires careful thought regarding how the memory distribution and inter-process communications are to be implemented. The most memory-consuming computational objects are the `column_bundle`s, and for a large system, no single processor in a DMP computer can store the entire data structure in its local memory banks. Therefore, we parallelize the storage of `column_bundle`s by distributing equal numbers of the columns comprising a `column_bundle` among the processors.

Given this distribution by columns, the application of basis-dependent operators is unchanged from how it is done in a serial context: each processor applies the operator to the columns that are assigned to it, and perfect parallelization is achieved as no inter-processor communication is required. The large basis-independent matrix products, however, are more challenging to parallelize as they necessarily involve inter-processor communication.

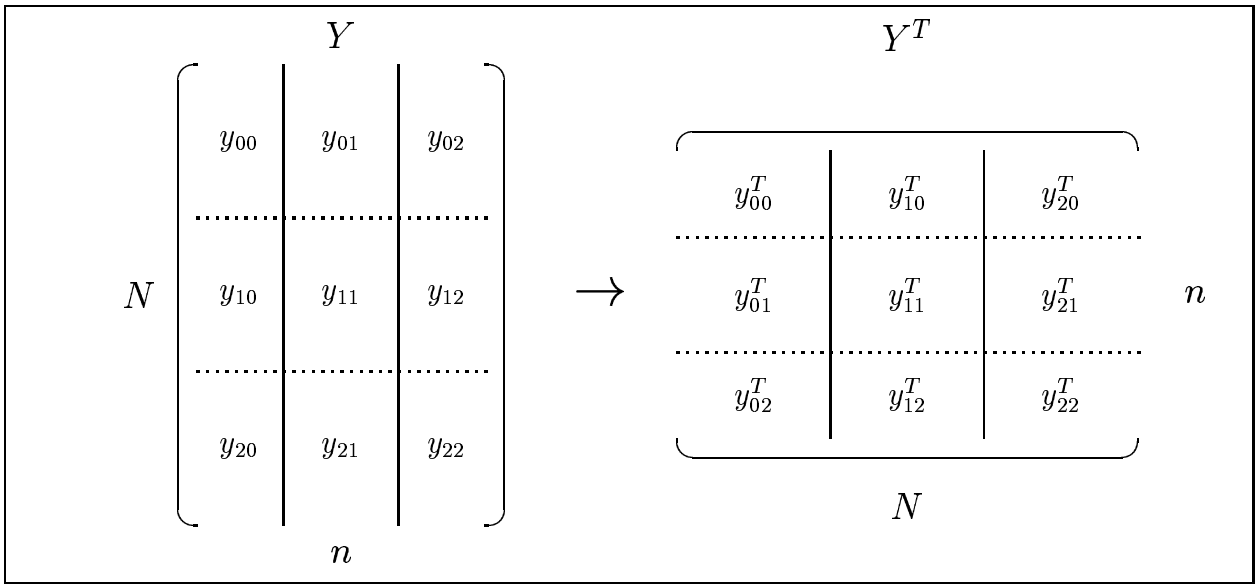

Figure 5-12: Transposition of distributed matrices – Schematic diagram showing how the transpose Y^T of the distributed `column_bundle` Y is obtained in a DMP calculation, which in this example has $p = 3$ processors. Across all the processors, Y is $N \times n$ and Y^T is $n \times N$. Solid vertical lines show the distribution of the columns of Y or Y^T among the processors, so that each processor stores a $N \times (n/p)$ block of Y and a $n \times (N/p)$ block of Y^T. The horizontal dotted lines show the division of Y and Y^T into blocks that must be communicated between processors to achieve the transposition: the block y_{ij} is sent from processor j to processor i. Processor i then transposes the block and stores it in the jith section of Y^T.

Consider again the product $M = Y_1^\dagger Y_2$, which in terms of components is given by

$$M_{ij} = \sum_k (Y_1)^*_{ki} (Y_2)_{kj} \, .$$

The column-wise parallel distribution of Y_1 and Y_2 means that the full range of the i and j indices is distributed among the processors while the full range of the k index is accessible locally by each processor. Since Y_1 and Y_2 are $N \times n$, each processor has a $N \times (n/p)$ block (i.e. n/p columns of length N) in its local memory, where p is the number of processors. Unfortunately, computing M using this column-wise distribution requires a great deal of inter-processor communication. For example, the processor computing the ith row of M requires knowledge of *all* the columns of Y_2, so that in total, the Nn data items describing Y_2 will have to be sent $p - 1$ times between all the processors.

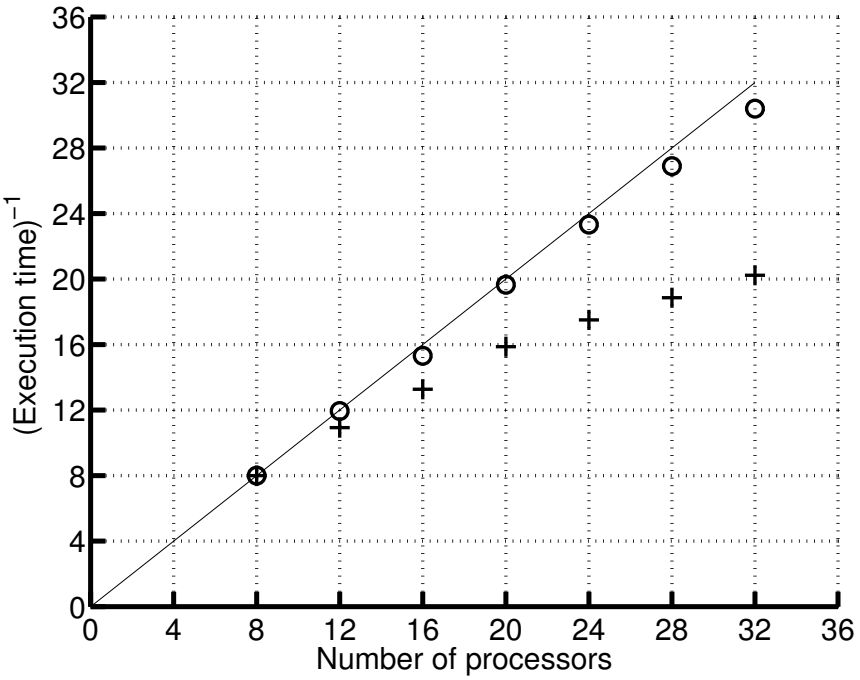

Figure 5-13: Scaling of DMP parallelization – Performance of the DMP parallelized plane-wave code for a 256 silicon atom system as a function of the number of processors for the parallelized $M = Y_1^\dagger Y_2$ multiplication (circles) and for the code overall (pluses). In both cases, the performance has been normalized to the respective performance with a single processor based on extrapolation from the eight processor run. The straight line with slope of unity represents ideal scaling for perfect parallelization.

A more efficient communication pattern can be developed that avoids this redundancy. Denoting the transpose of Y by Y^T, the matrix product can be rewritten as

$$M_{ij} = \sum_k (Y_1^T)^*_{ik} (Y_2^T)_{jk} \, .$$

Hence, if we first transpose Y_1 and Y_2, then the column-wise distribution of the transposed `column_bundle`s insures that the full range of the i and j indices can be accessed locally on each processor while the full range of the k index is now distributed among the processors. Figure 5-12 presents a schematic of how the transposition of a `column_bundle` Y is be accomplished: each processor has a $N \times (n/p)$ block of Y whose contents it sends to $p-1$ other processors as $p-1$ blocks of size $(N/p) \times (n/p)$. Next, each processor receives $p-1$ similar blocks sent to it by other processors,

transposes them, and stores them in the appropriate sections of Y^T. Each processor sends or receives only $Nn(p-1)/p^2$ data items, and a total of $Nn(p-1)/p$ data items are communicated among all the processors.

The computation of M in the transposed mode has the same operation count as in the non-transpose mode (i.e. $O(n^2N)$ operations), which when distributed across processors, amounts to $O(n^2N/p)$ operations per processor. Of course, we block the matrix multiplications on each processor to ensure the best performance. Finally, a global sum across all the processors' $n \times n$ resulting matrices is required to obtain the final answer M, and this requires $\log_2 p$ communications of size n^2 when using a binary tree.

Assuming that processors can simultaneously send and receive data across the network, the time required to perform the transpose is $O(nN/p)$, the time needed to perform the multiplications is $O(n^2N/p)$, and the time required to perform the final global sum is $O(n^2 \log_2 p)$. For large problem sizes, the time needed to perform the multiplications will always be larger than the time required for the communications by a factor of $\sim n$. Thus interprocessor communications are, in the end, never an issue for a sufficiently large physical system, and the computation will be perfectly parallelized in the asymptotic limit of an infinitely large system.

Figure 5-13 shows a plot of the performance of our DMP parallelized plane-wave code for a single conjugate-gradient step when run on a 256 atom silicon cell. The choice of pseudopotential, cutoff, and k-point sampling are the same as for the SMP calculation above. The basis set is of size $N = 23563$ and the system has $n = 512$ valence bands. The calculations were run on an IBM SP2 system with 336 nodes, and each node has four 332 MHz Power2 Architecture RISC System/6000 processors and 1.536 GB of memory. Again, we see excellent and near perfect scaling for the parallelized matrix multiplications, validating our claim that the transposition approach combined with the large system size provides for very good parallelization. With 32 processors, each parallelized multiplication runs at an average rate of 188 MFLOPS per processor (57% of the processor clock rate), which is impressive given the fact that the processors are busily communicating the data required for the transpositions

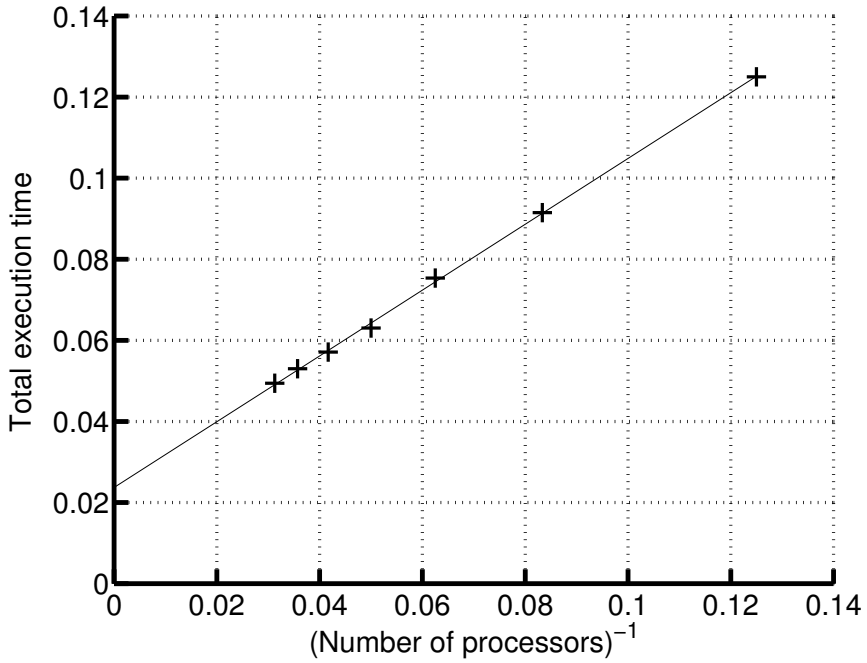

Figure 5-14: Amdahl's analysis of DMP scaling – Total execution time of the DMP parallelized plane wave code for a 256 silicon atom system as a function of the reciprocal of the number of processors utilized. The line of least-squares fit to the data points is shown as well. Execution times are normalized in units of the execution time for a single processor as extrapolated based on the eight processor run.

and global sums.

The plot also shows the saturation of the total performance with increasing number of processors. With eight processors, the overall performance translates into an average rate of 254 MFLOPS per processor (76% of the clock rate), whereas with 32 processors the rate has reduced to 160 MFLOPS per processor (48% of the clock rate). Clearly, the serial portions of the calculation begin to contribute significantly to the run time. Following the discussion of the SMP results above, Figure 5-14 presents a plot of the total execution time versus inverse number of processors. The extrapolated serial time T_0 in this case is only 2-3% of the total theoretical run time on a single processor, which shows how effectively the calculation has been parallelized. In addition, for 32 processors, the total run time is only two times larger that T_0, signalling the end of significant gains from the use of more processors.

Chapter 6

Locality of the Density Matrix in Metals, Semiconductors and Insulators

Density functional theory (DFT) [43, 58] describes many-body systems via a single-particle formalism and is the basis for modern, large-scale calculations in solid-state systems [77]. The one-particle density matrix $\hat{\rho} \equiv \sum_n |\psi_n\rangle f_n \langle\psi_n|$, which describes the state of a single-particle quantum system, is the key quantity needed for the computation of physical observables: total system energies, atomic forces, and phonons can all be computed directly from $\hat{\rho}$.

Remarkably, despite the de-localized nature of the single particle states $|\psi_n\rangle$, which may extend across an entire solid, the physics of the electronic states in a given region of a material is affected only by the local environment. Reflecting this, the force on an atom depends mostly on the positions of its nearest neighbors. This electronic localization is manifest in the "nearsightedness" [56] of $\hat{\rho}$: $\rho(\vec{r}, \vec{r}\,') \equiv \langle\vec{r}|\hat{\rho}|\vec{r}\,'\rangle \sim \exp(-\gamma|\vec{r} - \vec{r}\,'|)$ where $\gamma > 0$. This exponential decay has been verified numerically [34, 90] and analytically [54, 57, 81, 22, 32].

The locality of ρ not only is important for understanding the nearsightedness of effects arising from electronic structure but also has direct practical impact on DFT calculations. Recently, methods have been proposed [103, 66, 60, 21, 25, 89, 33, 75,

40, 74] that use ρ directly and exploit its locality. Computationally, these methods scale as $O(N)$, where N is the number of atoms in the simulation cell. However, their prefactors depend strongly on γ: some scale as N/γ^6 [60, 21, 40] and others as N/γ^3 [33]. Knowing how γ depends on the system under study is thus critical for carrying out such calculations. For a review of $O(N)$ methods, see [84].

Generally, solid-state systems have an underlying periodic structure. The introduction of localized defects [57] or surfaces [81] does not change the spatial range of ρ from that of the underlying periodic lattice. Thus, understanding the locality of ρ even for perfectly periodic systems is of direct relevance for realistic material studies. To date, the generic behavior of γ is poorly understood. For insulators in one dimension, Kohn has shown that $\gamma \propto \sqrt{-E_n}$ in the tight-binding limit where E_n is an atomic ionization energy [54]. Motivated by this, it has been assumed [55, 34, 75] and argued [6] that $\gamma \propto \sqrt{\Delta}$ in multiple dimensions and more general conditions, where Δ is the band gap. For metals, it has been assumed [34] and argued [6] that $\gamma \propto \sqrt{T}$, where T is the electronic temperature. However, the results in [6], which to date represent the only effort to determine γ generically, are based on the assumption that the inverse of the overlap matrix of a set of Gaussian orbitals decays in a Gaussian manner. On the contrary, the inverses of such overlap matrices decay only exponentially, and thus the behavior of γ warrants further study.

Here, we show that the behavior of γ is more complex than previously assumed. For insulators, γ is determined by the analytical behavior of the filled bands, which is determined by the strength of the periodic potential which, in turn, is most strongly reflected in the size of the direct band gaps. Indirect gaps, being more accidentally related to the strength of the potential, have a more haphazard relation to γ, which we do not consider here.

For insulating systems, we find that γ has the following asymptotic behavior as a function of the *direct* gap Δ, lattice constant a, and electron mass m,

$$\gamma \sim \begin{cases} a\Delta m/\hbar^2 & \text{for } a^2\Delta \to 0 \quad \text{(weak-binding)} \\ ??? & \text{for } a^2\Delta \to \infty \quad \text{(tight-binding)} \end{cases} .$$

The indeterminacy in the tight-binding limit results from the electronic states becoming atomic orbitals, and thus, γ depends on the details of the underlying atomic potential. Some systems (see below) exhibit $\gamma \propto \sqrt{\Delta}$, but this is *not* universal, as previously assumed.

One can, however, make a definitive statement in the weak-binding limit, which is of direct importance for small-gap systems such as those with weak pseudopotentials or gaps due to Jahn-Teller distortions. For example, semiconductors such as Si and GaAs have gaps that are significantly smaller than their band widths, and we expect them to fall into the weak-binding case. In Si and GaAs, we find $a^2 \Delta m/\hbar^2 \sim 4$ and ~ 2.5, respectively. Inspecting Figure 6-1, we see that for such values the behavior of γ is well in the weakly-bound limit.

For metals with a fixed number of electrons, we find

$$\gamma \sim \begin{cases} k_B T |\vec{\nabla}_k \varepsilon|^{-1} & \text{for } T \to 0 \quad \text{(quantum)} \\ \left[\frac{m}{\hbar^2} k_B T \ln(k_B T/\varepsilon_F)\right]^{\frac{1}{2}} & \text{for } T \to \infty \quad \text{(classical)} \end{cases},$$

in terms of the temperature T, the typical gradient of the band energy $\varepsilon_{\vec{k}}$ on the Fermi surface, and the Fermi energy ε_F.

The low-temperature result is of direct practical interest for calculations in metals. (This result was also found in a contemporaneously submitted publication [32].) We find behavior resembling that proposed in [34, 6], i.e. $\gamma \propto \sqrt{T}$, only at extremely high temperatures.

6.1 Definitions and key terminology

We now present analytical arguments that substantiate the above results and shed light on the physical mechanisms leading to and differentiating among the different limits. We consider periodic systems with lattice vectors of characteristic length a. We choose units such that $\hbar^2/m = 1$ and $k_B = 1$ in order to avoid cumbersome mathematical expressions; the results presented above are easily recovered by inserting $\hbar^2/m$ and k_B, as appropriate, in each step of the analysis below.

The Bloch wave-functions $\psi_{n\vec{k}}$, Wannier functions W_n, Fermi-Dirac fillings $f_{n\vec{k}}$, and density matrix $\rho(\vec{r},\vec{r}\,')$ are related via

$$\begin{aligned} W_n(\vec{r},\vec{R}\,) &= \Omega_B^{-1}\int d\vec{k}\, e^{-i\vec{k}\cdot\vec{R}}\psi_{n\vec{k}}(\vec{r}\,) \\ F_n(\vec{R}) &= \Omega_B^{-1}\int d\vec{k}\, e^{i\vec{k}\cdot\vec{R}} f_{n\vec{k}} \\ \rho_n(\vec{r},\vec{r}\,') &= \sum_{\vec{R}}\sum_{\vec{R}'} W_n(\vec{r},\vec{R}\,)\, F_n(\vec{R}-\vec{R}')\, W_n^*(\vec{r}\,',\vec{R}') \\ \rho(\vec{r},\vec{r}\,') &= \sum_n \rho_n(\vec{r},\vec{r}\,'). \end{aligned} \tag{6.1}$$

The integrals are over the first Brillouin zone with volume Ω_B. $\vec{R}$ ranges over the lattice vectors. $\rho_n(\vec{r},\vec{r}\,')$ is the density matrix of the nth band, and ρ is a simple sum over all ρ_n. Thus, we need only study the behavior of ρ_n for a given n. We analyze the behavior of γ for the two possible cases of practical interest, insulators at low temperature and metals at non-zero temperature.

6.2 Insulators ($T = 0$)

When the chemical potential μ falls in the energy gap, all fillings $f_{\vec{k}}$ are 1 or 0. A filled band with $f_{\vec{k}} = 1$ has $F(\vec{R}) = \delta_{\vec{R},0}$ so that ρ is simply

$$\rho(\vec{r},\vec{r}\,') = \sum_{\vec{R}} W(\vec{r},\vec{R}) W^*(\vec{r}\,',\vec{R}).$$

Wannier functions are exponentially localized [90, 54, 55, 57, 81, 23, 71, 70] and satisfy $W(\vec{r},\vec{R}\,) = W(\vec{r}-\vec{R},0)$. Thus, only a finite set of $\vec{R}$ contribute significantly to the above sum, and the decay rates of ρ and W are the same. Therefore, we need only determine γ for the Wannier functions.

As a concrete example and an initial orientation, we solve for γ exactly for the lowest band of a model one dimensional system for all binding strengths. We choose the periodic potential to be that of an array of attractive delta-functions of strength $V > 0$, $U(x) = -V\sum_n \delta(x-na)$. Following [54], we define $\mu(\varepsilon) \equiv \cos(a\sqrt{2\varepsilon}) - V\sin(a\sqrt{2\varepsilon})/\sqrt{2\varepsilon}$. The band structure is found by solving $\cos(ka) = \mu(\varepsilon_k)$ for real

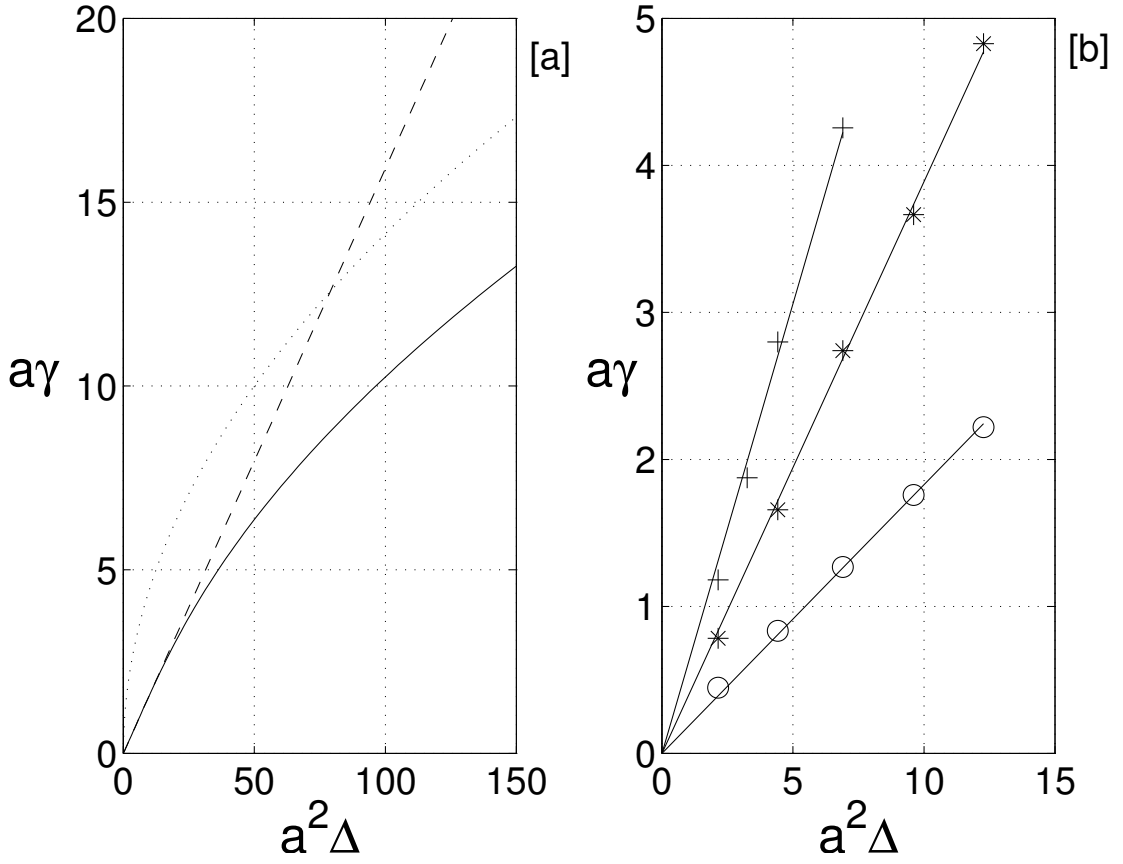

Figure 6-1: [a] $a\gamma$ versus $a^2\Delta$ for a periodic array of attractive delta potentials (solid curve). The dashed line is $a\gamma = a^2\Delta/(2\pi)$. The dotted curve is $a\gamma = a\sqrt{2\Delta}$, the leading asymptotic behavior of γ for large $a^2\Delta$. [b] $a\gamma$ versus $a^2\Delta$ for a cubic lattice of Gaussian potentials: γ in the [100] (circles), [110] (stars) and [111] (pluses) directions.

k. Focusing on the lowest band, we denote $\tilde{\varepsilon}$ as the value of ε where $\mu(\varepsilon)$ achieves its first minimum. Kohn [54] has shown that $\gamma = \cosh^{-1}|\mu(\tilde{\varepsilon})|$. We solve the above transcendental system numerically for different values of V and plot $a\gamma$ as a function of $a^2\Delta$ in Figure 6-1[a]. The behavior at small Δ is clearly linear, showing that $\gamma \sim a\Delta$ for a weak potential. The leading asymptotic behavior is $\gamma \sim \sqrt{\Delta}$ for a strong potential (i.e. large Δ).

In this case, the general notion that $\gamma \propto \sqrt{\Delta}$ is clearly incorrect for weak potentials, and it is natural to ask whether this result is peculiar to our simple model or whether it is more universal. As we now argue, for a weak potential, $\gamma \sim a\Delta$ is quite general, whereas in the case of a strong potential, the behavior of γ is not unique

and depends on the details of the atomic system underlying the periodic lattice. The crossover from weak to strong potential behavior should occur when Δ is of order of the band width. In the figure, this occurs for $\Delta \sim 5(\pi/a)^2$. We now analyze each case separately.

6.2.1 Weak-binding insulators

We wish to find γ in the limit of a weak periodic potential $U(\vec{r})$. Eqs. (6.1) show that the Wannier function is the Fourier transform of $\psi_{\vec{k}}$. Thus the range δk in $\vec{k}$-space where $\psi_{\vec{k}}$ has its strongest variations determines the spatial range of W. From basic Fourier analysis, $\gamma \sim \delta k$.

A simple heuristic argument shows that $\gamma \sim a\Delta$. Starting with a free electron description, a weak potential $U(\vec{r})$ causes the opening of a gap Δ at the edges of the Brillouin zone. The extent δk of the region about the zone-edges where $\varepsilon_{\vec{k}}$ deviates most appreciably from its free electron value is given by $\delta k^2/2m^* \sim \Delta$, where m^* is the effective mass at the zone edge. Standard treatments [5] show that for weak potentials $m^* \sim a^2\Delta$. Combining these results, we see that $\delta k \sim a\Delta$, whence $\gamma \sim a\Delta$.

This heuristic argument gives the desired result, but there are hidden assumptions. The argument is based solely on the behavior of the band structure $\varepsilon_{\vec{k}}$, whereas W is determined by the wave functions $\psi_{\vec{k}}$. One must be sure that $\varepsilon_{\vec{k}}$ and $\psi_{\vec{k}}$ vary over the same range δk. Thus, We present a more precise argument in terms of $\psi_{\vec{k}}$ alone.

Letting $\psi_{\vec{k}}(\vec{r}) = e^{i\vec{k}\cdot\vec{r}} u_{\vec{k}}(\vec{r})$, Eqs. (6.1) show that W is also a Fourier transform of $u_{\vec{k}}$. Away from the edges of the Brillouin zone, $u_{\vec{k}}$ is given perturbatively by

$$u_{\vec{k}}(\vec{r}) = 1 + \sum_{\vec{G}\neq 0} \frac{\langle \vec{G}|\hat{U}|\vec{0}\rangle \, e^{i\vec{G}\cdot\vec{r}}}{\left[k^2 - |\vec{k}+\vec{G}|^2\right]/2} + O(U^2), \tag{6.2}$$

where $\langle \vec{r}|\vec{G}\rangle = e^{i\vec{G}\cdot\vec{r}}$ and $\vec{G}$ is a reciprocal lattice vector: $u_{\vec{k}}$ is smooth and analytic in $\vec{k}$, and $u_{\vec{k}} \approx 1$. However, close to the zone edges, $|\vec{k}| \approx |\vec{k}+\vec{G}|$ and $u_{\vec{k}}$ deviates appreciably from unity in a region satisfying $[k^2 - |\vec{k}+\vec{G}|^2]/2 \leq V$, where V is the typical size of the matrix elements of U. Since $|\vec{G}| \sim 1/a$, this region has a width

$\delta k \sim aV \sim a\Delta$ and hence $\gamma \sim a\Delta$.

As a concrete example, we study a cubic lattice of attractive Gaussian potentials with rms width a/π. We vary the depth of the potential, and for each depth, we compute Δ and ρ by sampling the Brillouin zone on a cubic grid of size 40^3 and expanding $\psi_{\vec{k}}$ in plane waves with $|\vec{G}| \leq 12\frac{\pi}{a}$. Diagonalizing the resulting Hamiltonian gives the ground-state $\psi_{\vec{k}}$ from which we compute the density matrix. Sampling $\rho(0, \vec{r}\,')$ in the [100], [110] and [111] directions gives exponentially decaying envelopes upon which we perform linear fits on log plots to extract γ. Figure 6-1[b] shows our results, from which the behavior $\gamma \propto \Delta$ is evident.

Finally, for the case of a one-dimensional system with a weak periodic potential (i.e. $a^2\Delta \ll 1$), we show that generically $\gamma = \frac{a\Delta}{2\pi}$. As shown in [54], if we consider the band energy ε_k as an analytic function of the complex variable k, then γ is the imaginary part of the singularity of ε_k which lies closest to the real axis in the complex k-plane.

For k away from the edge of the Brillouin zone (where $\mathrm{Real}(k) = \frac{\pi}{a}$), perturbation theory yields

$$\varepsilon_k = \frac{k^2}{2} + \langle 0|\hat{U}|0\rangle + \sum_{G\neq 0} \frac{|\langle G|\hat{U}|0\rangle|^2}{[k^2 - (k+G)^2]/2} + O(U^3)\,. \tag{6.3}$$

The displayed terms and those subsumed into $O(U^3)$ all contain similar energy denominators, denominators which do not vanish as long as $|\mathrm{Real}(k)| < \frac{\pi}{a}$. Therefore, we conclude that ε_k is analytic in k away from the zone edge. However, for $k \to \frac{\pi}{a}$, the states k and $k - \frac{2\pi}{a}$ become degenerate, and the expansion of Eq. (6.3) is no longer valid. Standard degenerate perturbation theory [5] yields the following form for ε_k close to the zone edge,

$$\varepsilon_k = \frac{k^2 + \left(k - \frac{2\pi}{a}\right)^2}{4} \pm \sqrt{|V|^2 + \frac{1}{16}\left[k^2 - \left(k - \frac{2\pi}{a}\right)^2\right]^2} + O(U^2)\,. \tag{6.4}$$

Here, $V = \langle G = \frac{2\pi}{a}|\hat{U}|G = 0\rangle$. The terms subsumed into $O(U^2)$ stem from couplings to states other than k and $k - \frac{2\pi}{a}$ and are analytic in k because they contain non-

vanishing energy denominators. The band gap is at the zone edge and has value $\Delta = 2|V|$.

The expression of Eq. (6.4) has a branch cut for $k = \frac{\pi}{a} + iy$ where $|y| \geq \frac{a|V|}{\pi}$. Therefore, we have $\gamma = \frac{a|V|}{\pi}$ or, since $\Delta = 2|V|$, $\gamma = \frac{a\Delta}{2\pi}$.

6.2.2 Tight-binding insulators

The potential U is the periodic sum of an atomic potential V_{at}, $U(\vec{r}) = \sum_{\vec{R}} V_{at}(\vec{r} - \vec{R})$. For sufficiently strong V_{at}, system properties are determined by the atomic potential. The Wannier functions become atomic orbitals localized about the minima of V_{at}. Now, γ depends on the details of V_{at} and no single universal scaling can be found. To demonstrate the complexity and richness of this limit, we discuss briefly different examples of atomic potentials that lead to differing forms for γ. Note that in this atomic limit the lattice constant a is irrelevant in determining γ.

For the Coulomb potential, $V_{at}(\vec{r}) = -Ze^2/r$. In the limit $Ze^2 \to \infty$, we have hydrogenic states centered on the lattice sites with energies $E_n = -Z^2e^4/2n^2$ and Bohr radii $a_0 = n^2/Ze^2$. The gap Δ is an energy difference between atomic states, and so $\Delta \sim Z^2e^4$. Also, $\gamma \sim a_0^{-1}$, and so we conclude $\gamma \sim \sqrt{\Delta}$. More generally, for any atomic potential with only a single dimensionful parameter (e.g. Ze^2 above), dimensional analysis gives $\gamma \sim \sqrt{\Delta}$.

However, a similar analysis applied to a Gaussian potential, $V_{at}(\vec{r}) = -Ve^{-r^2/2\sigma^2}$, gives $\gamma \sim \Delta\sigma$, whereas, for a spherical well, $V_{at}(\vec{r}) = -V\theta(\sigma - r)$, we find that γ has *no* dependence on Δ. Thus in the tight-binding limit, it is difficult to make generic statements regarding γ.

6.3 Metals ($T > 0$)

In metals, the fillings $f_{\vec{k}}$ exhibit rapid variations in $\vec{k}$ across the Fermi surface and $F(\vec{R})$ in (6.1) becomes long-ranged. The Wannier functions, being independent of the fillings, remain exponentially localized, as discussed above. These facts combined with the structure of the sum in (6.1) imply that γ in this case is determined by

$F(\vec{R})$.

For an initial orientation, consider a band with a free electron-like form $\varepsilon_k = k^2/2$, whose spherical Fermi surface is contained inside the first Brillouin zone. For such a band, $F(\vec{R})$ is given by (6.1) with $f_k = (1 + \exp[(k^2/2 - \mu)/T])^{-1}$. Below, we will use the fact that the density matrix of a true free electron gas is proportional to this F, $\rho(\vec{r}, \vec{r}\,') \propto F(\vec{r} - \vec{r}\,')$.

Because the Fermi surface is contained within the first zone, we may extend the $\vec{k}$ integral for F to infinity. Changing to spherical coordinates, integrating by parts and using trigonometric identities yields

$$F(\vec{R}) \;=\; \frac{1}{2\Omega_B T}\left(\frac{1}{R}\frac{\partial}{\partial R}\right)^2 \int_{-\infty}^{\infty} \frac{dk\,\cos(kR)}{\cosh^2\left[(\frac{k^2}{2} - \mu)/2T\right]}\,. \tag{6.5}$$

When closing the integral in the upper complex k-plane, the relevant poles of the integrand are at $k = \tilde{k}_l \equiv \pm\sqrt{2\mu \pm 2i\pi T(2l+1)}$ for integers $l \geq 0$. The residues of these poles contain the factor $e^{i\tilde{k}_l R}$ which gives rise to oscillations due to the real part of $\tilde{k}_l$ and exponential decay due to its imaginary part.

When $T \to 0$, μ equals the Fermi energy $\mu = \varepsilon_F = k_F^2/2$ where $k_F = (3\pi^2 n)^{1/3}$ and n is the electron density. Thus we have $\tilde{k}_l \approx i\pi T(2l+1)/k_F \pm k_F$. As $R \to \infty$, the $l = 0$ contribution dominates, so that $\gamma = \pi T/k_F$.

As $T \to \infty$, the ideal gas result $\mu \approx T\ln(n\lambda_T^3)$ holds where $\lambda_T = \sqrt{2\pi/T}$ is the thermal de Broglie wavelength. In this limit, $\tilde{k}_l \approx \sqrt{2\mu}$ so that $\gamma = \mathrm{Im}\sqrt{2T\ln(n\lambda_T^3)} \sim \sqrt{T\ln(T/\varepsilon_F)}$.

Note that if we approximate the integrand of Eq. (6.5) by $e^{\mu/T - k^2/2T}$, which corresponds to using Maxwell-Boltzmann fillings, F will have a Gaussian form $F(\vec{R}) \propto e^{-TR^2/2}$. However, one can show that this approximation is only valid for small R. For large R, an exponential tail $e^{-\gamma R}$ remains where γ is as described above.

We now present separate arguments showing that these asymptotic forms for γ are correct for metals in general.

6.3.1 Metals as $T \to 0$

We first consider first $T = 0$. Bands below the Fermi level then have $f_{\vec{k}} = 1$ and $F(\vec{R}) = \delta_{\vec{R},0}$, and, as for the insulating case, their density matrices decay exponentially. However, for bands that cross the Fermi level, $f_{\vec{k}}$ jumps discontinuously from unity to zero wherever $\varepsilon_{\vec{k}} = \mu$. As is well known, the Fourier transform of a discontinuous function has algebraic falloff, and thus $F(\vec{R})|_{T=0} \propto |\vec{R}|^{-\eta}$ where $\eta > 0$. Such bands therefore dominate the decay of ρ as $T \to 0$.

At finite T, the fillings are $f_{\vec{k}} = (1 + e^y)^{-1}$ where $y = \left(\varepsilon_{\vec{k}} - \mu\right)/T$. As $T \to 0$, the fillings now go from unity to zero in a narrow region about the Fermi surface defined by vectors $\vec{k}_F$ satisfying $|\varepsilon_{\vec{k}_F} - \mu| \sim T$. To determine the width of this region, we approximate $f_{\vec{k}}$ about the Fermi vector $\vec{k}_F$ via $y \approx \vec{\nabla}\varepsilon \cdot (\vec{k} - \vec{k}_F)/T$. The width of the transition region and γ thus are given by $\gamma \sim \delta k \sim T/|\vec{\nabla}\varepsilon|$. This argument holds for any Fermi surface no matter how complex (metallic or semi-metallic) at sufficiently low T such that δk is smaller than the typical scale of the features of the Fermi surface.

As a further verification for the general case, we study a body-centered cubic lattice of tight-binding s-orbitals with lattice constant $a = 4.32$ Å, for which the Fermi surface is non-spherical. We choose the tight-binding matrix element so that the band structure has the free electron effective mass at $\vec{k} = 0$. Choosing μ to be $2/5$ of the way from the band minimum to the band maximum, we calculate $F(\vec{R})$ for various values of T. Sampling the Brillouin zone on a 200^3 grid, plotting $F(\vec{R})$ and finding exponentially decaying envelopes, we perform linear fits on a log plot and extract γ. Figure 6-2 shows that indeed $\gamma \propto T$ for $T \to 0$.

6.3.2 Metals as $T \to \infty$

Here the electron kinetic energy is much larger than the periodic potential so that we may approximate $\varepsilon_{\vec{k}} = k^2/2$: the system is a classical ideal gas and is not of interest for solid-state calculations. Our previously derived result for a free electron gas yields $\gamma \sim \sqrt{T \ln(T/\varepsilon_F)}$. Only in this limit do we find a result resembling that of [6].

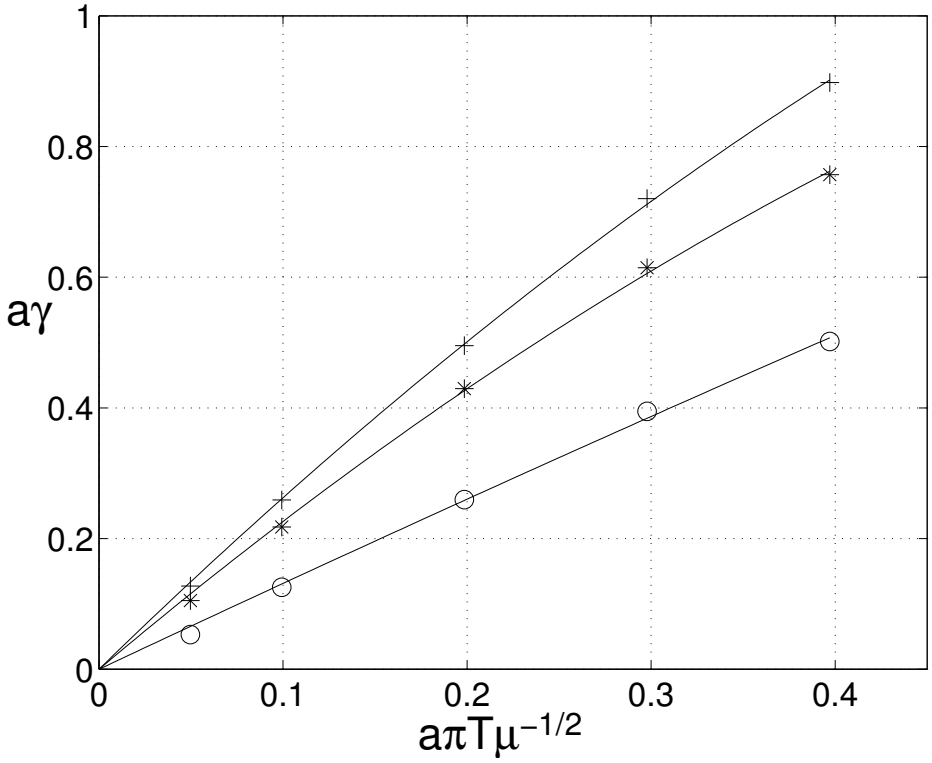

Figure 6-2: $a\gamma$ versus $a\pi T\mu^{-\frac{1}{2}}$ for a bcc lattice of tight-binding s-orbitals: γ in the [100] (circles), [110] (stars) and [111] (pluses) directions. μ is measured from the bottom of the band (see text).

Appendix A

Plane-wave implementation of the basis-dependent operators

A widely used basis-set for *ab initio* calculation has been the plane-wave basis set [77]. Plane waves are ideally suited for periodic calculations that model the bulk of a crystalline material. In addition, plane waves provide uniform spatial resolution throughout the entire simulation cell, and the results of the calculations can be converged easily by simply increasing the number of plane waves in the basis set. We will use plane waves as a concrete example of how the basis-dependent operators of Section 5.3.1 are to be implemented.

Given the lattice vectors of a periodic supercell, we compute the reciprocal lattice vectors and denote the points of the reciprocal lattice by the vectors $\{G\}$. Each element of our basis set $\{b_\alpha(r)\}$ will be a plane wave with vector G_α,

$$b_\alpha(r) = \frac{e^{iG_\alpha \cdot r}}{\sqrt{\Omega}} \ ,$$

where Ω is the real-space volume of the periodic cell. This basis is orthonormal, and the overlap operator $\mathcal{O}$ is the identity operator,

$$\mathcal{O} = I \, .$$

The integrals of the basis functions s are given by

$$s_\alpha = \sqrt{\Omega}\,\delta_{G_\alpha,0}\,,$$

so that the $\bar{\mathcal{O}}$ operator is given by

$$\bar{\mathcal{O}}_{\alpha\beta} = \delta_{\alpha,\beta} - \delta_{G_\alpha,0}\cdot\delta_{G_\beta,0}\,.$$

The Laplacian operator L is diagonal in this basis and is given by

$$L_{\alpha\beta} = -\|G_\alpha\|^2\,\delta_{\alpha\beta}\,.$$

The forward transform $\mathcal{I}$ is given by a Fourier transformation. Specifically, for a point p on the real-space grid, we have that

$$\mathcal{I}_{p\alpha} = \frac{e^{iG_\alpha\cdot p}}{\sqrt{\Omega}}.$$

Consider applying $\mathcal{I}$ to the column vector χ and evaluating the result at the point p:

$$(\mathcal{I}\chi)_p = \sum_\alpha \mathcal{I}_{p\alpha}\,\chi_\alpha = \frac{1}{\sqrt{\Omega}}\sum_\alpha e^{iG_\alpha\cdot p}\,\chi_\alpha\,.$$

This is the forward Fourier transform of χ.

For the case of plane waves, the inverse transform $\mathcal{J}$ can be chosen to be the inverse of $\mathcal{I}$, $\mathcal{J}=\mathcal{I}^{-1}$, as per the discussion of Section 5.3.1. It follows that

$$\mathcal{J} = \mathcal{I}^{-1} = \left(\frac{\Omega}{N}\right)\mathcal{I}^\dagger \quad \text{or} \quad \mathcal{J}_{\alpha p} = \frac{\sqrt{\Omega}}{N}e^{-iG_\alpha\cdot p}\,,$$

where N is the number of points in the real-space grid. Applying $\mathcal{J}$ to a column vector ξ, we have

$$(\mathcal{J}\xi)_\alpha = \frac{\sqrt{\Omega}}{N}\sum_p e^{-iG_\alpha\cdot p}\,\xi_p\,.$$

Thus $\mathcal{J}$ is a reverse Fourier transform. The operators $\mathcal{I}^\dagger$ and $\mathcal{J}^\dagger$ are also Fourier

transforms with appropriate scaling factors. Computationally, Fast Fourier Transforms [30, 17, 27] can be used to implement these operators most efficiently.

The last remaining basis-dependent item is the ionic potential V_{ion}. For periodic systems, this potential is a periodic sum of atomic potentials $V_{at}(r)$,

$$V_{ion}(r) = \sum_{R,I} V_{at}(r - R - r_I) \ ,$$

where R ranges over the real-space lattice sites, I ranges over the atoms in the unit cell, and r_I is the position of the Ith atom. Based on this, the elements of the vector V_{ion} of Eq. (5.13) are given by

$$(V_{ion})_\alpha = \frac{S(G_\alpha)\hat{V}_{at}(G_\alpha)}{\sqrt{\Omega}} \ ,$$

where the structure factor $S(q)$ is given by

$$S(q) \equiv \sum_I e^{-iq\cdot r_I} \ ,$$

and the Fourier transform of the atomic potential $\hat{V}_{at}$ is defined by

$$\hat{V}_{at}(q) \equiv \int d^3r \ e^{-iq\cdot r} \, V_{at}(r) \, .$$

Appendix B

Non-local potentials

In this section we show how non-local potentials can easily be incorporated into the DFT++ formalism. The total non-local potential operator $\hat{V}$ for a system is given by a sum over each atom's potential,

$$\hat{V} = \sum_I \hat{V}_I \,,$$

where $\hat{V}_I$ is the non-local potential of the Ith atom. The non-local energy is given by the expectation of $\hat{V}$ over all the occupied states $\{\psi_i\}$ with fillings f_i,

$$E_{nl} = \sum_i f_i \langle \psi_i | \hat{V} | \psi_i \rangle = \sum_I \sum_i f_i \langle \psi_i | \hat{V}_I | \psi_i \rangle \,.$$

Given the linearity of E_{nl} with respect to the atoms I, in our discussion below we will only consider the case of a single atom and will drop the index I. The results below can then be summed over the atoms to provide general expressions for multiple atoms.

Using the expansion coefficients $C_{\alpha i}$ of Eq. (5.7), we rewrite the non-local energy as

$$E_{nl} = \sum_i f_i \langle \psi_i | \hat{V} | \psi_i \rangle = \sum_{i,\alpha,\beta} f_i C^*_{\alpha i} \langle \alpha | \hat{V} | \beta \rangle C_{\beta i} = \mathrm{Tr}(VCFC^\dagger) = \mathrm{Tr}(VP) \,, \qquad \text{(B.1)}$$

where $|\alpha\rangle$ is the ket representing the basis function $b_\alpha(r)$, P is the single-particle density matrix of Eq. (5.20), the matrix F was defined as $F_{ij} = \delta_{ij} f_i$, and the matrix elements of the non-local potential are defined as

$$V_{\alpha\beta} \equiv \langle\alpha|\hat{V}|\beta\rangle = \int d^3r \int d^3r' \; b^*_\alpha(r)\, V(r,r')\, b_\beta(r')\,.$$

The matrix V clearly depends on both the basis set and the potential. The contribution of the non-local potential to the total Lagrangian of Eq. (5.29) is given simply by $\mathrm{Tr}(VP)$. Following the derivations of Eqs. (5.35) and (5.36), we see that the single-particle Hamiltonian H is modified only by the addition of V,

$$H = -\frac{1}{2}L + \mathcal{I}^\dagger[\mathrm{Diag}\; V_{sp}]\,\mathcal{I} + V\,. \tag{B.2}$$

We now write the potentials in separable form,

$$\hat{V} = \sum_{s,s'} |s\rangle M_{ss'} \langle s'|\;, \tag{B.3}$$

where s and s' range over the quantum states of the atom, $M_{ss'}$ are matrix elements specifying the details of the potential, and $|s\rangle$ is the ket describing the contribution of the sth quantum state to the potential. Typical choices of s are the traditional atomic state labels nlm and possibly the spin σ. Once we define the matrix elements $K_{\alpha s} \equiv \langle\alpha|s\rangle$, which are again basis-dependent, we find two equivalent forms for E_{nl},

$$\begin{aligned} E_{nl} &= \sum_{i,s,s'} f_i \langle\psi_i|s\rangle M_{ss'} \langle s'|\psi_i\rangle = \sum_{i,s,s',\alpha,\beta} f_i C^*_{\alpha i} K_{\alpha s} M_{ss'} K^*_{\beta s'} C_{\beta i} \\ &= \mathrm{Tr}\left[M(K^\dagger C)F(K^\dagger C)^\dagger\right] = \mathrm{Tr}\left[KMK^\dagger P\right]\;. \end{aligned} \tag{B.4}$$

The first form involving $K^\dagger C$ is most useful for efficient computation of the energy, and the second form is most useful for derivation of the gradient (cf. the discussion of Eqs. (5.28) and (5.29)). The energetic contribution to the Lagrangian is given by Eq. (B.4) and the contribution to the Hamiltonian H is $KMK^\dagger$, which replaces V in Eqs. (B.1) and (B.2).

A further specialization involves the popular case of Kleinmann-Bylander potentials [53] where the double sum in Eq. (B.3) is reduced to a single sum over s. Thus, the matrix M is a single number (i.e. a 1×1 matrix) which we denote by m_s, and the matrices K are in fact column vectors. The expression for the total non-local energy, this time including the sum over atoms I, is

$$E_{nl} = \sum_{I,s} m_{Is}(K_{Is}^\dagger C)F(K_{Is}^\dagger C)^\dagger = \mathrm{Tr}\left[\left(\sum_{I,s} m_{Is}K_{Is}K_{Is}^\dagger\right)P\right].$$

Unfortunately, this expression is not very efficient for evaluating the energy of a system with many atoms, as the sum on I is large but the matrix $K_{Is}^\dagger C$ only has a single row. This limits our ability to exploit the cache effectively (which only occurs for large matrix sizes).

We can rewrite the above energy expression so as to employ larger matrices and thus achieve greater computational efficiency. To do this, we define a diagonal matrix $\bar{M}_s$ that contains the m_{Is} values for all the atoms, $(\bar{M}_s)_{IJ} \equiv \delta_{IJ}m_{Is}$, and we define the matrices A_s via $(A_s)_{\alpha I} \equiv (K_{Is})_\alpha$. We then reorganize the previous expression for the non-local energy,

$$E_{nl} = \sum_s \mathrm{Tr}\left[\bar{M}_s(A_s^\dagger C)F(A_s^\dagger C)^\dagger\right] = \mathrm{Tr}\left[\left(\sum_s A_s\bar{M}_sA_s^\dagger\right)P\right].$$

If we have N basis functions, n quantum states $\{\psi_i\}$, and n_a atoms in the system, then C is $N \times n$ and A_s is $N \times n_a$. Thus, for large system sizes, the products $A_s^\dagger C$ involve matrices with large dimensions, and optimized matrix-multiplication routines function at peak efficiency.

Appendix C

Multiple k-points

We consider the generalization of our formalism to the case of multiple k-points, which arises in the study of periodic systems. In periodic cells, the wave functions satisfy Bloch's theorem and can be labeled by a quantum number k, a vector in the first Brillouin zone. The quantum states obey the Bloch condition

$$\psi_k(r+R) = e^{ik\cdot R}\psi_k(r)\,,$$

where R is a lattice vector of the periodic cell. This implies that $\psi_k(r) = e^{ik\cdot r}u_k(r)$ where u_k is a periodic function of r, $u_k(r+R) = u_k(r)$. We define the expansion coefficients C_k for the vector k as being those of the periodic function $u_k(r)$ and arrive at (cf. Eq. (5.7))

$$\psi_{km}(r) = e^{ik\cdot r}\sum_\alpha (C_k)_{\alpha m}\, b_\alpha(r)\,, \tag{C.1}$$

where the integer m labels the energy bands (i.e. different states at the same value of k). The Fermi-Dirac fillings may also have a k-dependence and are denoted as f_{km}.

In addition to k-vectors, calculations in periodic systems attach a weight w_k to the wave vector k. The rationale is that we require the integrals of physical functions over the Brillouin zone in order to compute the Lagrangian, energies, and other quantities. Ideally, we would like to integrate a function $g(k)$ over the Brillouin zone, but in a

practical computation this must be replaced by a discrete sum over a finite number of k-points with weights w_k. That is, we perform the following replacement

$$\int d^3k \; g(k) \to \sum_k w_k \, g(k) \, .$$

The required generalizations of the DFT++ formalism are straightforward and are outlined below. The density matrices (cf. Eq. (5.20)) now depend on k-points

$$P_k = w_k C_k F_k C_k^\dagger \, ,$$

where the filling matrix is $(F_k)_{mm'} = \delta_{m,m'} f_{km}$, and the expansion coefficient matrices C_k are given by Eq. (C.1). We define the total density matrix P through

$$P = \sum_k P_k \, .$$

The electron density n (cf. Eq. (5.21)) is given by

$$n = \sum_k \mathrm{diag} \left(\mathcal{I} P_k \mathcal{I}^\dagger \right) \, .$$

The electron-ion, exchange-correlation, and electron-Hartree energies depend only on n, and provided the above k-dependent expression for n is used, these contributions require no further modification from the forms already given in Eqs. (5.23), (5.24), and (5.27) respectively.

The only change required to the basis-dependent operators involves the use of the Laplacian for computing the kinetic energy. The proper generalization is to define k-dependent Laplacian matrices L_k through

$$(L_k)_{\alpha\beta} \equiv \int d^3r \; \left[e^{ik\cdot r} b_\alpha(r) \right]^* \, \nabla^2 \, \left[e^{ik\cdot r} b_\beta(r) \right] \, .$$

This immediately leads to the following expression for the kinetic energy:

$$T = -\frac{1}{2}\sum_k \mathrm{Tr}\left(L_k P_k\right) .$$

We still require the operator L as defined in Eq. (5.9) for operations involving the Hartree field ϕ and the Poisson equation. The L operator is L_k evaluated at $k = 0$.

The generalization of non-local potentials (Appendix B) to multiple k-points is also straightforward. The energy expression of Eq. (B.1) generalizes to

$$E_{nl} = \sum_k \mathrm{Tr}(V P_k) .$$

Having completed the specification of the Lagrangian with multiple k-points, the generalizations required for the orthonormality condition and the expressions for the derivatives of the Lagrangian follow immediately. We introduce overlap matrices U_k and unconstrained variables Y_k (cf. Eqs. (5.31) and (5.33)),

$$U_k = Y_k^\dagger \mathcal{O} Y_k \quad \text{and} \quad C_k = Y_k U_k^{-1/2} ,$$

where for simplicity we have set all subspace-rotation matrices to identity, $V_k = I$. The differential of the Lagrangian takes the form (cf. Eq. (5.35))

$$d\mathcal{L}_{LDA} = \sum_k \mathrm{Tr}(H_k\, dP_k) .$$

The Hamiltonians H_k depend on k only through the kinetic operators L_k,

$$H_k = -\frac{1}{2}L_k + \mathcal{I}^\dagger \left[\mathrm{Diag}\ V_{sp}\right]\mathcal{I} \ + V .$$

The expression for the single-particle potential V_{sp} is unmodified from that of Eq. (5.36) as it only depends on the total electron density n. The term V is to be added only if non-local potentials are employed (see Appendix B).

The expressions of Eq. (5.37) for the derivative of the Lagrangian also generalize

in the following straightforward way,

$$
\begin{aligned}
d\mathcal{L}_{LDA} &= 2\ \mathrm{Re} \sum_k \mathrm{Tr}\left[dY_k^\dagger \left(\frac{\partial \mathcal{L}_{LDA}}{\partial Y_k^\dagger} \right) \right], \text{ where} \\
\left(\frac{\partial \mathcal{L}_{LDA}}{\partial Y_k^\dagger} \right) &\equiv w_k \left\{ \left(I - \mathcal{O} C_k C_k^\dagger \right) H_k C_k F_k U_k^{-1/2} + \mathcal{O} C_k Q_k([\tilde{H}_k, F_k]) \right\}, \text{ and} \\
\tilde{H}_k &\equiv C_k^\dagger H_k C_k \ ,
\end{aligned}
$$

where Q_k is the natural generalization of the Q operator which uses the eigenvalues and eigenvectors of U_k (Appendix E).

Appendix D

Complete LDA code with k-points and non-local potentials

In this section, we summarize and gather together the expressions for the LDA Lagrangian and its derivatives in the DFT++ formalism for a system with k-points and non-local potentials. This type of system provides the natural starting point for studying bulk systems and the properties of defects in bulk-like systems [77].

As we have emphasized previously, it is sufficient for us to display the formulae for the Lagrangian and its derivatives because formulae in the DFT++ language specify all the operations that must be performed and translate directly into computer code. (See Section 5.6.) Given the Lagrangian and its derivatives, we can use a variety of methods to achieve self-consistency. (See Section 5.5.)

We follow the notation of Appendix C and refer the reader to it for relevant details and definitions. The point we wish to emphasize is the compactness of the formalism and how it allows us to specify an entire quantum-mechanical Lagrangian or energy function in a few lines of algebra which explicitly show the operations required for the computation. We specialize to the case of Kleinmann-Bylander [53] non-local potentials (Appendix B).

$$
\begin{aligned}
\mathcal{L}_{LDA} &= -\frac{1}{2}\sum_k w_k \,\mathrm{Tr}\left(F_k C_k^\dagger L_k C_k\right) + (\mathcal{J}n)^\dagger\left[V_{ion} + \mathcal{O}\mathcal{J}\epsilon_{xc}(n) - \bar{\mathcal{O}}\phi\right] \\
&\quad + \sum_k w_k \sum_s \mathrm{Tr}\left(\bar{M}_s (A_s^\dagger C_k) F_k (A_s^\dagger C_k)^\dagger\right) + \frac{1}{8\pi}\phi^\dagger L\phi\,, \\
\frac{\partial \mathcal{L}_{LDA}}{\partial Y_k^\dagger} &= w_k\left\{\left(I - \mathcal{O}C_k C_k^\dagger\right) H_k C_k F_k U_k^{-1/2} + \mathcal{O}C_k Q_k([C_k^\dagger H_k C_k, F_k])\right\}\,, \\
\frac{\partial \mathcal{L}_{LDA}}{\partial \phi^\dagger} &= -\frac{1}{2}\bar{\mathcal{O}}\mathcal{J}n + \frac{1}{8\pi}L\phi\,, \\
H_k &= -\frac{1}{2}L_k + \mathcal{I}^\dagger\left[\mathrm{Diag}\; V_{sp}\right]\mathcal{I} + \sum_s A_s \bar{M}_s A_s^\dagger\,, \\
V_{sp} &= \mathcal{J}^\dagger V_{ion} + \mathcal{J}^\dagger\mathcal{O}\mathcal{J}\epsilon_{xc}(n) + [\mathrm{Diag}\;\epsilon'_{xc}(n)]\mathcal{J}^\dagger\mathcal{O}\mathcal{J}n - \mathcal{J}^\dagger\bar{\mathcal{O}}\phi\,.
\end{aligned}
$$

Appendix E

The Q operator

In this appendix, we define the Q operator which appears in expressions for the derivative of the Lagrangian, e.g. in Eq. (5.37). The formal properties satisfied by Q are also presented, properties used in the derivation of the expression for the derivative based on the connection between Q and the differential of the matrix $U^{1/2}$. (See the derivation starting from Eq. (5.36) and resulting in Eq. (5.37) in Section 5.3.5.)

We start with the Hermitian matrix U. Let μ be a diagonal matrix with the eigenvalues of U on its diagonal, and let W be the unitary matrix of eigenvectors of U. Thus, the following relations hold: $U = W\mu W^\dagger$, $W^\dagger W = WW^\dagger = I$, and $UW = W\mu$.

Consider the differential of the matrix U. The Leibniz rule results in

$$dU = dW\,\mu\,W^\dagger + W\,d\mu\,W^\dagger + W\,\mu\,dW^\dagger\,.$$

Using the unitarity of W, we have

$$W^\dagger dUW = W^\dagger dW\,\mu + \mu\;dW^\dagger W + d\mu\,.$$

Differentiating the relation $W^\dagger W = I$, we have that $dW^\dagger W + W^\dagger dW = 0$ or $dW^\dagger W = -W^\dagger dW$. Substituting this above, we arrive at the relation

$$W^\dagger dUW = [W^\dagger dW, \mu] + d\mu\,. \tag{E.1}$$

This equation describes how differentials of the eigenvalues and eigenvectors of U are related to the differential of U, and it is simply a convenient matrix-based expression of the results of first-order perturbation theory familiar from elementary quantum mechanics. To see this equivalence, we first examine the diagonal elements of Eq. (E.1) and find

$$d\mu_n = (W^\dagger dU W)_{nn}\,, \tag{E.2}$$

the familiar expression for the first order shift of the eigenvalue μ_n. Considering off diagonal matrix elements of Eq. (E.1) leads to

$$(W^\dagger dW)_{nm} = \frac{(W^\dagger dU W)_{nm}}{\mu_m - \mu_n} \quad \text{for } n \neq m\,, \tag{E.3}$$

which is the expression for the first order shift of the mth wave function projected on the nth unperturbed wave function.

Next, we consider $f(U)$, an arbitrary analytic function of U. Using the eigenbasis of U, we can write $f(U) = W f(\mu) W^\dagger$ where by $f(\mu)$ we mean the diagonal matrix obtained by applying f to each diagonal entry of μ separately. Following the same logic as above, the differential of $f(U)$ satisfies

$$W^\dagger d[f(U)]W = [W^\dagger dW, f(\mu)] + f'(\mu) d\mu\,.$$

Computing matrix elements of the above equation and using Eqs. (E.2) and (E.3), we arrive at the general result

$$(W^\dagger d[f(U)]W)_{nm} = (W^\dagger dU W)_{nm} \cdot \begin{cases} f'(\mu_n) & \text{if } m = n \\ \left[\frac{f(\mu_m) - f(\mu_n)}{\mu_m - \mu_n}\right] & \text{if } m \neq n \end{cases}\,. \tag{E.4}$$

We now apply this result to the case where $f(U) = U^{1/2}$. This means that $f(\mu_n) = \sqrt{\mu_n}$ and that $f'(\mu_n) = 1/(2\sqrt{\mu_n})$ in Eq. (E.4). By employing the algebraic identity $(\sqrt{x} - \sqrt{y})/(x - y) = 1/(\sqrt{x} + \sqrt{y})$, we arrive at the expression

$$(W^\dagger d[U^{1/2}]W)_{nm} = \frac{(W^\dagger dU W)_{nm}}{\sqrt{\mu_n} + \sqrt{\mu_m}} = (W^\dagger Q(dU) W)_{nm}\,, \tag{E.5}$$

where we define the operator $Q(A)$ for an arbitrary matrix A to be

$$(W^\dagger Q(A) W)_{nm} \equiv \frac{(W^\dagger A W)_{nm}}{\sqrt{\mu_n} + \sqrt{\mu_m}} \,. \tag{E.6}$$

From this definition of Q, it is easy to prove that the following identities are satisfied for arbitrary matrices A and B and arbitrary power p:

$$\begin{aligned}
Q(dU) &= d[U^{1/2}] \\
\mathrm{Tr}\,(Q(A)B) &= \mathrm{Tr}\,(AQ(B)) \\
\mathrm{Tr}\,(Q(U^p A)B) &= \mathrm{Tr}\,(U^p Q(A)B) \\
\mathrm{Tr}\,(Q(AU^p)B) &= \mathrm{Tr}\,(Q(A)U^p B) \\
A &= Q(A)U^{1/2} + U^{1/2}Q(A)\,.
\end{aligned} \tag{E.7}$$

These are the identities used in the derivation of the expression for the derivative of the Lagrangian with respect to Y in Section 5.3.5.

Bibliography

[1] H. C. Anderson. *Journal of Chemical Physics*, 72:2384, 1980.

[2] T. A. Arias. *Reviews of Modern Physics*, 71:267, 1999.

[3] T. A. Arias and T. D. Engeness. Beyond wavelets: Exactness theorems and algorithms for physical calculations. In D. Landau, S. P. Lewis, and H. B. Schuttler, editors, *Computer Simulations in Condensend Matter Physics*, volume 12. Springer Verlag, Heidelberg, 1999.

[4] T. A. Arias, M. C. Payne, and J. D. Joannopoulos. *Physical Review Letters*, 69:1077, 1992.

[5] N. W. Ashcroft and N. D. Mermin. *Solid State Physics*. Harcourt Brace College Publishers, orlando edition, 1976.

[6] R. Baer and M. Head-Gordon. *Physical Review Letters*, 79:3962, 1997.

[7] Z. S. Basinski and M. S. Duesbery. Dislocation modelling of physical systems. Pergamon, Oxford, 1980.

[8] R. P. Bell. *Transactions of the Faraday Society*, 27:797, 1931.

[9] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, and J. Hermans. *Jerusalem Symposia on Quantum Chemistry and Biochemistry*. Reidel, 1981.

[10] G. Beylkin, R. R. Coifman, and V. Rokhlin. *Communications in Pure and Applied Mathematics*, 44:141, 1992.

[11] J. R. K. et al. Bigger. *Physical Review Letters*, 69:2224, 1992.

[12] M. Born. *Z. Phys.*, 1:45, 1920.

[13] F. Buda, J. Kohanoff, and M. Parrinello. *Physical Review Letters*, 69:1272, 1992.

[14] L. T. Canham. *Applied Physics Letters*, 57:1046, 1990.

[15] D. M. Ceperley and B. J. Alder. *Physical Review Letters*, 45:566, 1980.

[16] T. Chandrapatha and A. Belegundu. *Introduction to Finite Elements in Engineering*. Prentice Hall, upper saddle river, nj edition, 1997.

[17] J. W. Cooley and J. W. Tukey. *Mathematics of Computation*, 19:297, 1965.

[18] Gabor Csanyi, Sohrab Ismail-Beigi, and T. A. Arias. Paramagnetic structure of the soliton of the 30^o partial dislocation in silicon. *Physical Review Letters*, 80:3984, 1998.

[19] Johnson D. D. *Physical Review B*, 38:12807, 1988.

[20] E. R. Davidson and D. Feller. *Chemical Review*, 86:681, 1986.

[21] M. S. Daw. *Physical Review B*, 47:10895, 1993.

[22] J. des Cloizeaux. *Physical Review*, 135:A658, 1964.

[23] J. des Cloizeaux. *Physical Review*, 135:A698, 1964.

[24] G. Dolling. *Inelastic Scattering of Neutrons in Solids and Liquids*, volume II. International Atomic Energy Agency, Vienna, Austria, 1963.

[25] D. A. Drabold and O. F. Sankey. *Physical Review Letters*, 70:3631, 1993.

[26] M. S. Duesbery. In *Dislocations 1984*, page 131, Paris, 1984. CNRS.

[27] P. Duhamel and M. Vetterli. *Signal Processing*, 19:259, 1990.

[28] D. J. EagleSham, A. E. White, L. C. Feldman, N. Noriya, and D. C. Jacobson. *Physical Review Letters*, 70:1643, 1993.

[29] I. Fells and E. A. Moelwyn-Hughes. *Journal of the Chemical Society*, 2:1326, 1958.

[30] M. Frigo and S. G. Johnson. Fftw: An adaptive software architecture for the fft. In *Proceedings ICASSP*, volume 3, page 1381, 1998. Software available at http://www.fftw.org.

[31] J. Fritsch and P. Pavone. *Surface Science*, 344:159, 1995.

[32] S. Goedecker. *Physical Review B*, 58:3501, 1998.

[33] S. Goedecker and L. Colombo. *Physical Review Letters*, 73:122, 1994.

[34] S. Goedecker and M. Teter. *Physical Review B*, 51:9455, 1995. Used to be GT.

[35] S. Goedecker and C. J. Umigar. *Physical Review A*, 55:1765, 1997.

[36] X. Gonze, D. C. Allan, and M. P. Teter. *Physical Review Letters*, 68:3603, 1992.

[37] O. Gunnarson and B. I. Lundqvist. *Physical Review B*, 13:4274, 1976.

[38] A. K. Head. *Phys. Stat. Solidi*, 5:51, 1964.

[39] A. K. Head. *Phys. Stat. Solidi*, 6:461, 1964.

[40] E. Hernandez and M. J. Gillan. *Physical Review B*, 51:10157, 1995.

[41] P. B. Hirsch. In *Proceedings of the 5th International Conference on Crystallography*, page 139, 1960.

[42] J. P. Hirth and J. Lothe. *Theory of Dislocations.* John Wiley and Sons, Inc., New York, 1982.

[43] P. Hohenberg and W. Kohn. *Physical Review*, 136:864, 1964.

[44] M. S. Hybertsen. *Physical Review Letters*, 72:1514, 1994.

[45] M. S. Hybertsen and M. Needels. *Physical Review B*, 48:4608, 1993.

[46] J. Ihm, D. H. Lee, J. D. Joannopoulos, and J. J. Xiong. *Physical Review Letters*, 51:1872, 1983.

[47] Sohrab Ismail-Beigi and T. A. Arias. Edge-driven transition in surface structure of nanoscale silicon. *Physical Review B*, 57:11923, 1998.

[48] Sohrab Ismail-Beigi and T. A. Arias. Locality of the density matrix in metals, semiconductors, and insulators. *Physical Review Letters*, 82:2127, 1999.

[49] Sohrab Ismail-Beigi and T. A. Arias. New algebraic formulation of ab initio calculation. In press as an invited contribution for special issue of *Computer Physics Communications* on Parralel Computing in Chemical Physics, August 1999.

[50] Sohrab Ismail-Beigi and T. A. Arias. Ab initio study of screw dislocations in Mo and Ta: A new picture of plasticity in bcc transition metals. *Physical Review Letters*, 84:1499, February 2000.

[51] J. G. Kirkwood. *Journal of Chemical Physics*, 2(7):351, 1934.

[52] C. Kittel. *Introduction to Solid State Physics*. John Wiley and Sons, Inc., New York, 6th edition, 1986.

[53] L. Kleinmann and D. M. Bylander. *Physical Review Letters*, 48:1425, 1982.

[54] W. Kohn. *Physical Review*, 115:809, 1959.

[55] W. Kohn. *Chemical Physics Letters*, 208:167, 1993.

[56] W. Kohn. *Physical Review Letters*, 76:3168, 1996. used to be K.

[57] W. Kohn and R. J. Onffroy. *Physical Review B*, 8:2485, 1973.

[58] Walter Kohn and L. J. Sham. *Physical Review*, 140:A1133, 1965.

[59] M. Kohyama. *Journal of Physics: Codensed Matter*, 3:2193, 1991.

[60] X.-P. Li, R. W. Nunes, and D. Vanderbilt. *Physical Review B*, 47:10891, 1993.

[61] D. R. Lide, editor. *Handbook of Chemistry and Physics.* CRC, Boca Raton, FL, 77 edition, 1996.

[62] R. Lippert, T. A. Arias, and A. Edelman. *Journal of Computational Physics*, 140:270, 1998.

[63] P. A. Marrone, T. A. Arias, W. A. Peters, and J. W. Tester. *Journal of Physical Chemistry A*, 102:7013, 1998.

[64] P. A. Marrone, P. M. Gschwend, K. C. Swallow, W. A. Peters, and J. W. Tester. *Journal of Supercritical Fluids*, 12:239, 1998.

[65] N. Marzari, D. Vanderbilt, and M. C. Payne. *Physical Review Letters*, 79:1337, 1997.

[66] F. Mauri, G. Galli, and R. Car. *Physical Review B*, 47:9973, 1993.

[67] A. L. McClellan. *Tables of Experimental Dipole Moments.* W. H. Freeman and Co., San Francisco, 1963.

[68] S. Miertus, E. Scrocco, and J. Tomasi. *Chemical Physics*, 55:117, 1981.

[69] J. A. et al. Moriarty. *Journal of Engineering Materials and Technology*, 121:120, 1999.

[70] A. Nenciu and G. Nenciu. *Physical Review B*, 47:10112, 1993.

[71] G. Nenciu. *Communications in Math. Phy.*, 91:81, 1983.

[72] S. J. Nosé. *Journal of Chemical Physics*, 81:511, 1984.

[73] S. Obara and A. Saika. *Journal of Chemical Physics*, 84:3963, 1986.

[74] P. Ordejon, E. Artacho, and S. M. Soler. *Physical Review B*, 53:10441, 1996.

[75] P. Ordejon, D. A. Drabold, R. M. Martin, and M. P. Gumback. *Physical Review B*, 51:1456, 1995.

[76] R. G. Parr and W. Yang. *Density-Functional Theory of Atoms and Molecules.* Oxford University Press, New York, 1989.

[77] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos. *Reviews of Modern Physics*, 64:1045, 1992.

[78] J. Perdew and A. Zunger. *Physical Review B*, 23:5048, 1981.

[79] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C.* Cambridge University, Cambridge, 1988.

[80] A. Rappe, K. Rabe, E. Kaxiras, and J. D. Joannopoulos. *Physical Review B*, 41:2127, 1990.

[81] J. J. Rehr and W. Kohn. *Physical Review B*, 10:448, 1974.

[82] S. F. Rice, R. R. Steeper, and C. A. LaJeunesse. Destruction of representative navy wastes using supercritical water oxidation. Technical Report Sandia Report SAND94-8203 UC-402, Sandia National Laboratories, Livermore, CA, 1993.

[83] N. Roberts and R. J. Needs. *Surface Science*, 236:112, 1990.

[84] Goedecker. S. *Reviews of Modern Physics*, jan 1999.

[85] D. Salvatierra, J. D. Taylor, P. A. Marrone, and J. W. Tester. *Ind. Eng. Chem. Res.*, 38:4169, 1999.

[86] S. Sawada. *Vacuum*, 41:612, 1990.

[87] E. Schmid. In *Proceedings of the International Congress of Applied Mechanics*, page 342, 1942.

[88] W. Sigle. *Philosophical Magazine A*, 79:1009, 1999.

[89] E. B. Stechel, A. R. Williams, and P. J. Feibelman. *Physical Review B*, 49:10088, 1994.

[90] U. Stephan and D. A. Drabold. *Physical Review B*, 57:6391, 1998.

[91] T. B. Thomason, G. T. Hong, K. C. Swallow, and W. R. Killilea. The modar supercritical water oxidation process. In H. M. Freeman, editor, *Innovative Hazardous Waste Treatment Technology Series*, volume 1, page 31. Technomic Publishing Co., Lancaster, PA, 1990.

[92] J. Tomasi and M. Persico. *Chemical Reviews*, 94:2027–2094, 1994.

[93] J. Travis. Building bridges to the nanoworld. *Science*, 263:1703, 1994.

[94] M. Uematsu and E. U. Franck. *Journal of Physical and Chemical Reference Data*, 9(4):1291, 1980.

[95] B. P. Van der Gaag and A. Scherer. *Applied Physics Letters*, 56:481, 1990.

[96] V. Vitek. *Cryst. Lattice Defects*, 5, 1974.

[97] P. Wesseling. *An Introduction to Multigrid Methods.* Wiley, Chichester, New York, 1992.

[98] J. E. Whiteman. *The Mathematics of Finite Elements and Applications: Highlights 1996.* John Wiley and Sons, New York, 1997.

[99] S. Wilson. *Advances in Chemical Physics*, 67:439, 1987.

[100] C. Woodard, S. Kajihara, and L. H. Yang. *Physical Review B*, 57:13459, 1998.

[101] W. Xu and J. A. Moriarty. *Physical Review B*, 54:6941, 1996.

[102] W. Xu and J. A. Moriarty. *Computational Material Science*, 9:348, 1998.

[103] W. Yang. *Physical Review Letters*, 66:1438, 1991.

[104] H. Yorikawa, H. Uchida, and S. Muramatsu. *Journal of Applied Physics*, 79:3619, 1996.

[105] A. Zunger and W. Lin-Wang. *Applied Surface Science*, 102:350, 1996.