CONTENTS

An Occupied Subspace Optimization for Linear Scaling in Large-Scale *Ab Initio* Electronic Structure Calculations

By

David Brian Raczkowski

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Physics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

Davis

Approved:

Ching-Yao Fong, Professor of Physics

Barry M. Klein, Professor of Physics

Warren Pickett, Professor of Physics

Committee in Charge

2000

David Brian Raczkowski

September 2000

Physics

An Occupied Subspace Optimization for Linear Scaling in Large Scale *Ab Initio* Electronic Structure Calculations

Abstract

We present an approach to electronic structure calculations that replaces the traditional technique of diagonalizing the Hamiltonian matrix for systems with a gap between the occupied and unoccupied eigenvalues. Instead of eigenfunctions that are generally delocalized, we directly calculate non-orthogonal localized orbitals in order to obtain the band energy, which is the sum of the occupied eigenvalues, and the total charge density of a system. This localization allows for the systematic approximation that the orbitals have a non-zero contribution from only a subset of all of the basis functions. The computational effort is placed where the interactions are the strongest. The result is that the calculation can be more efficient with the dominant parts scaling as O(N).

We chose a gaussian basis as our representation of *primitive* space for this approach due to its locality, moderate number of basis functions, established utility, and accessibility to a mature, working code (Seqquest). The non-orthogonal localized orbitals are obtained by a minimization of a generalized Rayleigh quotient. We discuss many numerical and geometrical considerations that formed our decisions for implementing the minimization procedure. We discuss the effects of localization on the accuracy of total energies, forces, and relaxed geometries in calculations involving silicon carbide and Yttria-stabilized Zirconia. The linear scaling of the certain parts of the algorithm is established up to systems of 1568 atoms.

Approved:

Ching-Yao Fong, Professor of Physics

Acknowledgements

I wish to thank . . .

. . . C. Y. Fong for being a mentor; for showing me physical insight into many mathematical equations in physics; for always being open to discussion; for his stressing and openness of the truth and honesty in physics; and for his help in securing my next position.

. . . Ellen Stechel for being a mentor; for her great advice and suggestions without which this work could not have been accomplished; for help in securing financial support during two summer internships; and for her heartfelt desire to do what is best for science.

. . . Peter Schultz for being a mentor; for making me think hard about my future; for promptly replying with thorough answers via e-mail; for writing understandable code, uncommon in physics and with the sometimes unwieldy Fortran77; and for his help in securing financial support for my last year of work.

. . . Ross Lippert for his mathematical insight into physics problems; for clear and concise explanations that have lead to my better understanding of a geometrical picture of minimization and the eigenvalue problem; and for ridiculing me when I was willing to settle for a less than perfect algorithm.

. . . Chris Wolverton for introducing me to what computer simulation can do in a real-world physics problem; and for helping me respect the intelligent users of code as being anything but technicians.

To my Mom and Dad, and

Mike, Paul, Therese, and John

INTRODUCTION

The impact of theoretical computer simulations is starting to be felt and will only continue to grow. Scientists of all fields are viewing the computer as a necessary resource that will complement experiment and pure theory. The extraordinary acceleration of computer resources has facilitated this endeavor. Computer simulations are now at the point that they can spur improvement in the materials of the same microchip on which the calculations occur. There are simulations for every length and time scale from the microscopic to the macroscopic with certain techniques attempting to combine all into one simulation. There may be a day when all levels of manufacturing, whether of microchips, chemicals, or pharmaceutical drugs, will be simulated on a computer before money and time is spent on the actual fabrication.

Calculations are run at many theoretical levels, each with differing degrees of accuracy and computational effort. Quantum mechanical methods for materials can be very accurate; but they are computationally intensive calculations so that they are presently viable for only relatively small systems (∼100 atoms), especially when including temperature effects. In order to handle systems with more atoms and longer time steps for the evolution of the nuclei and electrons over time (sampling phase space), approximate techniques are implemented. Inter-atomic potentials (∼10,000 atoms) define a force field around the atoms offering a more efficient method for total energy calculations and time evolution of the system[1]. In materials science for treating larger systems (∼100,000 atoms), the atoms are restricted to a lattice. A parameterized Hamiltonian describes the energy with Monte Carlo algorithms to explore configurational entropy[2] and the kinetic Monte Carlo method[3] to explore evolution of the system. For even larger systems, parameterized differential equations such as that for heat flow are solved using continuum mechanics. These macroscopic processes are on the scale of meters.

No matter how fast computers are, scientist will always be pushing the limit of accuracy in theoretical calculations demanding systems with more atoms and longer time steps. Therefore, it is necessary to create the most efficient algorithms and code in order to take full advantage of the resources available. An example of better code is the use of inter-atomic potentials on massively parallel machines in biological molecular dynamics which require long times for a protein to fold or enough phase space to be explored in order to create a statistically valid ensemble. An example for better algorithms is the use of hyper-dynamics[4] in modeling epitaxial growth with inter-atomic potentials. Many algorithms exploit approximations on all levels in order to achieve the desired accuracy with a reasonable computational effort.

One quantum mechanical approach with much applicability is Density Functional Theory (DFT), see Sec. 2.2. This is the level of theory used for all calculations in this paper. DFT gives very accurate ground-state structural properties and in many instances quantitatively accurate energy information. It is relatively expensive computationally compared to a parameterized Hamiltonian or inter-atomic potentials, but less than calculations which take into account electron correlation at a higher level. There has been a push recently for new algorithms to extend DFT and tight-binding models to larger systems. The bottlenecks are the calculation of the matrix elements of the Hamiltonian including the long-range Coulomb interaction, and the diagonalization of this matrix. This thesis addresses the latter. The crux of the thesis is therefore the theory and implementation of a new algorithm to push the applicability of DFT to larger systems. All calculations reported here are done on single processors.

Eigenfunctions: Traditional Solutions of DFT

DFT allows for a tractable approximate solution to a many-body Schroedinger equation. The solution of this equation gives understanding and quantitative knowledge of the energy and positions of the electrons and nuclei of a material. Within the independent electron approximation, the resulting equations are a set of non-linear differential equations. The non-linearity results from the electron-electron interaction.

The first step of solving this as for other differential equations is to choose a suitable functional representation space for the electronic wavefunctions. The wavefunctions will be linear combinations of the basis functions in this function space. Once a suitable *primitive* function space for the electrons has been chosen, the set of non-linear differential equations are solved by fixing the potential due to the electrons which in traditional formulations defines an eigenvalue problem. The wavefunctions in this formulation are eigenfunctions of the DFT Hamiltonian. The solution of an eigenvalue problem scales as $O(N^3)$ for N equal to the number of electrons in a system.

The one-electron eigenfunctions are of great utility but should not be clung to as the only method for describing the electronic states. They are the stationary states of the Hamiltonian and are needed for quantitative studies for the optical spectrum, electron transport by Green function analysis, and provide simple formulas for many-body wavefunctions (Slater determinants). However, generally they are delocalized, ensuring an $O(N^3)$ scaling. This drawback manifests itself in the lack of a local understanding of the electronic environment.

This drawback has spurred much work to find other avenues for describing the electronic ground state. All of the other techniques must then find an alternative to solving the eigenvalue problem or a technique for transforming the eigenfunctions into localized orbitals, e.g. a unitary transformations for Wannier orbitals in a crystal. These techniques amount to finding a more suitable set of coordinates for the *occupied* space, the vector space spanned by the occupied eigenfunctions that defines the electronic ground state. An analogy for a more suitable choice of coordinate representation would be to use spherical over cartesian coordinates for the solution of a particle in a spherical potential. Spherical coordinates are the most suitable choice for this problem. Learning from this analogy, for some physical systems and for some desired results, it is now clear that eigenfunctions might not be the most suitable choice for the solution of the electronic ground state. We now address these localized orbitals and the benefits that they possess.

Localized Orbitals and Linear Scaling

The idea of using solutions other than the eigenfunctions has existed, more predominantly in chemistry, for a long time. Fock[5] pointed out that a unitary transformation of the eigenfunctions gave localized orthogonal orbitals. In the physics community, Wannier[6] also described the transformation of all Bloch orbitals, eigenfunctions of a solid, in studying the electronic excitation levels in insulating crystals.

An early use of localized orbitals was by Coulson[7] in discussing the dipole moment of the C-H bond. The orbitals were localized along the bond to form four equivalent C-H bond orbitals. Lennard-Jones and Pople[8] pointed out that these equivalent orbitals presumably maximize the sum of the orbital self-repulsion (self-interaction) terms in the electronic interaction energy. These orbitals of maximum " localization" would therefore be suitable for methods accounting for electron correlation because the inter-orbital correlation would be at a minimum and the intra-orbital correlation would be the dominant correction.

Recent work has applied these benefits of localized orbitals to materials calculations. Localized orbitals have been used to calculate polarization of materials with and without the presence of an electric field[9−11]. In these calculations, the criterion for localization was that of Marzari and Vanderbilt,[12] which is to minimize the expectation value of (¡r^2¿-¡**r**¿^{2}) for a Wannier orbital. Localized orbitals have also been used for the unoccupied (virtual) orbitals for correlation techniques[13,14].

Lowdin[15] defined mathematical equations to account for the non-orthogonality of the localized orbitals, e.g. in the definition of the single-particle density matrix. Adams[16] presented a formulation for directly finding these orbitals, which was lacking in previous formulations that had to form them from the eigenfunctions. He derived a minimization principle, but for practical reasons chose to solve approximate solutions for each atom[17]. The ideas presented by him and others[18,19] were mostly to justify empirical tight-binding models and to develop fast but approximate methods. Later these same ideas were used to foster the chemical pseudo-potential ideas of Anderson[20,21], which also were empirical tight-binding models. Adams[22] and Gilbet[23] also saw the benefits of these orbitals for use in correlation calculations, specifically multiconfigurational self-consistent-field theory (MCSCF). Adams[24] also presented the idea of molecularly invariant orbitals, which has been continued in the work of Hierse and Stechel[25]. Adams also suggested the use of the localized orbitals as a starting point for a population analysis[26], or for the defining the charge associated with an atom and bond order.[27,28]

In this thesis we utilize a recently realized benefit of localized orbitals, linear scaling[29−32] of computational effort with system size (number of electrons). As the orbitals are localized, the interaction of an electron wavefunction with its environment has a limited spatial extent. This follows from the chemical intuition that for most systems, due to screening of the Coulomb interaction, interactions are not long-ranged. For example, the C-C bond is almost identical in any system where it appears regardless of the far environment. The spatial extent is determined by ignoring interactions below a certain numerical value or distance. Therefore, even as one looks at larger systems, the interactions for a given electron wavefunction stays constant. The result is linear scaling of the computational effort with system size.

In order to achieve linear scaling, one cannot obtain the localized orbitals from the eigenfunctions. One must calculate the orbitals directly, usually by a variational minimization procedure. This is done with several different methods within this orbital formulation and in techniques that bypass the orbitals by a density matrix formulation.[33,34] This thesis presents a new minimization method based on existing formulations[35,36] in order to obtain linear scaling for the computationally dominant parts of the algorithm.

Outline

Chapter 2 details many of the approximations involved in solving for the ground state of a system in DFT. The purpose of this chapter is to review all the approximations involved in real-world calculations. If one carefully uses these approximations, quantitatively accurate results are obtained. Part of this thesis consists of the investigation of how to implement the approximations inherent for a linear scaling algorithm in an effective, accurate, and efficient manner.

Chapter 3 starts by reviewing the general concepts of minimization algorithms. We then give some mathematical background of the role that the (generalized) eigenvalue problem has in electronic structure calculations. The eigenvalue problem is presented as equivalent to a constrained minimization of the Rayleigh quotient. Viable alternatives to this constrained minimization allowing for the relaxing of some constraints are discussed.

The main quantity sought for by the minimization algorithms is the *occupied* space. An introduction to linear scaling algorithms as well as the constraint (Grassmann) manifold inherent in finding the occupied space is presented as a concise summary of recent mathematical work.[35,36] We relate how some linear scaling algorithms handle the constraints. We finally present a Grassmann conjugate algorithm for solving of the electronic ground state. The algorithm accounts for non-orthogonality at every level for the involved vector spaces.

Chapter 4 describes the implementation of the algorithm into the serial gaussian DFT code, SEQQUEST. Section 4.1 gives an overview of the strategy for the sparse storage of matrices. This subsection describes the storage in detail. The purpose of the subsection is to help programmers, who wish to make modifications, to obtain some understanding before jumping directly into the code. Section 4.2 handles modification to the Grassmann conjugate gradient algorithm necessary for sparse calculations. Section 4.3 gives the strategy used for the sparse matrix multiplications and details, again to aid in future modifications. Section 4.4 describes the general method already in place in the code for handling calculations of the self-consistent matrix elements and details the additions made for handling sparse matrices. Section 4.5 gives a brief summary of the modifications needed to allow the existing subroutines, for the calculation of forces, to handle sparse matrices.

Chapter 5 gives the results of calculations using this algorithm. Attention is given to accuracy, convergence, computational effort, and practical use. First, we shall briefly describe our first test system, silicon, with an emphasis on computational effort and ability to handle large systems. For silicon carbide, we stress accuracy and convergence properies. We next turn our attention to a calculation involving forces. Yttria-stabilized zirconia is our test system for studying the effect that the localization of the occupied orbitals has on forces, relaxed geometries, and formation energies. Our last calculations involve the modification of the electronic ground state search to account for deep level defects. A single nitrogen atom replacing a silicon atom in silicon carbide serves this purpose.

1 J. A. McCammon and S. C. Harvey, *Computer Simulations of Liquids* (Cambridge University Press, 1987).

2 P. D. Tepesch, G. D. Garbulsky, and G. Ceder, Phys. Rev. Lett. **74**, 2272 (1995).

3 A. F. Voter, Phys. Rev. B **34**, 6819 (1986).

4 A. F. Voter, Phys. Rev Lett. **78**, 3908 (1997).

5 V. Fock, Z. Physik **61**, 126 (1930).

6 G. H. Wannier, Phys. Rev. **52**, 191 (1937).

7 C. A. Coulson, Trans. Faraday Soc. **38**, 433 (1942).

8 J. E. Lennard-Jones and J. A. Pople, Proc. Roy. Soc. (London) **A202**, 166 (1950).

9 D. Vanderbilt and R. D. King-Smith, Phys. Rev. B **48**, 4442 (1993).

10 F. Bernardini, V. Fiorentini, and D. Vanderbilt, Phys. Rev. Letters **79**, 3958 (1997).

11 P. Fernandez, A. Dal Corso, and A. Baldereschi, Phys. Rev. B **58**, 7480 (1998).

12 N. Marzari and D. Vanderbilt, Phys. Rev. B **56**, 12 847 (1997).

13 A. Shukla, M. Dolg, P. Fulde, et al., Phys. Rev. B **60**, 5211 (1999).

14 R. Knab, W. Forner, J. Cizek, et al., J. of Mol. Struct. **366**, 11 (1996).

15 P.-O. Löwdin, J. Chem. Phys. **18**, 365 (1950).

16 W. H. Adams, J. Chem. Phys. **34**, 89 (1961).

17 W. H. Adams, J. Chem. Phys. **37**, 2009 (1962).

18 C. Edmiston and K. RuedenBerg, Rev. Mod. Phys. **35**, 457 (1963).

19 T. L. Gilbert, in *Molecular orbitals in Chemistry, Physics, and Biology*, edited by P.-O. Löwdin and B. Pullman (Academic, New York, 1964).

20 P. W. Anderson, Phys. Rev. Lett. **21**, 13 (1968).

21 P. W. Anderson, Physics Reports **110**, 311 (1984).

22 W. H. Adams, Phys. Rev. **156**, 109 (1967).

23 T. L. Gilbert, Phys. Rev. A **6**, 580 (1972).

24 W. H. Adams, J. Chem. Physics **42**, 4030 (1965).

25 W. Hierse and E. B. Stechel, Phys. Rev. B **54**, 16 515 (1996).

26 R. S. Mulliken, J. Chem. Phys. **23**, 1833 (1955).

27 C. A. Coulson, Proc. Roy. Soc. (London) **A169**, 419 (1939).

28 B. H. Chirgwin and C. A. Coulson, Proc. Roy. Soc. (London) **A201**, 3428 (1950).

29 G. Galli and M. Parrinello, Phys. Rev. Lett. **69**, 3547 (1992).

30 F. Mauri and G. Galli, Phys. Rev. B **50**, 4316 (1994).

31 P. Ordejn, D. A. Drabold, R. M. Martin, et al., Phys. Rev. B **51**, 1456 (1995).

32 E. B. Stechel, A. R. Williams, and P. J. Feibelman, Phys. Rev. B **49**, 10 008 (1994).

33 X.-P. Li, R. W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10 891 (1993).

34 D. R. Bowler and M. J. Gillan, Comp. Phys. Com. **120**, 95 (1999).

A. Edelman, T. A. Arias, and S. T. Smith, SIAM J. on Matrix Anal. Appl. **20**, 303 (1998).

R. A. Lippert and M. P. Sears, Report SAND99-2986, Sandia National Laboratories, Albuquerque, NM USA, (1999).

APPROXIMATIONS

Approximations are crucial in the theoretical calculation of real-world physics problems. Condensed matter physics is no exception. One could never hope to exactly calculate the interactions in the complex macroscopic materials of the modern world. One needs to concentrate on the interactions leading to the properties that are most important for the problem at hand. It takes a very skilled, wise, and experienced physicist to implement, in an accurate and descriptive manner, the myriad of techniques that are used today. Each approximation has its own place. The most important thing is to know in a calculation where the approximations are, where they come from, and what results and for what systems will the approximations have an adverse impact.

Many Body Hamiltonian

The solution of the many-body Schrodinger equation is the key problem in condensed matter physics and has spawned many approximations. The equation for N electrons and K nuclei, respectively defined with the coordinates **r** and **R** implicitly including spin, is

$\mathrm{H}(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N};\ \mathbf{R}_1,\ldots,\mathbf{R}_\mathrm{K})\ \Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N};\ \mathbf{R}_1,\ldots,\mathbf{R}_\mathrm{K}) = \mathrm{E}\ \ \Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N};\mathbf{R}_1,\ldots,\mathbf{R}_\mathrm{K})$ (2.1)

In a condensed matter context, the Hamiltonian is

$\sum_{\mathrm{i}=1}^{\mathrm{N}}(-\frac{\hbar^2}{2m}\nabla_i^2 \quad - \quad \sum_{\mathrm{j}=1}^{\mathrm{K}}\frac{Z_j e^2}{|\mathbf{r}_i-\mathbf{R}_j|} \quad + \quad \frac{1}{2}\sum_{\mathrm{j}\neq\mathrm{i}}^{\mathrm{N}}\frac{e^2}{|\mathbf{r}_i-\mathbf{r}_j|}) \quad + \frac{1}{2}\sum_{\mathrm{i}=1}^{\mathrm{K}}\sum_{\mathrm{j}\neq\mathrm{i}}^{\mathrm{K}}\frac{Z_i Z_j e^2}{|\mathbf{R}_i-\mathbf{R}_j|}$ (2.2)

The first approximation usually made is the Born-Oppenheimer (adiabatic) approximation, which decouples the movement of the nuclei and the electrons. This is the first example of taking advantage of different length and/or time scales. As the significant electronic velocity is $\mathrm{v_f} \approx 10^8$ cm/sec, the electrons in most cases move much faster than the nuclei, typically 10^5 cm/sec.

Since the electrons move so fast and therefore quickly settle in their ground-state configuration, the electrons can be solved for separately. So we have an electronic, $\mathrm{H_E}$, and a nuclear, $\mathrm{H_N}$, Hamiltonian. Generally, the electronic states are solved quantum mechanically in the field of a static nuclear configuration with the electronic many-body Schroedinger's equation.

$\sum_{\mathrm{i}=1}^{\mathrm{N}}(-\frac{\hbar^2}{2m}\nabla_i^2 \quad - \quad \sum_{\mathrm{j}=1}^{\mathrm{K}}\frac{Z_j e^2}{|\mathbf{r}_i-\mathbf{R}_j|} \quad + \quad \frac{1}{2}\sum_{\mathrm{j}\neq\mathrm{i}}^{\mathrm{N}}\frac{e^2}{|\mathbf{r}_i-\mathbf{r}_j|})\Psi$ $(\mathrm{r}_1,\ldots,\mathrm{r}_\mathrm{N})=\mathrm{E}\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N})$ (2.3)

The solution for the nuclear Hamiltonian is to incorporate the forces from the electrons and the nuclei into molecular dynamics or to relax the atomic positions to find the geometry with the lowest energy. Molecular dynamics samples phase space to get a statistically accurate ensemble at finite temperature or models growth and diffusion of atoms. For geometry relaxation, the forces are fed into a minimization algorithm to find the positions, usually just a local minimum, with the lowest energy developed from a starting configuration.

One cannot solve Eq (2.3) exactly so one generally uses an independent electron picture. Although recent advances in Monte Carlo techniques[1−3] have made the calculation of the many-body wavefunction accessible for electronic structure calculations, the computational effort is very high and the codes are not readily available in a convenient user-friendly package. Even if Monte Carlo techniques are used, they need the help of the results of an independent electron calculation for an initial guess at the functional form for $\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N})$. Monte Carlo calculations have made a big impact by the providing the exchange-correlation energy density, see Sec. 1.2, for density functional theory.

We now consider the many-body wavefunction to be comprised of N single-particle wavefunctions. The many-body wavefunction $\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N})$ is now approximated by

$\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N}) = \Psi_1(\mathbf{r}_1)\Psi_2(\mathbf{r}_2)\ldots\ \Psi_\mathrm{N}(\mathbf{r}_\mathrm{N})$. (2.4)

The final wavefunction is a simple combination of vectors of length M, the number of basis functions in the *primitive* space. This combination manifests itself as a matrix of M x N defining the *occupied* space as made up of the lowest N eigenfunctions of a single-particle Hamiltonian.

The Hartree approximation, which results in a self-consistent field (SCF) equation traditionally involving an eigenvalue problem, was the first to give a prescription for solving the many-body equation in terms of single particle states. The Coulomb term becomes an effective mean field that acts on every electron with the same strength. This average potential $\mathrm{v}(\mathbf{r})$, due to the charge density

$\mathrm{n}(\mathbf{r}) = \sum_{i=1}^{N}\Psi_\mathrm{i}{}^*(\mathbf{r})\Psi_\mathrm{i}(\mathbf{r})$ (2.5)

of the N electrons, is

$$\mathrm{v}(\mathbf{r}) = e^2\int\frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}'. \qquad (1)$$

The resulting eigenvalue equation is

$$(-\frac{\hbar^2}{2m}\nabla_i^2 - \sum_{\mathrm{j}=1}^{\mathrm{K}}\frac{Z_j e^2}{|\mathbf{r}_i-\mathbf{R}_j|} + \mathrm{v}(\mathbf{r}))\Psi_i = E_i\Psi_i \qquad (2)$$

For every eigenvalue solution the wavefunctions form the new potential and hence the Hamiltonian for a new eigenvalue problem. This is continued until the input and output charge density or potential doesn't change within some prescribed accuracy. Typically the new potential is chosen as some linear combination of the previous potentials.

This approximation neglects the Pauli-exclusion principle, which was remedied by the following prescription of Fock. The many-body wavefunction is written as a Slater determinant

$$\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N}) = \begin{vmatrix} \Psi 1(\mathbf{r}1) & \Psi 2(\mathbf{r}1) & \ldots & \Psi N(\mathbf{r}1) \\ \Psi 1(\mathbf{r}2) & \Psi 2(\mathbf{r}2) & \ldots & \Psi N(\mathbf{r}2) \\ . & . & . & . \\ \Psi 1(\mathbf{r}N) & \Psi 2(\mathbf{r}N) & \ldots & \Psi N(\mathbf{r}N) \end{vmatrix} .$$

(2.8)

The wavefunction now satisfies the physical condition that the electrons are indistinguishable and are fermions, thus $\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N})$ is anti-symmetric and should have a sign change upon interchanging two electrons. If this new form of $\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N})$ is now used to calculate the charge density and the resulting potential, the result is an additional term, the exchange term

$$-\sum_{i=1}^{N} \int \frac{\Psi j(\mathbf{r}')\Psi^* i(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \delta_{s_i s_j} d\mathbf{r}' , \qquad (3)$$

in Eq. (2) for the Hamiltonian involving electrons of the same spin.

Slater[4] suggested a simplification for non-uniform systems, replacing the exchange term by a local energy given by the exchange term of a uniform gas at the given charge density. Both of these methods neglect correlation. Electron correlation is the error resulting from the mean field Coulomb interaction and the independent electron approximation for the electrons compared to the exact solution. It arises from the fact that an electron's movements are correlated with the electrons around it. This interaction has a varying degree of importance for different systems. Finally, Kohn, with the help of Hohenberg and Sham, developed a technique to add in correlation effects.

Density Functional Theory

In 1964, Hohenberg and Kohn[5] (HK) developed an exact formal variational principle for the ground-state energy, in which the density n($\mathbf{r}$) is the variable function. Since $\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N})$ is a functional of n($\mathbf{r}$), so is evidently the kinetic, $\mathbf{T}$, and interaction, $\mathbf{U}$, energy. They define a universal functional

F[n($\mathbf{r}$)] $\equiv$ ($\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N})$ **, T+U** $\Psi(\mathbf{r}_1,\ldots,\mathbf{r}_\mathrm{N})$), (2.10) which applies to all electronic systems in their ground state in any external potential. F[n($\mathbf{r}$)] is minimized with respect to n($\mathbf{r}$) as a more tractable alternative than solving the many-body electronic Schrodinger equation. They derived an expression for F[n($\mathbf{r}$)] in the cases: (1) n($\mathbf{r}$) deviates only slightly from uniformity, i.e. n($\mathbf{r}$)=n_0 + ñ($\mathbf{r}$), with ñ/n_0 $\rightarrow$0, and (2) n($\mathbf{r}$) is slowly varying but not necessarily almost constant, i.e. n($\mathbf{r}$)=$\phi(\mathbf{r}/r_0)$, $r_0 \rightarrow \infty$. For the second case, they derive an expansion of F[n] in successive orders of r_0^{-1} or, equivalently of the gradient operator ∇ acting on n($\mathbf{r}$).

They start by showing that a unique ground-state charge density can be found for any external potential, v($\mathbf{r}$). Conversely they also show that v($\mathbf{r}$) is a unique functional of n($\mathbf{r}$), apart from a trivial additive constant. Since the charge density can be created from the single particle states, an independent electron equation should lead to the exact solution. There is a problem in knowing the functional of the charge density and the single particle equation though.

For an external potential v($\mathbf{r}$), F[n(r)] is defined by

F[n($\mathbf{r}$)] $= \int v(\mathbf{r})n(\mathbf{r})d\mathbf{r} + \frac{1}{2}\int\int \frac{n(\mathbf{r})n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}' + G[n(\mathbf{r})]$
(2.11)

G[n($\mathbf{r}$)] is a universal function of the density representing the kinetic, exchange, and correlation energy of a system. From Eq. (2.10) and Eq. (2.11), G[n($\mathbf{r}$)] is defined by

G[n($\mathbf{r}$)] $= \frac{1}{2}\int \nabla_\mathbf{r}\nabla_{\mathbf{r}'} n_1(\mathbf{r},\mathbf{r}')|_{r=r'} d\mathbf{r} + \frac{1}{2}\int\int \frac{C_2(\mathbf{r},\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}'$.
(2.12)

Here n_1($\mathbf{r}$,$\mathbf{r}$') is the one partcle density matrix: and C_2($\mathbf{r}$,$\mathbf{r}$') is the two-particle correlation function defined in terms of the one- and two-particle densities as

C_2($\mathbf{r}$,$\mathbf{r}$') =n_2($\mathbf{r}$,$\mathbf{r}$') - n($\mathbf{r}$) n($\mathbf{r}$') (2.13)

In 1965, Kohn and Sham[6] developed an implementation of this theorem. The equations are for single particle states much in the same fashion as the Hartree-Fock equations are. The practical difference is that these equations have a compensation for electron correlation. G[n($\mathbf{r}$)] is separated into the kinetic and exchange-correlation parts

G[n($\mathbf{r}$)] = T[n($\mathbf{r}$)] + E_xc[n($\mathbf{r}$)], (2.14)
and they stated that HK showed that if n($\mathbf{r}$) is sufficiently slowly varying

E_xc[n($\mathbf{r}$)] $= \int n(\mathbf{r})\varepsilon_{xc}(n(\mathbf{r}))d\mathbf{r}$ (2.15)

This is known as the local density approximation (LDA). In the slowly varying limit ε_xc, the exchange-correlation energy density, is assumed known. From the stationary property of Eq (2.11), we get with $\upsilon_{xc}(n) = d(n\varepsilon_{xc}(n))/dn$ (2.16)
the one-particle Schroedinger equation

$$(-\frac{\hbar^2}{2m}\nabla_i^2 - \sum_{\mathrm{j}=1}^{\mathrm{K}} \frac{Z_j e^2}{|\mathbf{r}_i - \mathbf{R}_j|} + v(\mathbf{r}) + \upsilon_{xc}[n(\mathbf{r})])\Psi_i = e_i\Psi_i, \qquad (4)$$

Eq. (4) is solved in a self-consistent manner using Eqs. (2.5) and (1) as is done for the Hartree-Fock equations. The final electronic energy is given by

E= $\sum_{i=1}^{N} e_i - \frac{1}{2}\int\int \frac{n(\mathbf{r})n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}' + \int n(r)[\varepsilon_{xc}(n(r)) - \upsilon_{xc}(n(r))]dr$. (2.18)

There have been many subsequent attempts to improve on LDA or the local spin density approximation (LSD), where the potential to spin up and down electrons are calculated individually. The first attempt[7], the gradient-expansion method, was to use some additional terms in the gradient expansion of $E_{xc}[n(\mathbf{r})]$ in HK. This work gave correlation energies of the wrong sign. Self-interaction corrections[8,9] to the LSD approximation have been useful, but not entirely satisfactory.

Instead of additional terms involving gradients in $E_{xc}[n(\mathbf{r})]$, ε_{xc} can be made to have functional dependence upon gradients. The resulting equation is

$E_{xc}[n] = \int n(\mathbf{r})\varepsilon_{xc}(n(\mathbf{r}), \nabla n(\mathbf{r}))d\mathbf{r}$ (2.19)

Langreth and Mehl and co-workers[10−13] developed a generalized gradient approximation (GGA) that predicts rather accurate correlation energies for atoms[11,12], molecules, and solids.[14] This was subsequently been modified[15,16] while still keeping within only a gradient correction. There are also meta-generalized gradient approximations (MGGA)[17−19]. These formulations include the Laplacian

$$\nabla^2 n(\mathbf{r}) \tag{5}$$

or the kinetic energy density of the Kohn-Sham orbitals,

$\tau\ (\mathrm{r}) = \sum_{i=1}^{N} |\nabla\Psi i(\mathbf{r})|^2$ (2.21)

Ref. [16], using the latter of these two forms, comments that the improvements of the MGGA seemed unreachable with the GGA form. GGA or MGGA are still a local functional for numerical purposes. They are semilocal in the sense that gradient information is included, but they should not be described as non-local as cited in some literature. These are numerically efficient as they keep the potential local, only information about $n(\mathbf{r})$ at a single point determines the potential at that point.

A non-local functional[20], which the exact exchange-correlation functional most certainly is, has recently appeared in the literature. The exchange-correlation functional is based on a short-range approximation for the Coulomb hole function. The Coulomb hole function, which includes the electron pair correlation function, is integrated over real space to give a value for ε_{xc}. The kinetic energy contribution to $E_{xc}[n]$ is calculated by an inter-electronic coupling parameter λ and integrating between the limits λ=0 and λ=1 for the λ-dependent Coulomb hole function.

Mermin[21] formulated a finite temperature generalization of Eq. (2.11). He showed that the grand canonical potential could be written in the form

$$\Omega = \int v(\mathbf{r})n(\mathbf{r})d\mathbf{r} + \frac{1}{2}\int\int \frac{n(\mathbf{r})n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}' + G[n(\mathbf{r})] - \mu\int n(\mathbf{r})d\mathbf{r} \tag{6}$$

where $G[n]$ is a unique functional of the density and μ is the chemical potential, both at a given temperature T.

Basis Set Approximations

In order to solve the DFT or Hartree-Fock equations, one must first choose a *primitive* space of basis functions. Ideally one would use a complete Hilbert space, but in practice one must discretize the basis with a finite number of elements in order to obtain tractable matrix representations of the operators and the electronic wavefunctions. Complete bases, such as plane waves, are truncated in a controlled manner by increasing the size of the basis until the total energy is converged, termed arbitrarily complete.

The basis set of plane waves can be quite large, and even though it is conceptually suited for free-electron metals they often do not work well for other realistic systems. Incomplete and specifically non-orthogonal bases of atomic orbitals, such as gaussians, are used for efficiency. These linear combination of atomic orbitals (LCAO) bases are almost always the choice of quantum chemists as they physically resemble the exponential decay of atomic and molecular wavefunctions. Since the basis functions are non-orthogonal there is not a simple prescription for adding more basis functions until the energy is reasonably converged. The art of making accurate basis functions has been the subject of many publications.[22−25] If one is interested in very accurate results, such as calculations for capturing most of the correlation energy, then one typically uses large basis sets. Basis sets by Woon and Dunning[26] are such basis sets typically used for configuration interaction calculations.

Even though the basis sets used may be far from complete, they can still form an accurate description of the Hilbert space. Since only relative energies are important, the interactions only need to be determined to the same systematic accuracy in all of the different calculations. This is simply done if the basis set is converged. Another approximation for LCAO bases is that interactions in a periodic solid are neglected at certain distances. In addition, some LCAO bases do not strictly go to zero with distance so the integral of two basis functions are also chosen to be neglected at certain distances. This prescription allows for the linear scaling of the calculation of the matrix elements.

This thesis elaborates on another approximation to a different Hilbert space, the vector space of the occupied wavefunctions. The localization of the occupied orbitals, by a "localization region", causes an approximation of the space spanned by the lowest eigenvectors of a generalized eigenvalue problem. As with the convergence of the underlying basis sets, the total energy does not have to be converged with respect to the size of the localization region, but rather the relative energy is obtained accurately. As with the creation of LCAO basis, such as gaussians, this takes art and experience in testing on different systems to develop this approach.

Pseudopotentials

Within density functional theory, there are many approximations due to the necessity of efficient implementation. In the last section, we discussed incomplete or truncated basis sets. For plane waves and to a lesser extent LCAO basis sets, the basis sets were still too large for implementation of the desired accuracy, and the convergence of the total energy with an increasing number of plane waves was still too slow. The solution was to reformulate the Hamiltonian with the approximation of pseudopotentials for the interaction of valence electrons with the nuclei and the core electrons. The resulting smoother, more free electron-like, pseudo-wavefunctions for the valence electronic states have the same energetics and the same scattering properties outside the core region. There is a motivating physical reasoning as to why the electrons could be described by such free electron-like states. There is an almost complete cancellation between the large negative potential energy of the electron near an atomic nucleus, and the positive kinetic energy associated with the rapid oscillations of the wavefunction near the core[27].

The pseudopotential method has its roots in the Orthogonalized Plane Wave (OPW)[28] method, which minimizes the necessary number of plane waves by a basis set transformation. The idea is that the wavefunction of a valence electron is nearly a plane wave, or a linear combination of plane waves, in the region between the ion cores, but near the ion core there are many oscillations requiring many plane waves. These oscillations can be removed by subtracting off atomic core states. So for ξ being the OPW basis and

$$\Psi = \sum C_n \xi_n, \tag{7}$$

the wavefunction can be described with fewer planes waves by including the atomic core orbitals, η_t, in the definition of ξ,

$$\xi_k = e^{i\mathbf{k}\cdot\mathbf{r}} - \sum_t |\eta_t\rangle\langle\eta_t|e^{i\mathbf{k}\cdot\mathbf{r}}\rangle \quad . \tag{8}$$

The coefficients of the atomic orbitals are chosen[29] to make the resulting valence electron orthogonal to the atomic core states. The resulting equation is

$$\sum_m [\langle n|H|m\rangle - E\langle n|m\rangle]C_m = 0 \tag{9}$$

Phillips and Kleinman[30,31] demonstrated the effect of orthogonalization as a pseudopotential. Subsequent work[32] pointed out that the original ideas, which were mostly for a qualitative or formal justification of the nearly free electron approximation and similar crude models, could be used outright for a method to carry out quantitative calculations. One can now rearrange the previous discussion into a methodology for transforming the DFT equation

$H\Psi_v=(T+V)\Psi_v= E_v\Psi_v$ (2.26)

Into an equation for the pseudo-orbitals Θ

$(H+V_R)$ $\Theta v=(T+V+V_R)\Theta_v= E_v\Theta_v$

when V_R is a nonlocal repulsive potential which cancels off most of V, leaving a weak net potential $(V+V_R)$, which is the pseudopotential.

A giant step forward in creating pseudopotentials was taken by Hamann, Schluter, and Chiang[33] who inverted the Schrodinger equation to obtain a model pseudopotential. The need to calculate the true wavefunction was minimized as the pseudo-wavefunction now had more of the same properties. These were termed norm-conserving pseudopotentials. After a few years of experience, the pseudopotential method developed to the maturity of being transferable for elements in differing environments.[34] Different forms for efficiency have also been introduced such as the separable nonlocal Kleinman-Bylander (KB)[35] form, which decreases the effort in creating the matrix elements of the pseudopotential operator.

A norm-conserving pseudopotential must obey the following conditions. (i) The valence pseudo-wavefunction must be nodeless because of or desire for smooth pseudo-wavefunctions. (ii) The normalized atomic radial pseudo-wavefunction with angular momentum l has to be equal to the normalized radial all-electron wavefunction beyond the chosen cutoff radius r_c(normalization condition, which is important to obtain the correct valence charge density). (iii) The charges enclosed within r_c for both wavefunctions have to be equal. (iv) The valence all-electron and pseudopotential eigenvalues for an atom must agree.

Pseudopotentials are now a staple in many electronic structure calculations in physics and chemistry. Recent improvements have included greater transferability by creating the pseudopotential from all-electron atom potentials at arbitrary energies.[36] This work acknowledged the importance of and difficulty in implementing the KB form in this new technique. The KB form got renewed interest for its ease in calculating the non-local part of the pseudopotential matrix in real space. This real space projection is necessary for the computationally more efficient multiplication of the potential energy matrix on wave functions in real space[37], as opposed to Fourier space. Extending Hamann's work with respect to the KB form, Blochl[38] presented a formulation for directly creating the transferable KB pseudopotentials.

Following from both of these works, Vanderbilt[39] presented ultra-soft pseudopotentials (USPP) in response to the problem that the number of plane waves was still prohibitively large for accurate calculations for transition metals and second row elements. If one dropped the norm-conserving constraint (possible with the newly created greater transferability), one could use larger r_c and create smoother potentials resulting in smoother pseudo wavefunctions and fewer plane waves needed. While keeping within a norm-conserving approach, Troullier and Martins[40] developed softer pseudopotentials although not as soft as a USPP.

Miscellaneous

There are also other seemingly small approximations usually made for practical reasons that often get overlooked. One example occurs in charge state calculations. Papers[41] often refer to the usual method of a calculating the energy of system when a non-zero net charge occurs within the unit cell. The method is to use a uniform background charge[42] to compensate for the net charge. If the unit cell has a net charge, then the solution of Poisson's equation would give an infinite value for the Coulomb potential and ultimately the energy of the material. In practice this amounts to the neglect of the constant term in a Fourier expansion of the Coulomb potential. The exact approximation, the testing of the accuracy, and reliability for the current approach seem to be almost always overlooked. In defense of the research, there is no easy means of testing the accuracy, and part of the testing would need to include calculations of systems normally too large for the computational resources of most physicists.

P. A. Schultz has recently proposed a new algorithm, local moment countercharge (LMCC)[43,44] to have a more physical compensation of charge than a uniform background. He has proposed the use of gaussian charge distributions located at a charge site. This is done similarly for dipole corrections as well. Even for this more physically motivated method, the three-dimensional unit cells necessary, estimated to be around ∼1000 atoms, for accurate results are still too large for most computer resources. This is one good area that linear scaling algorithms could make an impact.

[1] S. Fahy, X. W. Wang, and S. G. Louie, Phys. Rev. B **42**, 2503 (1990)

[2] L. Mitas, E. L. Shirley, and D. M. Ceperley, J. Chem. Phys. **95**, 3467 (1991)

[3] A. J. Williamson, R. Q. Hood, R. J. Needs, and G. Rajagopal, Phys. Rev. B**12**, 12 140 (1998)

[4] J. C. Slater, Phys. Rev. B **81**, 385 (1951).

[5] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).

[6] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).

[7] S.-K. Ma and K. A. Bruechner, Phys. Rev. **165**, 18 (1968).

[8] J. P. Perdew and A. Zunger, Phys. Rev B **23**, 5048 (1981).

[9] S. H. Vosko and L. Wilk, J. Phys. B **16**, 3687 (1983).

[10] D. C. Langreth and J. P. Perdew, Phys. Rev. B **21**, 5469 (1980).

[11] D. C. langreth and M. J. Mehl, Phys. Rev. B **28**, 1809 (1983).

[12] C. D. Hu and D. C. Langreth, Phys. Scr. **32**, 391 (1985).

[13] C. D. Hu and D. C. Langreth, Phys. Rev. B **33**, 943 (1986).

[14] U. von Barth and A. C. Perdoza, Phys. Scr. **32**, 353 (1985).

[15] J. P. Perdew, K.Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[16] A. D. Becke, Phys. Rev. A **38**, 3098 (1988).

[17] J. P. Perdew, Phys. Rev. B **55**, 1665 (1985).

[18] S. K. Ghosh and R. G. Parr, Phys. Rev. A **34**, 785 (1986).

[19] J. P. Perdew, S. Kurth, A. Zupan, et al., Phys. Rev. Lett. **82**, 2544 (1999).

[20] M. Filatov and W. Thiel, Int. J. Quant. Chem. **62**, 603 (1997).

[21] N. D. Mermin, Phys. Rev. **137**, A1441 (1965).

[22] S. Huzinaga, J. Chem. Phys. **42**, 1293 (1965).

[23] W. J. Hehre, R. F. Stewart, and J. A. Pople, J. Chem. Phys. **53**, 932 (1970).

[24] N. Godbout, D. R. Salahub, J. Andzelm, et al., Can. J. Chem. **70**, 560 (1992).

[25] J. Andzelm, E. Radzio, and D. R. Salahub, J. Comp. Chem. **6**, 520 (1985).

[26] D. E. Woon and J. Thom H. Dunning, J. Chem. Phys. **98**, 1358 (1992).

[27] M. H. Cohen and V. Heine, Phys. Rev. **122**, 1821 (1961).

[28] C. Herring, Phys. Rev **57**, 1169 (1940).

[29] V. Heine, Proc. Roy. Soc. (ondon) **A240**, 354 (1957).

[30] J. C. Phillips and L. Kleinman, Phys. Rev. **116**, 287 (1959).

[31] L. Kleinman and J. C. Phillips, Phys. Rev. **118**, 1153 (1960).

[32] B. J. Austin, V. Heine, and L. J. Sham, Phys. Rev. **127**, 276 (1962).

[33] D. R. Hamann, M. Schluter, and C. Chiang, Phys. Rev. Lett. **43**, 1494 (1979).

[34] G. B. Bachelet, D. R. Hamann, and M. Schluter, Phys. Rev. B **26**, 4199 (1982).

[35] L. Kleinman and D. M. Bylander, Phys. Rev. Lett. **48**, 1425 (1982).

[36] D. R. Hamann, Phys. Rev. B **40**, 2980 (1989).

[37] R. Car and M. Parrinello, Phys. Rev. Lett. **40**, 2980 (1985).

[38] P. E. Blochl, Phys. Rev. B **41**, 1990 (1990).

[39] D. Vanderbilt, Phys. Rev. B **41**, 7892 (1990).

[40] N. Troullier and J. L. Martins, Phys. Rev. B **43**, 1993 (1991).

[41] G. Stapper, M. Bernasconi, N. Nicoloso, et al., Phys. Rev. B **59**, 797 (1999).

[42] Y. Bar-Yam and J. D. Joannopolous, Phys. Rev. B **30**, 1844 (1984).

[43] P. A. Schultz, Phys. Rev. B **60**, 1551 (1999)

[44] P. A. Schultz, Phys. Rev. Lett. **84**, 1942 (2000)

SOLVING FOR THE ELECTRONIC GROUND-STATE

In the past 10 years, numerous minimization techniques have appeared in condensed matter literature for solving the electronic ground state[1−19], within tight-binding models (TB), DFT, and Hartree-Fock theory. They have proven to be especially useful in algorithms that calculate the total energy and charge density with an effort linearly proportional to the number of atoms for systems with an energy gap, i.e. insulators and semiconductors. Most linear scaling applications have used TB, which have short-range interactions, due to the fact the matrices are better conditioned and sparser in these models, making the calculations more tractable. The longer-range interactions of DFT and Hartree-Fock are more demanding, thus preventing a quantitative application of a linear scaling algorithm.

In order to achieve the full accuracy of DFT and Hartree-Fock, the Hamiltonian either needs to be of larger size and/or less sparse than in TB. In addition, the localization region (see Sec. 3.1.2 and 4.1.2), which impacts the accuracy of the total energy, needs to be larger to account for the longer-range interactions. A very important consideration for the use of linear scaling algorithms is the point that they become more efficient than diagonalization. Since this crossover varies with localization, which should be chosen based on the desired accuracy, the benefits of linear scaling algorithms are naturally diminished. These demands have limited the application for quantitative results of a linear scaling algorithm within DFT and Hartree-Fock.

In order to meet these demands with a more efficient algorithm that preserves the accuracy of DFT, we address the natural length scales in which the physical system localizes. We have chosen to use nonorthogonal orbitals to span the occupied space. We utilize recent work[20,21]that details the underlying mathematics of the vector spaces, without localization. We investigate numerous technical issues related to the practical implementation of such an algorithm with localization.

Sec. 3.1 gives some mathematical background for the reader not familiar with the minimization algorithms. Sec. 3.2 relates how localization of the physical system at different levels translates into linear scaling of the minimization algorithms. Sec. 3.3 introduces the common geometric framework, the Grassmann manifold[20,21], of the minimization techniques. The aim of this section is to present this recent mathematical insight in a concise, accessible manner and to add new perspectives that pertain to linear scaling. Sec 3.4 presents our minimization algorithm, which has application to any problem that requires the sum of any number of the lowest eigenvalues of a standard or generalized eigenvalue problem and the sub-space spanned by the corresponding eigenvectors. Effort is especially made to ease conceptual understanding while maintaining mathematical rigor. The description of vector spaces is general and deals with possible nonorthogonality at every level.

Mathematical Background

Minimization Algorithms: Basics

The problem of finding the extrema of functions appears in many technical fields. Minimization algorithms have been used for sufficiently feeding soldiers as economically as possible (hence the term "cost function"), in integrated circuit design, for transportation scheduling, and, of course, for finding the ground-state of a system of atoms. I will only describe the basic ideas of finding a minimum, which could be local. There are algorithms that make use of these local minimization schemes to find a global minimum, such as simulated annealing.[22] I also only discuss algorithms for which the function is at least quadratic in the variables, i.e. no linear programming.[23] I emphasize application for electronic structure calculations. In electronic structure, there are basically three minimization techniques used, all of which use the gradient of the function: (1) steepest descent, (2) conjugate gradient and (3) quasi-Newton.

All minimization algorithms have the similarity of requiring a search direction. A movement in this direction is made to minimize the function. The difference of the algorithms comes about in how the search direction is created and how and to what accuracy the proper step size is calculated. It is beneficial for the reader to visualize oneself as moving among hills and valleys in a search for the minimum.

In steepest descent, the search direction is always in the direction opposite of the gradient. As this is the direction of greatest decrease, it seems to be a reasonably suitable choice. A local minimum will eventually be found once the gradient equals zero. With the search direction chosen, we need to know how far to move. One could move a fixed distance every time, but this would generally be a poor course of action. If the current point were far from the one-dimensional local minimum along the current gradient, then one would want a large step size. If one were very close to the minimum in the search direction, then one would want to move only a little bit. So, ideally, we would like to have a step size that varies with the local environment. At least, one needs refinement of the step size if an increase in the function is found.

There are several ways to incorporate a variable step in the search direction and they amount to a one-dimensional line minimization in the current direction. One first must have an estimate of the step length. One way to do this is to make a quadratic approximation about the point $\mathbf{x}_0$ in the direction $\mathbf{x}$, for Λ a scalar giving the distance along $\mathbf{x}$.

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) - \mathbf{b} \cdot \Lambda\mathbf{x} + \frac{1}{2}\Lambda\mathbf{x} \cdot \mathbf{A} \cdot \Lambda\mathbf{x} \tag{10}$$

Here $\mathbf{b}$ is the gradient and $\mathbf{A}$ is the Hessian, the second derivative operator. So if the function was quadratic in the search direction, the step length to get a zero derivative w.r.t. λ is

$$\Lambda = \frac{\mathbf{b} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x}} \tag{11}$$

This should be a good initial guess, but may not be sufficient, as a quadratic approximation may not be good. The test of sufficiency is usually by some criteria of a decrease of the function and of the gradient, commonly called Wolfe conditions. In some cases one may need to do further refinement. The general scheme is to bracket the current guesses for λ until the Wolfe conditions are met. This is typically done by interpolating the function, possibly utilizing gradient information, as a polynomial in λ and using the position of the minimum of the polynomial as your new value for λ. Brent's algorithm[24] was historically one of the first such algorithms without gradient information. We use a similar variant with gradient information by More and Thuente.[25]

Steepest descent turns out to not be a very good algorithm though. The method will perform many small steps in going down a long, narrow valley, even if the valley is perfectly quadratic. The concept of "non-interfering" directions, more conventionally called *conjugate directions*, is useful. In the approximation of Eq. (10), the gradient is

$$\nabla f = \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \tag{12}$$

Now as we move along some direction, the change in the gradient is

$$\delta(\nabla f) = \mathbf{A} \cdot (\delta\mathbf{x}) \tag{13}$$

Suppose we have moved along some direction $\mathbf{u}$ to a minimum and now propose to move along some new direction $\mathbf{v}$. The condition that motion along $\mathbf{v}$ not *spoil* our minimization along $\mathbf{u}$ is just that the gradient, at the point after movement along $\mathbf{v}$, stay perpendicular to $\mathbf{u}$, i.e. that the change in the gradient be perpendicular to $\mathbf{u}$, is given by

$$0 = \mathbf{u} \cdot \delta(\nabla f) = \mathbf{u} \cdot \mathbf{A} \cdot \mathbf{v}. \tag{14}$$

If you do successive line minimization of a function along a conjugate set of directions, then you don't need to redo any of those directions. For a quadratic function of L variables the algorithm is guaranteed to find the minimum in L steps. This idea is used extensively with gradient information as the *conjugate gradient* method. The Hessian information is built up after successive line minimization and is carried from iteration to iteration even though it never explicitly calculated. These methods have been used extensively for minimizing problems involving many variables, in condensed matter specifically, for calculating the occupied subspace of electrons.

Quasi-Newton methods, also called variable metric, build up Hessian information explicitly. Specifically an approximate inverse Hessian $\mathbf{A}^{-1}$ is constructed. In Newton's method, we use Eq.(10) and Eq.(12) and set $\nabla f(\mathbf{x}) = 0$ to determine the next iteration point:

$$-\mathbf{A}^{-1} \cdot \nabla f(\mathbf{x}_i) = (\mathbf{x} - \mathbf{x}_i) \tag{15}$$

This gives the exact step for a quadratic function if the entire Hessian is calculated. If we have a non-quadratic function, then the Newton step can cause an increase in f($\mathbf{x}$) if the current point has a Hessian with a negative eigenvalue, not positive definite. Physically one is near the top of a hill in one direction. The algorithm only wants to make the gradient equal to zero so any critical point will do. The quasi-Newton algorithm solves this problem by starting with a positive definite, symmetric approximation to $\mathbf{A}^{-1}$ and maintains the positive definiteness. Far from the minimum this guarantees that one moves downhill, and close to the minimum enjoys the quadratic convergence of Newton's method. An added benefit is that one does not need to calculate the Hessian, which can be quite time consuming.

The quasi-Newton algorithm builds up the inverse Hessian by requiring that the Hessian correctly predict the change in gradient values from $\mathbf{x}_{\mathrm{i}}$ to point $\mathbf{x}_{\mathrm{i+1}}$ given by

$$\mathbf{A}^{-1} \cdot (\nabla f_{i+1} - \nabla f_i) = (\mathbf{x}_{i+1} - \mathbf{x}_i). \tag{16}$$

This can lead to many different update formulas. Broyden[26] originally imposed the condition that the inverse Hessian changes as little as possible. The *Broyden-Fletcher-Goldfarb-Shanno* (BFGS)[23,27,28] form is usually considered the best form. It is typically the choice for geometry relaxation in electronic structure calculations.

An interesting variant of this is due to Pulay[29], termed "direct inversion in the iterative subspace", and D. D. Johnson,[30] which is predominantly used for the SCF field mixing. For a highly non-quadratic function the update formula for the Hessian will cause information for the past search directions to become incorrect. Eq. (16) cannot be satisfied for all of the past search directions. So in these cases, the most recent search direction is too highly relied on updating the Hessian. The solution is to satisfy Eq. (16) in a least square sense. An appendix of Vanderbilt and Louie[31] gives a nice explanation of the motivating ideas of this method.

Recent advances have been to implement the algorithms without storing and using the whole inverse Hessian. For large systems, the Hessian can be quite large, which prevents its storage. This drawback has limited the use of quasi-Newton methods. The resolutions of this problem are known as limited memory quasi-Newton algorithms[32]. As originally implemented, the D. D. Johnson algorithm was similar in design for this purpose. The algorithm only uses a set amount of past gradient and search information. This information is then used to calculate the action of the inverse Hessian on the current gradient in order to get the next step size.

Eigenvalue Problem and Minimization

The ground-state total energy, E_T, and charge density, n(**r**), of a molecule or condensed matter system are fundamental quantities of a system. Within an independent electron picture, DFT is a common method to calculate these quantities. There are two main variational formulations for these calculations – density matrix[12–19] and orbital[1–11] approaches. Both give the same total energy and charge density but obtain them in slightly different ways. With regards to notation, bold face will be used for vectors and matrices.

The orbital formulation for systems with a gap traditionally solves a generalized matrix eigenvalue problem

$\mathbf{H}\Psi=\mathbf{S}\Psi\mathbf{E}$. (3.8)

For a given representation (primitive basis set of M functions), **H** is the M x M Hamiltonian matrix. **S** is the overlap matrix, and equals **I**, the unit matrix, in an orthonormal basis. Ψ is the M x N matrix comprised of the expansion coefficients of the M basis functions for the N lowest normalized eigenvectors of **H**. Ψ diagonalizes **H**, creating the diagonal N x N matrix **E**, and defines the occupied vector space. Generally each eigenvector is delocalized making Ψ a dense matrix.

When calculating E_T and n(**r**) of systems with a gap, all of the eigenvectors are equally weighted. This is not true for metals, which may have partial weighting, or occupancy. Only the collective properties of the eigenvectors are necessary to obtain E_T and n(**r**). The system can now be analyzed as a single entity and not in terms of its components, the eigenvectors. We define Ψ(r) to be the projection of the occupied space onto real space; therefore, if we approximate real space by a mesh of 100 points, $\Psi(\mathbf{r})$ would be a matrix of 100 x N. One can obtain E_T from the band energy, E_B, and n(**r**) using

$E_B = \mathrm{Tr}[\mathbf{E}]$ (3.9)

and

n(**r**) = diagonal matrix elements of $[\Psi(\mathrm{r})\ \Psi^{\dagger}(\mathbf{r}')]$, thus $\mathbf{r}=\mathbf{r}'$. (3.10)

This formulation can alleviate the burden of orthogonalization in the following way. For transformations $\Psi\mathbf{A} = \Phi$ with **A** being any N x N non-singular matrix, Φ spans the same space as Ψ. E_B and n(**r**) are invariant as long as the equations are generalized to handle nonorthogonality and lack of explicit normalization. The essential information, E_B and n(r), traditionally obtained by diagonalization are now obtained in a different manner that may prove to be more desirable in certain situations.

For the N lowest eigenfunctions, the matrix eigenvalue problem is exactly equivalent to minimizing the trace of a generalized Rayleigh quotient[33]

$E_B(\Phi) = \mathbf{Tr}[(\Phi^{\dagger}\mathbf{S}\Phi)^{-1}\ \Phi^{\dagger}\mathbf{H}\Phi\ ]$ (3.11)

w.r.t. Φ, under the constraints that $\Phi^{\dagger}\mathbf{S}\Phi = \mathbf{I}$, and $\Phi^{\dagger}\mathbf{H}\Phi=\mathbf{E}$ is a diagonal matrix.[7,8] As we desire only the **total** electronic energy and charge density of the entire system and not individual states, these constraints can be lifted and the results are unaltered as long as $\Phi^{\dagger}\mathbf{S}\Phi$ is not a singular matrix. The result is that Φ can be nonorthogonal and most importantly for our purposes be made local (see Sec. 3.1.2 and 4.1.2) with little sacrifice of accuracy if done properly. The familiar formula for the charge density, n(**r**), does have to be altered.

n(**r**) = diagonal matrix elements of $[\Phi(\mathbf{r})\ (\Phi^{\dagger}\mathbf{S}\Phi)^{-1}\ \Phi^{\dagger}(\mathbf{r}')]$[34] (3.12)

The density matrix formulation also obtains information only about the entire system and not individual states. The band energy, Eq. (3.13), is minimized w.r.t. the density matrix **P**, a hermitian M x M matrix.

$E_B(\mathbf{P}) = \mathrm{Tr}[\mathbf{PH}]$, (3.13)

As $\mathbf{P} = \Phi(\Phi^{\dagger}\mathbf{S}\Phi)^{-1}\Phi^{\dagger}$ and from the relation Tr[**AB**] = Tr[**BA**], we can see that Eq. (3.11) and Eq. (3.13) are equivalent. The matrix **P** is obtained directly without having to calculate Φ; but the idempotent constraint **PSP=P**, which is automatic when $(\Phi^{\dagger}\mathbf{S}\Phi)^{-1}$ is calculated, must be imposed. This is usually done by utilizing (see Sec. 3.3.1) the Mcweeny purification[35]

$\mathbf{3PSP} - \mathbf{2PSPSP} \rightarrow \mathbf{P}$ (3.14)

Iterations of this formula will bring a nearly idempotent **P** to idempotency. This is accomplished by driving the eigenvalues of **P** to 1 and 0 respectively for occupied and unoccupied orbitals. The polynomial in Eq. (3.14) can replace[12] **P** in Eq. (3.13) or be used iteratively[16] with a shifted and scaled **H** to get the proper **P**, or the two can be used in conjunction.[15] If the Mcweeny purification with the shifted and scaled **H** is used by itself, it is non-variational.

Linear Scaling and Varying Length Scales

A great utility of these minimization techniques is their ability to take advantage of a well-known chemical intuition recently codified as the "near-sightedness" principle.[19] This principle basically states the following: if two regions of charge are far enough apart, one region should not feel any change in the local environment of the other one. Within the orbital formulation, the interaction distance between electrons is made finite, as only certain matrix elements of Φ are designated non-zero by the use of a localization region. The fundamental properties of the system, not convenient numerical artifacts, should decide the localization. If the localization is imposed at very short length scales, accuracy is lost due to the neglect of significant interactions. Every calculation needs to ascertain the quantitative or qualitative accuracy desired in order to localize at the proper length scale.

The implementation of this principle requires using a basis set whose elements are strictly localized in real space, e.g. finite elements[36,37] or a tight-binding basis[2,6], or pseudo-localized in real space, e.g. gaussians made local by the use of cutoffs. Using a local basis, **H** and **S** become sparse as a matrix element is exactly equal to, or set to zero once a sufficient separation distance between the basis functions is reached. The desired accuracy dictates this interaction distance. Some specifics in the O(N) calculation of **H**[38−43] will be discussed in Chapter 4.

For increasing system size, Φ is made sparse first while still allowing **P**, used in the calculation of the total energy and charge density, to be quite extended. As the system becomes larger, **H**, **S**, and **P** sparsify next. $\Phi^\dagger\mathbf{S}\Phi$ and $\Phi^\dagger\mathbf{H}\Phi$ become sparse last due to the interaction of the occupied orbitals mediated by **H** and **S**. Since sparsity occurs at different system sizes (length scales), it is advantageous for the algorithm to exploit the sparsity at different stages.

With dense matrices, matrix multiplications take O(N^3) operations. If the matrices are sparse, the matrix multiplications can be carried out in O(N) steps by calculating only the elements that are non-zero. A matrix needs to be significantly sparse, about 10% of the elements being non-zero, before sparse routines become faster than machine-dependent optimized dense routines. For the system sizes studied, the sparsity of $\Phi^\dagger\mathbf{S}\Phi$ and $\Phi^\dagger\mathbf{H}\Phi$ did not warrant sparse multiplications; therefore, they were kept dense. A similar observation was noted in Ref. [9].

Geometry of the Vector Space: the Grassmann Manifold

The Grassmann manifold[20,21] of rank N is the set of all subspaces of rank N in some ambient (primitive) M dimensional space. There are two common representations used in physics:

Density Matrix: **P** - a hermitian M x M matrix with the idempotent constraint **PSP=P**

Orbital:

Orthogonal Ψ - a M x N matrix with the orthogonality constraint of $\Psi^\dagger\mathbf{S}\ \Psi = \mathbf{I}$

Non-orthogonal Φ - a M x N matrix with the constraint that $(\Phi^\dagger\mathbf{S}\Phi)^{-1}$ is non-singular.

We start with a description of the Grassmann manifold through the ground-state density matrix formulation. The search is in the M(M+1)/2-dimensional space of all hermitian M x M matrices. With the number of electrons conserved, each matrix is a point in this space and gives a unique electronic configuration for the 2N electrons, the occupied space. An electronic configuration defines the occupation magnitude of each eigenvector in the M-dimensional basis space. To account for any possible hermitian M x M matrix, the occupation magnitude could be any value. Most of these occupation values give unphysical solutions. A ground-state density matrix must satisfy idempotency. The Grassmann manifold, idempotent surface[15], corresponds to the points that satisfy this condition.

The description for the orbital formulation is more involved. For notational and conceptual reasons, we rearrange Eq. (3.11) by defining

$\bar{\mathbf{\Phi}}^\dagger = (\Phi^\dagger\mathbf{S}\Phi)^{-1}\ \Phi^\dagger$ (3.15)

which gives

$\mathbf{E}_\mathrm{B}(\bar{\mathbf{\Phi}}^\dagger,\Phi) = \mathbf{Tr}[\bar{\mathbf{\Phi}}^\dagger \mathrm{H}\Phi]$ (3.16)

as our functional to be minimized. Here we choose to have $\bar{\mathbf{\Phi}}^\dagger$ as our variable instead of $\bar{\mathbf{\Phi}}$as it is always the transpose that appears in our equations. Eq. (3.16) is in a dual basis[44−46] form where $\bar{\mathbf{\Phi}}^\dagger$ is the covariant matrix, one-form[47] or linear-form[48], of the matrix Φ. Biorthogonality, of which orthogonality is a special case, is automatically satisfied if the inverse is calculated in Eq. (3.15)

$\bar{\mathbf{\Phi}}^\dagger\mathbf{S}\Phi = (\Phi^\dagger\mathbf{S}\Phi)^{-1}\Phi^\dagger\mathbf{S}\Phi = \mathbf{I}.$ (3.17)

The points in space are still defined by our occupied space, but the occupied space is now defined by $(\bar{\mathbf{\Phi}}^\dagger,\Phi)$ and is related to **P** through

$\mathbf{P} = \Phi\bar{\mathbf{\Phi}}^\dagger.$ (3.18)

If $\bar{\mathbf{\Phi}}^\dagger$ and Φ are biorthogonal, then **P** is idempotent and Tr[**PS**]=2N; therefore, the point (**P** or $(\bar{\mathbf{\Phi}}^\dagger,\Phi)$) resides on the Grassmann manifold. There is an equivalence relation on the manifold. Given a nonsingular M x M matrix **A** and $\bar{\mathbf{\Phi}}^\dagger_\mathrm{A}$, the biorthogonal complement of $\Phi\mathbf{A}$, $(\bar{\mathbf{\Phi}}^\dagger,\Phi)$ and $(\bar{\mathbf{\Phi}}^\dagger_\mathrm{A},\Phi\mathbf{A})$ through Eq. (3.18) create the same **P** and thus define the same point on the Grassmann manifold.

An equivalent perspective is for each point to be defined by Φ and a covariant metric of the occupied subspace, which creates $\bar{\mathbf{\Phi}}^\dagger$ from Φ. If the correct metric is used the point lies on the Grassmann manifold. So it is in this space, Φ and a metric, that we minimize $\mathrm{E}_\mathrm{B}(\Phi)$, w.r.t. Φ, under the constraint that the minimum lies on the Grassmann manifold. The reason for introducing a constraint manifold into a previously unconstrained problem is that in an asymptotically linear scaling algorithm the exact covariant metric of the occupied subspace, $(\Phi^\dagger\mathbf{S}\Phi)^{-1}$, is not calculated. Algorithms must address the possible departure and return to the manifold.

Satisfying Constraints

The task of bringing the density matrix to reside on the Grassman manifold, idempotency surface, after each update of **P** has been described as translations, a Mcweeny path[15], due to repeated iterations of Eq. (3.14). Using sparse multiplications, this is an O(N) process. If Eq.

(3.14) is used only to replace $\mathbf{P}$ inside of Eq. (3.13) creating a new functional, the path will ideally move in close proximity to the Grassman manifold and the minimum will be found on the manifold. If the path wanders too far away from the manifold, some eigenvalues of $\mathbf{P}$ may diverge to $-\infty$ or $+\infty$.[12]

Satisfying biorthogonality, adhering to the Grassmann manifold, for system sizes where $(\Phi^\dagger \mathbf{S}\Phi)$ is not appreciably sparse (see Sec. 3.2) is straightforward as one should calculate a dense $(\Phi^\dagger \mathbf{S}\Phi)^{-1}$. For larger systems, however, the $O(N^3)$ scaling of the dense inverse dominates. One alternative to calculating the complete inverse is to calculate only selected parts of the inverse that fall within a cutoff radius by use of a conjugate gradient algorithm, thereby obtaining a sparse inverse and achieve linear scaling through sparse matrix multiplies. Other similar methods, which obtain a sparse factorization of $\mathbf{S}^{-1}$, make use of an incomplete Cholesky factorization[17] or the AINV[49] algorithm.

An alternative to calculating an approximate covariant metric is to fix the metric to be identity and iterate Φ to satisfy $\Phi^\dagger \mathbf{S}\Phi = \mathbf{I}$. This is similar in manner to the Mcweeny transformation for the density matrix. The iterative formula,

$\Phi\ (\mathbf{3/2\ I} - \mathbf{1/2}\ \Phi^\dagger \mathbf{S}\Phi) \rightarrow \Phi$ (iterative polar decomposition),[21] (3.19)

drives the matrix square root of $(\Phi^\dagger \mathbf{S}\Phi)$ to be $\mathbf{I}$.

A third option is to use

$\mathbf{2D\text{-}D}(\Phi^\dagger \mathbf{S}\Phi)\mathbf{D} \rightarrow \mathbf{D}$ (Schultz's or Hotelling's method) (3.20)

to drive $\mathbf{D}$, an approximate sparse inverse of the overlap in the occupied space, to a more exact inverse. It could also be used as a replacement for the inverse. In this case the occupied space would conform to the chosen metric $\mathbf{D}$. The choice of $\mathbf{D}=\mathbf{I}$ results in orthogonal orbitals.

The fourth option is to use a truncated power series expansion to any odd order as an approximate $(\Phi^\dagger \mathbf{S}\Phi)^{-1}$. For $\mathbf{D}=\mathbf{I}$[2,3], Eq. (3.20) is the series expansion to first order. This approximate inverse becomes the exact inverse once the function finds the appropriate minimum. For higher orders, evidence[52] suggests the orbitals do not orthogonalize, but are kept from de-orthogonalizing. So at the minimum, biorthogonality is achieved for orbitals that are slightly nonorthogonal.

It is not plainly evident which method is best. A nonorthogonal orbital method might be preferable over orthogonal methods given the assumption that nonorthogonal orbitals are more localized.[50,51] The convergence rate for the minimization is slower for the algorithms that are allowed to wander off of the Grassman manifold.[52] This effect was only studied for dense linear algebra, but the effects should carry over to sparse linear algebra. In order to adhere strictly to the Grassman manifold, we calculate the dense $(\Phi^\dagger \mathbf{S}\Phi)^{-1}$, because $\Phi^\dagger \mathbf{S}\Phi$ was not appreciably sparse.

Grassmann Conjugate Gradient (GCG) Algorithm

Within the orbital formulation, a Grassman conjugate gradient algorithm[20,21] is used to minimize Eq. (3.11). The discussion here is for dense matrices. The aim is to introduce the algorithm without the complexity of localization. The algorithm is exactly the same as described in Ref. [21] with the addition of the parallel translation of the gradient.

Properties of a Gradient

We need to define a gradient, $\mathbf{G}=\nabla_\Phi \mathrm{E_B}(\Phi)$, to be used in producing new conjugate search directions to minimize $\mathrm{E_B}(\Phi)$. In [21], it was noted that the gradient of a functional should not be confused with its differential. The gradient must be in the direction of the greatest change of the functional. Even an infinitesimal movement in the direction of the gradient must cause $\mathrm{E_B}$ to change in value; therefore, the gradient must have no component along Φ because such a direction would not change the value of $\mathrm{E_B}$. This is a property of the functional and does not depend upon the representation. For this to be true, the gradient must lie in the tangent plane[20] of Φ, a constraint given by

$\Phi^\dagger \mathbf{SG} = \mathbf{0}$ and equivalently $\bar{\mathbf{\Phi}}^\dagger \mathbf{SG} = \mathbf{0}.$ (3.21)

A gradient requires a well-defined inner product. For an inner product $\langle \bullet \rangle$ and an infinitesimal step s in the direction of a displacement $\delta\mathbf{V}$ in the tangent plane, the inner product of the gradient with $\delta\mathbf{V}$ must equal the directional derivative in $\delta\mathbf{V}$, given by

$\langle \mathbf{G} \bullet \delta\mathbf{V}\rangle = \delta\mathbf{E}_\mathrm{B} = \mathbf{d/d}s\ \mathbf{E}_\mathrm{B}(\Phi + s\delta\mathbf{V})|_{s=0}.$ (3.22)

The only form that keeps the inner product invariant between arbitrarily complete representations, for vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ defined for Φ at the origin, is

$\langle \mathbf{X}_1 \bullet \mathbf{X}_2\rangle = \mathrm{Real}\{\mathrm{Tr}[(\Phi^\dagger \mathbf{S}\Phi)^{-1}\mathbf{X}_1^\dagger \mathbf{S}\mathbf{X}_2]\}.$ (3.23)

This gives $\langle \Phi \bullet \Phi\rangle = \mathbf{N}$ which conserves electron number which also is necessary for points on the Grassmann manifold.

Gradient, Search Direction, and Line Minimization

Starting with the differential of our functional[52]

$\mathbf{dE}_\mathrm{B}/\mathbf{d}\Phi = (\mathbf{I} - \mathbf{S}\Phi\bar{\mathbf{\Phi}}^\dagger)\mathbf{H}\Phi(\Phi^\dagger \mathbf{S}\Phi)^{-1}$ (3.24)

one can see that this does not satisfy Eq. (3.21). The gradient

$\mathbf{G} = (\mathbf{S}^{-1} - \Phi\bar{\mathbf{\Phi}}^\dagger)\mathbf{H}\Phi$ (3.25)

does satisfy Eq. (3.21). The differential, Eq. (3.24), in effect, lies on a cone around the gradient. This can be graphically seen in Fig. 3 of White et al.[53] It may provide search directions that are suitable for minimization, but the convergence will be degraded, as the gradient is not being used. For reasons of efficiency, the differential may be preferable if incorporating $\mathbf{S}^{-1}$ is too expensive. This may be the case for finite elements[36,37] or Mehrstellen[54] finite difference representations, for which the dimension of $\mathbf{S}$ is typically larger than for an LCAO basis.

The new search direction, $\mathbf{Z}_{I+1}$, was updated by the Polak-Ribière (PR) formula[20],

$\mathbf{Z}_{I+1} = \mathbf{G}_{I+1} + [\ \langle\ (\mathbf{G}_{I+1} - \mathbf{G}_I)\ \bullet\mathbf{G}_{I+1}\rangle\ /\ \langle\mathbf{G}_I\ \bullet\mathbf{G}_I\ \rangle\ ]\ \mathbf{Z}_I.$ (3.26)

This formula gave slightly better convergence than the Fletcher-Reeves (FR) form. A step size, Λ, in the search direction, $\mathbf{Z}$, must be chosen to attempt to minimize the functional in that direction. A quadratic approximation was used around the current point Φ for the step size of Λ in the direction $\mathbf{Z}$.

$\mathbf{E}_B(\Phi+\ \Lambda\mathbf{Z}) = \mathbf{E}_B(\Phi) +\ \Lambda\ \mathbf{Z}\ \bullet\mathbf{dE}_B/\mathbf{d}\Phi\ +\ {}^1\!/\!_2\quad \Lambda\ \mathbf{Z}\ \bullet\boldsymbol{H}\ \bullet\Lambda\ \mathbf{Z}$ (3.27)

The second term is just the normal dot product of $\mathbf{Z}$ with the differential. The last term here is the matrix element, $H(\mathbf{Z},\mathbf{Z})$ of the Grassmann Hessian. The general formula of the matrix element of the Grassmann Hessian given two tangent vectors $\mathbf{V}_1$ and $\mathbf{V}_2$, is

$\boldsymbol{H}\ (\mathbf{V}_1,\ \mathbf{V}_2) = \mathbf{Tr}[\bar{\mathbf{V}}_1^{\dagger}\mathbf{H}\mathbf{V}_2 - \bar{\mathbf{V}}_1^{\dagger}\mathbf{S}\mathbf{V}_2\ \bar{\boldsymbol{\Phi}}^{\dagger}\mathbf{H}\Phi].$[52] (3.28)

The overall cost of the calculation of $H\ (\mathbf{Z},\mathbf{Z})$ is small as compared to the two most expensive parts $\mathbf{HZ}$ and $\mathbf{SZ}$, which are already necessary for the next functional evaluation.

So to get Λ, we set $\mathrm{dE_B}(\Phi+\ \Lambda\mathbf{Z}\ )/\mathrm{d}\Lambda = 0$.

The step size can now easily be calculated by using Eq. (3.27), and we obtain

$\Lambda\ \ = \mathbf{Z}\ \bullet\ \mathrm{dE_B}/\mathrm{d}\Phi\ /\ H\ (\mathbf{Z},\mathbf{Z}).$ (3.29)

Φ is updated according to

$\Phi_{\mathrm{NEW}} = \Phi - \Lambda\mathbf{Z}$ (3.30)

Parallel Transport and Convergence

Since the search direction should stay in the plane tangent to Φ, the current $\mathbf{Z}$ needs to be orthogonal to Φ_{NEW} so that when Eq. (3.26) is used in the I+1 GCG iteration $\mathbf{Z}_{I+1}$ will be orthogonal to Φ_{NEW}. In this manner, Hessian information will be properly communicated from one iteration to the next. This is accomplished to all orders by a parallel transport of $\mathbf{Z}$ corresponding to the update in Φ. $\mathbf{G}_I$ should also be translated. In Eq. (3.26), $\mathbf{Z}_I^{\mathrm{NEW}}$ and $\mathbf{G}_I^{\mathrm{NEW}}$ replace $\mathbf{Z}_I$ and $\mathbf{G}_I$ respectively.

$\mathbf{Z}_I^{\mathrm{NEW}} = \mathbf{Z}_I + \Lambda\quad \Phi\bar{\mathbf{Z}}_I^{\dagger}\mathbf{S}\mathbf{Z}_I$ (3.31)

$\mathbf{G}_I^{\mathrm{NEW}} = \mathbf{G}_I + \Lambda\quad \Phi\bar{\mathbf{Z}}_I^{\dagger}\mathbf{S}\ \mathbf{G}_I$ (3.32)

The convergence of the algorithm can be tracked in two ways. The algorithm can be terminated when the change in $\mathrm{E_B}(\Phi)$ becomes smaller than a prescribed threshold. Termination can also occur once the norm of the gradient,

$\mathrm{Real}\{\mathrm{Tr}[(\Phi^{\dagger}\mathbf{S}\Phi)^{-1}\mathbf{G}^{\dagger}\mathbf{S}\mathbf{G}]\}$ (3.33)

dips below a certain threshold. We have chosen the latter. In self-consistent field (SCF) calculations, one must update the Coulomb and the exchange-correlation potential by the charge density of the new orbitals. We have chosen to do this only after the norm of the gradient becomes sufficiently small for a given potential. For dense matrices a value of 10^{-10} and a maximum iteration number of 15 was sufficient to obtain results within a few μRy of diagonalization.

This orbital minimization method has utility in any eigenvalue problem that extensively uses iterative solvers. In electronic structure, this comprises representations that use a much larger number of basis functions than occupied states, e.g. plane waves[7,10,11], finite difference[54–56], and finite elements.[36,37]

[1] E. B. Stechel, A. R. Williams, and P. J. Feibelman, Phys. Rev. B **49**, 10 008 (1994).

[2] P. Ordejón, D. A. Drabold, R. M. Martin, et al., Phys. Rev. B **51**, 1456 (1995).

[3] F. Mauri and G. Galli, Phys. Rev. B **50**, 4316 (1994).

[4] W. Hierse and E. B. Stechel, Phys. Rev. B **50**, 17 811 (1994).

[5] J. Kim, F. Mauri, and G. Galli, Phys. Rev. B **52**, 1640 (1995).

[6] U. Stephan, D. A. Drabold, and R. M. Martin, Phys. Rev. B **58**, 13 472 (1998).

[7] M. P. Teter, M. C. Payne, and D. C. Allan, Phys. Rev. B **40**, 12 255 (1989).

[8] M. J. Gillan, J. Phys.: Condens. Matter **1**, 689 (1989).

[9] G. Galli and M. Parrinello, Phys. Rev. Lett. **69**, 3547 (1992).

[10] R. D. King-Smith and D. Vanderbilt, Phys. Rev. B **49**, 5828 (1994).

[11] I. Stich, R. Car, M. Parrinello, et al., Phys. Rev. B **39**, 4997 (1989).

[12] X.-P. Li, R. W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10 891 (1993).

[13] M. S. Daw, Phys. Rev. B **47**, 10895 (1993).

[14] R. W. Nunes and D. Vanderbilt, Phys. Rev. B **50**, 17 611 (1994).

[15] D. R. Bowler and M. J. Gillan, Comp. Phys. Com. **120**, 95 (1999).

[16] A. H. R. Palser and D. E. Manolopoulos, Phys. Rev. B. **58**, 12 704 (1998).

[17] J. M. Millam and G. E. Scuseria, J. Chem. Phys. **106**, 5569 (1997).

[18] M. Challacombe, J. Chem. Phys. **110**, 2332 (1999).

[19] W. Kohn, Phys. Rev. Lett. **76**, 3168 (1996).

[20] A. Edelman, T. A. Arias, and S. T. Smith, SIAM J. on Matrix Anal. Appl. **20**, 303 (1998).

[21] R. A. Lippert and M. P. Sears, Report SAND99-2986, Sandia National Laboratories, Abuquerque, NM USA, (1999).

[22] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, Science **220**, 671 (1983)

[23] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C* (Cambridge University Press, 1995).

[24]R.P. Brent, *Algorithms for Minimization without derivatives* (Engewood Cliffs, NJ: Prentice-Hall, 1973), Chapter 5.

[25]J.J. More and D.J. Thuente (1990), "On line search algorithms with guaranteed sufficient decrease:, Mathematics and Computer Science Division Preprint MCS-P153-0590, Argonne National Labaoratory (Argonne, IL)

[26]C.G. Broyden, Math. Comp. **21**, 368 (1967)

[27]D. Goldfarb, Math. Comp. **24,** 23 (1970)

[28]D.F. Shanno, Math. Comp. **24,** 647 (1970)

[29]P.Pulay, Chem. Phys. Lett. **73,** 393 (1980)

[30]D.D. Johnson, Phys. Rev. B **38,** 12 807 (1988)

[31]D. Vanderbilt and S.G. Louie, Phys. Rev. B **30**, 6118 (1984)

[32]J. Nocedal, Math. Comp. **35**, 773 (1980)

[33]A. Edelman and S. T. Smith, BIT **36**, 494 (1996).

[34]P.-O. Löwdin, Int. J. Quantum Chem. **2**, 867 (1968).

[35]R. Mcweeny, Rev. Mod. Phys. **32**, 335 (1960).

[36]J. Pask, B. M. Klein, C. Y. Fong, et al., Phys. Rev. B **59**, 12 352 (1999).

[37]E. Tsuchida and M. Tsukada, J. Phys. Soc. Japan **67**, 3844 (1998).

[38]P. Schultz and P. J. Feibelman, to be published.

[39]C. Ochsenfeld, C. A. White, and M. Head-Gordon, J. Chem. Phys. **109**, 1663 (1998).

[40]K. N. Kudin and G. E. Scuseria, Chem. Phys. Lett. **289**, 611 (1998).

[41]E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **106**, 9708 (1997).

[42]R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **257**, 213 (1996).

[43]J. M. Perez-Jorda and W. Yang, Chem. Phys. Lett. **241**, 469 (1995).

[44]O. Sinanoglu, Theoret. Chim. Acta **65**, 233 (1984).

[45]E. Artacho and L. M. d. Bosch, Phys. Rev. A **43**, 5770 (1991).

[46]E. B. Stechel, in *Topics in Computational Materials Science*, edited by C. Y. Fong (World Scientific, 1998), p. 1.

[47]B. F. Schutz, *Geometrical methods of mathematical physics* (Cambridge University Press, 1980).

[48]R. W. R. Darling, *Differential Forms and Connections* (Cambridge University Press, 1994).

[49]M. Benzi, C. D. Meyer, and M. Tuma, SIAM J. Sci. Comput. **17**, 1135 (1996)

[50]S. Liu, J. M. Perez-Jorda, and W. Yang, J. Chem. Phys. **112**, 1634 (2000).

[51]P. W. Anderson, Phys. Rev. Lett. **21**, 13 (1968).

[52]R. A. Lippert and M. P. Sears, submitted to Phys. Rev. B, (2000).

[53]C. A. White, P. Maslen, M. S. Lee, et al., Chem. Phys. Lett. **276**, 133 (1997).

[54]E. L. Briggs, D. J. Sullivan, and J. Bernholc, Phys. Rev. B **54**, 14 362 (1996).

[55]J. R. Chelikowsky, N. Troullier, and Y. Saad, Phys. Rev. Lett. **72**, 1240 (1994).

[56]N. A. Modine, G. Zumbach, and E. Kaxiras, Phys. Rev. B **55**, 10 289 (1997).

.

SPARSE IMPLEMENTATION

We used a Gaussian-based LCAO method as implemented in the serial DFT code SEQQUEST[1] as the framework to implement the Grassmann conjugate gradient (GCG) minimization algorithm. Our calculations use a Troullier and Martins[2] pseudopotential for oxygen and the Hamann[3] pseudopotentials for the other atoms. A split-valence double zeta with polarization (DZP) basis was used. This basis set was optimized for accuracy with no concern for sparsity for **H** or **S**.

Each basis function is a single gaussian, $e^{-\alpha r^2}$, or possibly a *contracted* gaussian (a sum of gaussian functions) and a spherical harmonic for the angular dependence. A DZP basis set consists of 2 basis functions for every occupied atomic shell of an atom. We will refer to a *shell* as the set of basis functions with the same radial gaussian term and L value for the spherical harmonic but differing by the m value. A shell with L=0 would have one basis function associated with it, and a shell with L=1 would have 3 basis functions associated with it.

The single zeta shell is comprised of *contracted* gaussians with three to four gaussians that are short-ranged and thus give variational flexibility near the atom. The double zeta shell has a single gaussian with a smaller α making it longer ranged. The polarization shell describes the lowest L value of an unoccupied atomic shell. So the polarization shell for silicon would have L=2. With regards to terminology a gaussian will refer to one functional component of a basis function, also referred to as a *contracted* gaussian. The localized non-orthogonal orbitals, which define the *occupied* space, are linear combinations of *contracted gaussians* and are referred to as orbitals.

A large amount of infrastructure must be in place before the Grassmann conjugate gradient algorithm can be implemented with localization. The memory structure of each matrix must be reconfigured in order to account for sparsity. Each matrix is stored as a one-dimensional array containing the non-zero matrix elements, whose position and memory location are accessed by a sparsity list (described in Sec. 4.1). The multiplications of matrices must efficiently account for this sparsity.

Another critical issue that the matrix multiplications must handle is the growth of the matrix resulting from a multiply. If the multiplications calculated every possible non-zero element with every multiply, after a few multiplies the matrices would soon be nearly dense and negate the O(N) scaling. Thus the multiplications also must have a reasonable method for controlling the size of the matrix resulting from the multiply.

The overall managing of memory is mostly straightforward. The memory for each routine comes from a large workspace created at the start of the code. Before each sparse routine the segments of the workspace are allotted the proper size for the data structures necessary for that routine. The one problem when a matrix or sparsity list is first created is that one does not know *a priori* the size of the matrix or sparsity list. In order to keep the overall usage of memory O(N) one must use initial estimates, which are sufficiently large, for the sizes of the matrices and the sparsity lists.

Storage

The sparse storage of the M x M (number of basis functions) matrices **H**, **S,** and **P** was easily adapted from the linear scaling dense creation scheme.[1] The code has input cutoff values that determine if matrix elements are non-zero. The default values give total energies that are converged with respect to the cutoff values to within μRy/atom. **S** and its sparsity list are calculated first. The sparsity list of **S** is also used for the storage of **H** and **P**. It is possible that some elements of **H** are lost by the imposition of the sparsity pattern of **S**, but the effects are negligible with the above cutoffs. Using the same sparsity patterns saves memory and simplifies the calculations.

The matrices **H**, **S**, and **P** are all stored in a shell-blocked row format. The " shell-blocked" term refers to the fact that the data is stored by shell. So if l=1 for the row shell, the data is stored in blocks of 3 x (2*L of the column shell+1). A whole shell blocked row is a matrix of 3 x (number of columns with a non-zero value). The data is accessed with a pointer to the block of data, which gives the array number for the first element in that block of data. This information along with the size of the data block and the corresponding column shell for the data block is stored in the sparsity list. The list is a 3 x number of maximum neighbors x number of shells. The memory asymptotically scales linearly with system size, as the first 2 indices are constant.

The M x N matrices, Φ, **Z,** and **G,** have the same sparsity pattern determined by inputting strict distance cutoffs. Only gaussians within the localization sphere, measured from the center of an orbital (atom-centered or bond-centered) contribute to that orbital. For atom centered orbitals, several orbitals may share the same localization region. The code takes advantage of this property to cut down on storage and increase efficiency in matrix multiplies.

The option is available to only select certain shells from an atom. This means that the localization radius for the contribution of the s-shells, p-shells, and d-shells can differ. The reasoning was that each shell has a different spatial extent. If each shell has the same spatial extent, the code could be more efficient by taking this into account, although it currently does not. There are also M x N matrices of the type $\mathbf{S}\Phi$ or $\mathbf{H}\Phi$, which have more elements than Φ. These are termed *growth* matrices, as their effective localization radius grows larger than the localization radius of Φ due to the action of **S** or **H**.

All of the matrices of M x N are stored in a column format. A data block is stored for each shell contributing to an orbital, one column. The size of the data blocks is 2*L+1, for L the angular momentum of the contributing shell. The whole set of data for a given localization region, e.g. SiC with 4 orbitals on the carbon atoms, is viewed as a matrix of dimension (number of contributing basis functions) x 4. The data in the sparsity lists are again stored in a similar fashion as for the primitive matrices. The only difference is that now there are two lists necessary due to them not being square matrices (see Sec. 4.1.2). The sparsity lists for the larger of these type of matrices, $\mathbf{S}\Phi$, are structured the same as for Φ, but the lists just include more neighbors to account for the growth.

The sparsity pattern of the M x N matrices of the type $\mathbf{S}\Phi$ can be determined in two ways. The first method is by input of cutoffs exactly as for Φ. In the second method, an element is kept if its value is above an input threshold value. In the results presented in Chapter 5, the sparsity pattern was calculated and held fixed for each SCF cycle. The initial SCF cycle used the first method and subsequent cycles used the second method. This was done, as the initial estimate of Φ was not sufficient for the second method.

Storage of S and H

The main change to the existing routine, *SIJ*, for the dense calculation of the overlap matrix to form *SIJ_sp* is the storage of the data in a sparse format and the creation of the list, *list_pr_pr*, to access the data properly. The parameter *ns_cut* is the initial estimate for the average number of non-zero columns for a given row, which keeps the total memory usage O(N). The number of shells with which a shell overlaps significantly is stored in the array *nhbr_pr_pr*, and the number of total gaussians is stored in the array *nhbr_bas*. Since the significance, allowed to be non-zero, is evaluated on a shell by shell basis the sparsity list reflects this.

The index of the row and column shells being evaluated for significance is tracked in *ishell* and *jshell* respectively. The overlap elements for the data of *ishell* with the shells from j atom are stored temporarily in *s_sp_wrk*, comprising a block of data for each shell of atom j. If *ishell* and *jshell* overlap significantly then the array element *n_elem(jshell)* contains the number of elements for that block of data. If the shells do not overlap significantly, this array element is zero as it is never incremented. The significance of the overlap is determined by an equation involving the smallest exponential coefficients of a gaussian of the respective shells and the distance between the atoms on which they sit.

All of the shells for atom j are investigated and if *n_elem(jshell)* is non-zero then the proper data in *s_sp_wrk* is stored into the *s_sprs_tmp*. The proper counter arrays are incremented and *list_pr_pr_tmp* gets the proper values. It has three indices. The 3^{rd} array runs over all gaussian shells of the system. The next array runs over the number of shells with which each one overlaps significantly. When declared this 2^{nd} index has the size *nbr_max*, the largest number of neighbors for any shell. It is a fixed number, but its value is not known. The value is guessed to be *nlpp_cut,* which must be set larger than the resulting *nbr_max.* The 1^{st} index is of size three. The 1^{st} element gives the shell that a given neighbor index corresponds. The 2^{nd} index gives the pointer value to the starting point of the corresponding block of data in the overlap matrix. The 3^{rd} index gives the number of matrix elements in this block.

At this point there is no uniform structure to how the data is stored in the overlap matrix. The data is stored in blocks, but as we sweep through the storage we would jump all around a 2-dimensional map of the dense matrix. For the $\mathbf{S}\Phi$ matrix multiplication it is more efficient to store the data in a shell-blocked row format. We want the data accessed in a shell-blocked row to be consecutive memory addresses. The data will be loaded into cache quicker in preparation for a shell-blocked row of $\mathbf{S}$ to multiply the orbitals of a common localization region. This is accomplished by the routine *ROW_STORE_PR* within *AUX_SIJ_SP*. The sparsity list and the data storage are both updated. The overlaps of basis functions on an atom, *s1atom*, are also calculated in *AUX_SIJ_SP*. This is necessary for a correction[4] to the self-consistent potential, which compensates that the charge and self-consistent potential is evaluated on a uniform grid, see Sec 4.4.

The creation of the sparse Hamiltonian directly uses the sparse list for the overlap matrix. Each dense routine of the Hamiltonian creation has a sparse counterpart. So if *jshell* interacts significantly with *ishell*, *n_elem(jshell)* is non-zero. The proper placement of this block of data is found by the routine *locate.* This routine returns the neighbor number, *n_jshell*, that *jshell* is for *ishell.* The array element *list_pr_pr(2,n_jshell,ishell)* then gives the pointer to the data.

The significance of interaction in *TIJ_SP*, calculation of the kinetic energy matrix elements, is evaluated exactly as in *SIJ_SP.* In *ENII_SP* and *ENIJ_SP*, respectively the on-site and off-site Coulomb and exchange-correlation energy matrix elements of the atomic-reference valence charge density and the matrix elements of the local part of the pseudopotential interaction with the nuclei and core electrons are calculated. The cutoffs are again calculated in the same way with the range of the total local potential comprising the aforementioned potentials determined by 2 * (the smallest exponential coefficient for a gaussian on atom). This is the result of squaring the initial atomic wavefunctions to get the atomic-reference valence charge density. *ENNLOC_SP* calculates the non-local part of the pseudopotential energy matrix elements. The spatial extent of the non-local pseudopotentials are taken from the fits of Bachelet, Hamann, and Schluter.[5]

Initializing Φ and Calculating Growth Matrices

The sparsity list for Φ**, Z,** and **G** is determined only once. Currently, the same list is kept though the self-consistent process for a given geometry, atomic positions, and during the relaxation of the atomic positions. If the code is adapted for molecular dynamics calculations, then one might choose to update the sparsity list for Φ**, Z,** and **G**. The routine *INIT_PHI* sets up the initial guess for Φ, used as the starting point for the minimization of the band energy, Eq (3.11). It also calculates all of the information for the neighbor and sparsity lists for the Φ-like and the *growth* matrices.

As we mentioned before, the length of the data and sparsity pattern arrays are not known beforehand. Each new type of array must have some initial guess at its size. The estimate for the size of Φ is given by *nphi_cut* * (number of occupied orbitals), which needs to be greater than the calculated total elements of Φ, *n_pr_occ_elem.* Certain multiplications and operations require the data and sparsity information of Φ to be accessed for a given shell, i.e. row-wise, and for given orbital, i.e. column-wise. Thus, indexing of the elements of Φ is done by column (orbital) in *list_pr_occ* and by shell-blocked rows in *list_occ_pr.* The array of the number of shells contributing to every orbital is *nhbr_occ*, and the array the number of orbitals to which a shell contributes is *nhbr_pr.*

The 3rd index of *list_pr_occ* runs over the orbitals. The 2nd index runs over the contributing shells for an orbital. The parameter *nlpo_cut* is the estimate for the maximum number of gaussian shells contributing to an occupied orbital, *nhbr_occ_max.* The 1st element of the 1st index holds the shell for the given orbital and neighbor number. The 2nd element gives the pointer to the data block and the 3rd element gives the size of the data block, L value of the contributing shell.

The 3rd index of *list_occ_pr* runs over the shells. The 2nd index runs over the orbitals to which a shell contributes. The parameter *nlpo_cut* is the estimate for the maximum number of occupied orbitals to which a gaussian shell contributes *nhbr_pr_max.* The sparsity lists, *list_pr_occ2* and *list_occ_pr2*, for the *growth* matrices are set up in the same way. The estimates for the maximum of the 2nd indices of *list_pr_occ2* and *list_occ_pr2* are given respectively by (5**nlop_cut*) and by (5**nlpo_cut*), which must be greater than *nhbr_pr_max2* and *nhbr_occ_max2.* All of the estimates are checked that their length is sufficient after *INIT_PHI* returns to the main program.

We will now discuss the input variables that determine the sparsity list and the initial Φ. A *namelist* call inputs the variables within the module **init** found in the file **init_phi.dat**. The first variable checked is *bond*, which is a logical variable that determines if the orbitals are atom-centered, *false,* or bond-centered, *true.* For both cases an outer loop runs over all atoms; and based on the type of atom and the array *n_oc_at*, new data for the specified orbitals is created. Each element of *n_oc_at* specifies the number of orbitals on the given type of atom. For atom-centered, each number in this array gives the number of orbitals on each type of atom. For bond-centered orbitals, the number of orbitals is defined to be 4 if the value is non-zero. In this case, four orbitals are created at the bond-centers of the four neighbor atoms of the atom specified by *n_oc_at.* If bond-centered orbitals are desired for systems without sp^3 hybridization, one will need to modify the code appropriately.

In a search of the shells that will contribute, an inner loop runs over the atoms again. The radius cutoff that determines the shells that contribute is input as follows. Each type of atom has its own set of cutoffs. So for a given atom type, one now inputs a cutoff radius for each shell of every other type of atom. If one wants orbitals on a few atoms of a certain element different than other atoms of the same element then these atoms can be labeled as a different type. Currently, the maximum of shells is 6 and extra values at the end must be padded. The radius cutoff variable, *basis_cut*, is a three-dimensional array with the indices being (shell number) x (type of atom) x (type of atom). The radius cutoff for the growth matrix is input in exactly the same manner for the array *basis_cut2.*

The starting coefficients for the linear combination of contracted gaussians for the orbitals associated with the current atom are first stored in the temporary array *phi*, a 4 x (number of total basis functions). Four constitutes the current default setting for the maximum number of orbitals associated with an atom. After the inner loop over atoms is done, the non-zero values in *phi* are stored in *phi_sp.* The initial value for the coefficients of a given shell for an occupied orbital whether it is a bond orbital or an atom-centered orbital is given by the array *coeff.* The purpose of this is that if one wants to have a different amount of s and p character and for different single zeta and double zeta contributions.

For bond-centered atoms, *dist_nn* specifies the distance for an atom to be considered as a nearest neighbor for forming a bond orbital. For bond orbitals, the coefficient values entered as above are modified to point in the direction the nearest neighbor as specified by *dist_nn* on the populating atom and point back on the neighbor bond atom. If the orbitals are atom centered, the coefficients specified by *coeff* are modified by *isgn* to reflect the given hybridization. Each atom type has a 4x4 matrix. The columns modify the s, p_x, p_y, p_z – type contracted gaussians respectively. The rows are for each orbital on an atom. If there are less than 4 orbitals, the fourth row still must be there for padding.

Finally, we would like to comment on the dynamic calculation of the sparsity list for the *growth* matrices, which is done once every SCF cycle. A cutoff distance determines the sparsity list for the first SCF cycle. For subsequent SCF cycles, the sparsity list of the *growth* matrices is determined by *cut_gro* in the routine *CALC_L2*. **S** multiplies the current Φ for the determination, in *INTERSECT_MAG*. The routine loops over the localization regions and over the shells to determine if a data block has an element larger than *cut_gro*. If *add* is true, then that data block is added to the sparsity list. This is an $O(N^2)$ search as has been all of the searches.

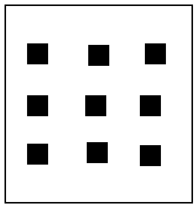
Sparse Grassmann Conjugate Gradient (GCG) Algorithm

The Grassmann conjugate gradient algorithm needs to be scrutinized a bit more once localization is imposed. Once localization of Φ is imposed the hard fought benefit of the insight of the mathematics begins to disappear. We are no longer dealing with the type of vector spaces usually encountered in tensor analysis books. Metrics, inner products, and the gradient are now not well defined anymore. Since we have properly constructed the aforementioned mathematical objects in the full vector space limit, the hope is that the usual linear algebra and tensor analysis formulas can be used with little repercussions. The routine *LIN_NRG_SOLV_SP* calculates the ground-state orbitals with the restriction of localization on the orbitals.

Action of S^{-1}

With localization, the definition of the gradient, Eq. (3.25), changes. Naively, one might calculate the full S^{-1} and truncate the result of Eq. (3.25) in order for **G** to lie in the same space as Φ. The first problem is that this is an $O(N^3)$ process. Regardless of the scaling, this is not a good approach. Including the full S^{-1} adversely affects the new "localized" **G** by adding in curvature in directions, basis functions, of the *primitive* space that are not allowed. This curvature information is then thrown out, but it already has impacted the calculation of the "localized" **G,** which should not be aware of the curvature outside of its allowable directions. This technique has been tested and has shown inferior convergence when compared to our following solution.

Our solution accounts for what the action of $\mathbf{S}^{-1}$ is actually accomplishing within the localized orbital picture. These orbitals are restricted to lie in a certain subspace of the entire vector space of our basis set. An orbital is affected only by the geometry of the vector space in its localized region. Therefore, only a certain part of the overlap matrix, **S**, is associated with an orbital. This part of **S** is inverted in order to obtain the part of gradient for a given orbital. First we retrieve the square submatrix of **S** corresponding to the overlap of the basis functions which are allowed to contribute to an orbital. This corresponds to the dark part of the matrix in Fig. 1. This



matrix is inverted and the result is stored in memory as a packed matrix. The action of $\mathbf{S}^{-1}$ is then just a matrix multiply of a column of

$\mathbf{(I - S\Phi\bar{\Phi}^\dagger)H\Phi}$, (4.1)

corresponding to a single localized orbital, by its corresponding overlap inverse. This is the action of the contravariant metric of the subspace in which the orbital resides. This method scales linearly.

Fig. 1 Schematic diagram showing the subspace of **S** inverted for the calculation of **G**

Line Minimization

Without localization the initial estimate of the step size from Eq (3.29) always gives a decrease in the energy. With localization, the step size sometimes gives an increase in the energy. This occurs more often with small localization values. A version of Brent's algorithm[6] using gradient evaluations has been implemented to ensure a decrease in E_B. For small localization regions, sometimes a decrease in E_B cannot be found in the search direction so the GCG algorithm is restarted. The maximum number of line minimization is specified by *maxfev*, for maximum function evaluation. Sometimes when a decrease in E_B in a direction cannot be found, a decrease can be found in the opposite direction. We tried searching in the opposite direction before restarting the GCG algorithm, but a significant benefit when compared to the computational effort was not found.

Non-Associativity of Matrix Multiplications

For dense matrices, matrix multiplication is associative, i.e. $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$. For sparse matrix multiplication, we no longer have this property. In order to keep the multiplications linear scaling certain matrix element multiplications are neglected. Thus, sparse matrix multiplications are not associative because every multiplication for every element is not done. So when the multiplication of the three matrices occurs, some matrix elements will not be accessed for one order of multiplication as for another order. They may be very close, but will not be exact.

One example is in the calculation of Eq. (3.26) to determine how much old search direction to include. In this equation we take inner products involving current and past gradients. With associativity, we could limit our operation account by taking into account previous calculations. Using the relation of our defined inner product on the Grassmann manifold of

$\langle \mathbf{G}_{\mathrm{I}} \bullet \mathbf{G}_{\mathrm{I}} \rangle = \mathrm{Real}\{\mathrm{Tr}[(\Phi^{\dagger}\mathbf{S}\Phi)^{-1}\mathbf{X}_1^{\dagger}\mathbf{S}\mathbf{X}_2]\},$ (4.2)

we can use this relation, with the right hand side being the standard dot product,

$\langle \mathbf{G}_{\mathrm{I}} \bullet \mathbf{G}_{\mathrm{I}} \rangle = \mathbf{G}_{\mathrm{I}} \bullet \mathrm{dE_B}/\mathrm{d}\Phi.$ (4.3)

With this relation we can use the matrix **SG,** which has been calculated indirectly from the calculation of the differential and is necessary for the eventual result of the gradient. So without associativity, we are required to do the multiplication **SG**. The numerical difference between the two methods of obtaining $\langle \mathbf{G}_{\mathrm{I}} \bullet \mathbf{G}_{\mathrm{I}} \rangle$ is not very pronounced. The choice was made to go with the more expensive latter one as it gave slightly better results and was more aesthetically pleasing.

Problems with Convergence

With our implementation of the GCG algorithm including the effects of localization, the norm of the gradient, Eq. (4.1) no longer converges to zero. The norm of the gradient now converges to a finite value, which with increasing localization approaches zero. We do not know if this effect is a natural consequence of localization in our basis or our lack of fully understanding the geometry of this new type of vector space caused by the localization of the orbitals. As long as the convergence is consistent and to a small enough value, the algorithm is reliable.

As the norm of the gradient doesn't converge to zero, it can no longer be used as the stopping criteria. An obvious choice is to look at initial estimate for the step size. If the step size is going to zero, one is no longer moving; and the calculation can be considered converged. Even if the line minimization is necessary, the final step size is always comparable to the estimate or smaller. It is also beneficial to have the convergence criteria equal in the limit of no localization. To this end, the convergence is measured by

$\mathbf{G} \bullet \mathrm{dE_B}/\mathrm{d}\Phi$ (4.4)

Without localization, this is exactly the norm of the gradient. It also parallels very closely the initial estimate for the step size due to the similarity of the numerator in Eq. (3.29).

Multiplications in the GCG Algorithm

All sparse matrix multiplies implement as much Basic Linear Algebra Subprograms (BLAS) as possible. BLAS allows access to general operation involving matrices and vectors with machine optimized code that runs several times faster than typical source code. The routines have the knowledge of the cache size, bus speed and capacity, and the CPU. In order to use the level 2 and 3 BLAS routines for matrix-vector and matrix-matrix operations respectively, the proper elements need to be organized into small dense chunks.

In several routines a multiplication list is created for computational efficiency. This list stores the positions of the data elements and/or positions of information in the sparsity list to be accessed. The retrieving of data can be as costly as the multiplication operation itself so it is important for this step to be as fast as possible. Using the multiplication list, one does not check which elements are to be multiplied; one already knows and just retrieves them. It can also be important to retrieve them large chunks with the level 1 BLAS routines for computational efficiency. As in Chapter 3, M is the number of basis functions, and N refers to the number of occupied orbitals.

S^{-1} *Projection Operator*

The routine, *proj_S_subspace*, calculates the separate local inverses for each localization region. The loop runs over the localization regions, and *inv_proj* calculates the inverse for a given list of contributing shells for an orbital for that localization region. Within *inv_proj*, we first collect the small overlap of the **Φ-**contributing shells. The code searches though *list_pr_pr* for a match of a shell, *jshell*, in the corresponding neighbor list of the row shell, *ishell*, in *list_pr_occ*. The submatrix *sinv* is inverted and stored in a packed format for use in *Lin_nrg_solv_sp*. In *Lin_nrg_solv_sp*, the BLAS routine *DSPMV* is utilized for the multiplication of each small inverse times the corresponding columns of a localization region of the matrix resulting from Eq (4.1).

S times Φ

For the multiply of a primitive M x M matrix times an orbital M x N matrix, we use a multiplication list to organize the data into dense arrays for use in BLAS routines. The multiplication list allows accessing of data from the array storage in large chunks with level 1 BLAS. The list contains pointers to blocks of data necessary for the shell-blocked rows of **S** times the orbitals of a common localization region. The pointers are found by merging the neighbor lists *list_pr_pr* and *list_pr_occ*. This routine is called three times and is the second most expensive multiplication. Approximately 1/3 of the time is spent for this routine.

The routine, *MULT_PR_PHI_INIT*, calculates the list for the multiplication of **S** times Φ for a given growth list before every SCF cycle. The only loop, using the variable *iloc*, runs over the localization regions, the column of the resultant matrix. Within the loop, *INV_PROJ2* calculates the list for **S** times the orbitals of *iloc*. Within *INV_PROJ2,* the outer loop, *inbr*, runs over the growth-contributing shells, which are the rows of the resultant matrix. The 2nd loop runs over the Φ-contributing shells needed to calculate the block of the resultant *growth* matrix corresponding to the block row *inbr* and block column *iloc*.

Within this inner loop, the data of the common shells of **S** and of the orbitals of *iloc* are incorporated into a list *pr_phi_ind*. This list stores the size and pointers to the blocks of necessary data involved in the multiplication. The pointers access continuous chunks of data of common shells. So whenever an non-common shell is found a new pointer is needed. The data is retrieved in these blocks by use of *DCOPY*. This blocking of the data is quite efficient for data retrieval and memory required by the list.

The steps of multiplication follow the steps of creating the multiplication list. In *LIN_NRG_SOLV_SP* a loop runs over each localization region (*iloc*), the columns of the resultant matrix. Within the loop, a call to the routine *MULT_PR_PHI* calculates the multiplication of **S** times the columns (orbitals) of *iloc*. The outer loop runs over the growth-contributing shells (rows) and the 2nd loop runs over the Φ-contributing shells. The list *pr_phi_ind* is then used in conjunction with *DCOPY* to access the elements for the multiplication of a blocked-row of **S** times the orbitals of *iloc*. The data is stored into temporary arrays and ready for multiplication.

The BLAS routine, *DGEMM,* is then used to carry out the multiplication. So if a shell is in the growth list then the BLAS call is for a matrix of (2*L+1) x (the number of gaussians in common) times a matrix of (the number of gaussians in common) x (number of orbitals in a localization region). This may be made more efficient if all the shells of an atom can be assumed to contribute. The calculation asymptotically scales linearly as the outer loop runs over the number of localization regions and the inner loops run over the number of shells in the growth list, a fixed number.

$\Phi\ ^{\dagger}$times SΦ

The multiplication of a transpose of an orbital N x M matrix by a *growth* M x N matrix to create an M x M *occupied* space matrix is done in a similar fashion to the previous multiplication. A multiplication list is also made at the beginning of each SCF cycle. The routine *MULT_PHITPHI_INIT* calculates the list of pointers to gaussian shells common for both Φ and **S**Φ. The outer loop, *iloc*, runs over the localization regions corresponding to the rows. The next loop, *jloc*, runs over the columns corresponding from one to *iloc*. This creates a list to calculate only the lower triangular part of $\Phi^{\dagger}\mathbf{S}\Phi$.

At this point, we calculate the common shells of the localization region *iloc* and the growth region *jloc*. The data is accessed only one element at a time. A similar scheme to access data in blocks as in the last section could be implemented. Since the overall cost of this routine is not high, the modification is not imperative. The number of common basis functions and common shells in this order are stored in *ptp_num*. The pointers and data size are stored in *ptp*. The list for the other triangular part is calculated at the end if it is deemed necessary to the user to allow $\Phi^{\dagger}\mathbf{S}\Phi$ to be non-symmetric. The logical variable *Symmetric* decides if the multiplication keeps the matrix symmetric. All multiplications currently impose $\Phi^{\dagger}\mathbf{S}\Phi$ to be symmetric.

The routine *MULT_PHI_OCC* carries out the multiplication. The outermost loop, *iloc*, runs over the rows. The 2nd loop runs from 1 to *iloc*. Now through several more loops, the proper data to calculate the data block corresponding to *iloc* and *jloc* are stored in temporary matrices, *phi_t1* and *phi_t2*. The routine DGEMM handles the multiplication of the transpose of *phi_t1* and of *phi_t2*. If N_G is the number of common gaussians, *phi_t1* has dimension N_G x (number of orbitals of localization region *iloc*); and *phi_t2* has dimension N_G x (number of orbitals of localization region *jloc*). If there are no common shells, the elements are set to zero. At this point if *Symmetric* is true the lower triangular part is copied into the upper triangular part. If *Symmetric* is false than the upper triangular part is calculated in exactly the same fashion.

Φ times ($\Phi^{\dagger}$SΦ)

The multiplication of an orbital M x N matrix times an occupied space N x N matrix does lend itself as easily to the use of BLAS. The routine is kept general so that the size of beginning M x N orbital matrix and the resulting M x N matrix can be of the localized or *growth* sparsity pattern. This multiplication currently utilizes no multiplication list. The rows of the M x N matrices were not seen to be blocked enough to take advantage of such a list. Even the accessing in chunks of number of orbitals for a localization region, 3 or 4 for our test systems, was not advantageous.

The routine can handle Φ only in column stored form, and the output can be in column or row format. The input of a *growth* matrix and output of a *growth* matrix is by far the most computationally intensive combination. This type of call is made two times, once for the gradient and once for the line minimization step, and is the most expensive multiplication. Approximately 1/3 to 1/2 of the time is spent for this routine. The routine might be made more efficient if all of shells can be assumed to contribute to a column of these matrices. The advantage could be a beneficial data-blocked accessing scheme and larger matrices in the calls to the BLAS routines.

The outer loop, *ishell*, runs over the number of shells, i.e. the rows of the input and output M x N matrices. A loop, *inbr*, runs over the number, *n_ish*, of orbitals to which *ishell* contributes for the input M x N matrix. The data of the orbitals (columns) to which a shell contributes are obtained from the *list_occ_pr*-like list of the input M x N matrix and stored in *phi_temp*. If L is the angular momentum of *ishell*, *phi_temp* is of dimension (2*L+1) x *n_ish*. The necessary small matrix, *occ_temp*, of the dense $\Phi^\dagger \mathbf{S}\Phi$ is retrieved. If *n_ish2* is the number of orbitals to which *ishell* contributes for the output M x N matrix, *occ_temp* has dimension *n_ish* x *n_ish2*. The multiplication is then carried out with a BLAS routine. Even though the storage of the N x N matrix is dense, the operation count scales linearly. Memory access could possibly hurt the scaling though.

Charge Density and Self-Consistent Matrix Elements

The charge density, $\mathrm{n}(\mathbf{r})$, is required to evaluate the self-consistent potential, comprising the exchange-correlation, $\mathrm{V_{xc}}$, and electronic Coulomb, $\mathrm{V_c}$, potentials. The calculation of the exchange-correlation potential requires $\mathrm{n}(\mathbf{r})$ to be evaluated on a real space mesh. Since $\mathrm{n}(\mathbf{r})$ is already required to be on a mesh, it makes sense to evaluate $\mathrm{V_c}$ by use of Poisson's equation

$$\nabla^2 \mathrm{V_c} = -4\pi \mathrm{n}(\mathbf{r}), \tag{17}$$

which utilizes $\mathrm{n}(\mathbf{r})$ on this mesh. An alternative method is using the Fast multipole method[9]. Evaluating $\mathrm{V_c}$ by Poisson's equation seems preferable, as almost all of the work is already required to evaluate $\mathrm{V_{xc}}$.

Eq. (17) is typically solved in a reciprocal space representation, plane waves. If $\mathrm{V_c}$ and $\mathrm{n}(\mathbf{r})$ are expressed in a Fourier series, Eq. (17) is transformed into an algebraic equation, which is very easily solved. If $\mathrm{n}(\mathbf{r})$ is expressed on a uniform real space mesh, the transformation to Fourier space is very efficient on serial machines by the use of Fourier transforms (FFTs). The solution of Poisson's equation requires a charge density that averages to zero in order to satisfy the periodic boundary conditions inherent in the use of Fourier series and in an infinite solid. Subtracting off charge equal to the number of electrons usually does the averaging to zero.

This is usually done in plane wave calculation by a uniform charge density, i.e. the constant term in the Fourier expansion is taken to be zero. Feibelman[4,8] introduced a better solution for use in LCAO. This physically motivated method subtracts off an atomic reference charge density, $\mathrm{n_{at}}(\mathbf{r})$, from $\mathrm{n}(\mathbf{r})$ to give $\delta\mathrm{n}(\mathbf{r})$, which averages to zero in the unit cell. With this method $\mathrm{V_c}$, a function of $\mathrm{n}(\mathbf{r})$, is split up into two parts.

$\mathrm{V_c}(\mathrm{n}(\mathbf{r})) = \mathrm{V_c}(\mathrm{n_{at}}(\mathbf{r})) + \mathrm{V_c}(\delta\mathrm{n}(\mathbf{r}))$ (4.6)

$\mathrm{V_c}(\mathrm{n_{at}}(\mathbf{r}))$ is solved analytically once for a given set of atomic positions, and $\mathrm{V_c}(\delta\mathrm{n}(\mathbf{r}))$ contributes to the self-consistent potential. In this fashion, Eq. (17) is transformed into

$$\nabla^2 \mathrm{V_c} = -4\pi \delta\mathrm{n}(\mathbf{r}), \tag{18}$$

The resulting $\delta\mathrm{n}(\mathbf{r})$ is slower varying and thus a coarser mesh can be used.

The density matrix or $\Phi(\mathbf{r})$ is needed for the calculation of $\mathrm{n}(\mathbf{r})$. Evaluating $\mathrm{n}(\mathbf{r})$ with $\Phi(\mathbf{r})$ along with their respective biorthogonal complement scales linearly only for truncated solutions. If eigenfunctions are the solutions, this scales as $\mathrm{O(N^2)}$. If the density matrix is used to calculate $\mathrm{n}(\mathbf{r})$, it scales as O(N). The pre-factor for the orbital method is smaller though because the occupied space is smaller than the primitive space.

For the linear scaling method, the density matrix method was slightly faster and was numerically more stable at achieving charge conservation on the uniform grid. As only large systems should be looked at with the linear scaling, the larger pre-factor was not an issue since the $\mathrm{O(N^2)}$ scaling should already dominate for the orbital method. As most of the machinery was already in place, the main effort for the calculation of $\mathrm{n}(\mathbf{r})$ was to efficiently access the proper elements of the density matrix in the sparse format. Similarly, the routine to integrate the self-consistent potential matrix mostly involve routines to access sparse data properly.

Calculating the Density Matrix

The routine *calc_dmat_sp* handles the sparse calculation of the density matrix with the formula
$\mathbf{P} = \Phi(\Phi^\dagger \mathbf{S}\Phi)^{-1}\Phi^\dagger$. (4.8) The resulting density matrix has the same sparsity pattern as the other primitive matrices.

The code first does the multiplication of $\Phi(\Phi^\dagger \mathbf{S}\Phi)^{-1}$, which utilizes *MULT_PHI_OCC*. The resulting matrix then multiplies $\Phi^\dagger$. The routine *MULT_PHI_PHIT* handles this. Currently it is not a very optimized routine. It only occurs once per SCF cycle and therefore is not very expensive.

In *MULT_PHI_PHIT*, the outer loop, *ishell*, runs over the shell-blocked rows of $\Phi(\Phi^\dagger \mathbf{S}\Phi)^{-1}$ and effectively $\mathbf{P}$. The next loop runs over the occupied orbitals, *jocc*, to which *ishell* contributes in the growth list. These orbitals are the columns of $\Phi(\Phi^\dagger \mathbf{S}\Phi)^{-1}$. This block of data corresponding to the row *ishell* and column *jocc* then multiplies every necessary block of $\Phi^\dagger$. The necessary multiplications are determined in *MERGE* given the sparsity structure of **P.** *MERGE* calculates the intersection of the shell neighbors of *jocc*, using *list_pr_occ*, in the localization region and the shell neighbors *ishell* of the primitive matrix, using *list_pr_pr*. So the (*ishell*, *jocc*) block of $\Phi(\Phi^\dagger \mathbf{S}\Phi)^{-1}$multiplies a column of the row *jocc* of $\Phi^\dagger$ if that column exists in $\mathbf{P}$.

For the calculation of the density matrix, the multiplications need to be consistent with the multiplication in the GCG algorithm. We tried calculating exactly all possible multiplications, i.e. no use of the *growth* cutoff, to calculate every matrix element of $\mathbf{P}$ that is accessed. The total electron value was significantly less accurate than with the consistent method described above.

Charge Density

The first step after the density matrix has been found is to calculate the charge density. The routine *GRIDRHO_LIN*, derived from the routine *GRIDRHO* that uses a non-sparse density matrix or eigenfunctions, calculates the charge density on a uniform grid. The pre-existing code of *GRIDRHO* and associated routines has the uniform grid separated into boxes typically of 5 points per side. The outer loop, *ibox*, runs over the boxes. Currently a $O(N^2)$ look-up table specifies which gaussian functions have a significant value within this box determined by *convgr*. A fixed number of gaussians will contribute to a box as the system size increases.

The routine *BOXWFN* returns the function values of the basis functions, of number *nobox*, at the grid points, of number *nrbox*, in the box. The result is a matrix $\mathbf{WF}(\mathbf{r})$ of size *nrbox* x *nobox*. $\mathbf{WF}(\mathbf{r})$ is the projection of the basis set onto the grid points within the box. If we have a box has 125 points, $\mathbf{WF}(\mathbf{r})$ would be a matrix of 100 x *nobox*. *BOXWFN* also returns a list, *list_sh_inbox*, of the *n_sh_inbox* gaussian shells, which contribute to the box.

At this point, the routine gathers the density matrix elements corresponding to the *nobox* basis functions. This can be seen as the local projection of the density matrix onto the physical space of *ibox*. In some ways, the method is similar with the density-matrix formulation[7] of the divide-and-conquer scheme.The next loop, *ish*, runs over the *n_sh_inbox* gaussian shells. This effectively is running over the rows for the projected density matrix. The neighbor list, in *list_pr_pr*, for *ish* is merged with *list_sh_inbox* in order to find common shells. The next loop also runs over *n_sh_inbox*; and if common shells exist, the data is stored in the small dense temporary *dmat_small*. If the shell is not common, the data block in *dmat_small* is set to zero. Now dense linear algebra takes over and the charge density is calculated with BLAS routines according to the formula

$$(\mathbf{WF}(\mathbf{r})\ \mathbf{P}_{\text{proj}})\mathbf{WF}(\mathbf{r})^{\dagger} \quad (4.9)$$

Self-Consistent Potential

The potential due to the inaccuracies of the grid is fixed with *VSLOFIX_SP*. Since the potential is not calculated analytically, the incompleteness of the grid is partly corrected. Only the one-center[4] correction is implemented in *VSLOFIX_SP*. The correction is calculated by multiplying the potential at the nearest grid point to the atom by value of *s1atom*, see Sec. 4.1.1, for the two gaussians making up the self-consistent matrix element. The outer loop, *iatm*, runs over the atoms. The block of data for the self-consistent potential for gaussians on the same atom is then modified as stated above. The routine *locate* is used again on the loop over the rows of the same site data block to find out where the data is located.

The next step is to integrate the self-consistent potential on the uniform grid in order to calculate the self-consistent matrix elements. The routine *VSLOMAT_SP* is structured similarly to *GRIDRHO_LIN*. The outer loop runs over the boxes of the uniform grid, and *BOXWFN* gets $\mathbf{WF}(\mathbf{r})$, which is necessary for the integration. Each row of $\mathbf{WF}(\mathbf{r})$ is scaled with the V_{sc} at that point to form the matrix **VWORK** of dimension *nrbox* by *nobox*. This box's contribution to the self-consistent matrix elements is then calculated in **MAT_WORK** by

$$\mathbf{WF}(\mathbf{r})^{\dagger}\ \mathbf{VWORK} = \mathbf{MAT\_WORK} \quad (4.10)$$

implemented with *SGEMM*.

The values in **MAT_WORK** are then summed up in a temporary storage *v_sp_work* of sparse *primitive* size. The proper data storage is found by looping over the shells in the box, *ishl*, and using *merge* to find the intersection *list_sh_inbox* and the neighbor list of *ishl*. The next loop only needs to run over the number of common shells as oppose to all of the shells in the box as in *GRIDRHO_LIN*. This is because only the common shells will give a contribution to the elements allowed to be nonzero. Once a common shell is found the block of data for those orbitals is added to the existing values from other boxes for that data block. After the loop over the boxes is done, the elements of *v_sp_work* are multiplied by an integration factor and added to the values existing from *VSLOFIX_SP* to form V_{sc}.

Miscellaneous

In *VSLOFIX_SP*, we calculated a correction to V_{sc}, and in *ATDEFCT_LIN* we calculate a correction *defatom* used to correct the energies evaluated on the grid. Only the one-center[4] correction is implemented in *ATDEFCT_LIN*. The array *defatom*, one element for each atom, is calculated. It is the sum of the multiplication of the appropriate elements of *s1atom*, see Sec. 4.1.1, with the corresponding matrix element of **P**. The outer loops run over the atoms and then over the shells for each atom in order to access each basis function *iorbat*. *Locate* is used to access the proper elements of the density matrix that multiply the appropriate elements of *s1atom*. Additional loops run over each basis function on the atom, *jorbat*. We now multiply the value corresponding (*iorbat, jorbat)* values of *s1atom* and **P**.

The self-consistent potential due to the delta charge density between the final result and the initial atomic reference density is added to the Hamiltonian of the atomic reference for each new geometry. This is not a straightforward addition as the old primitive matrix of the self-consistent potential may have a different sparsity pattern. The conversion is done in the routine *convert_old_pr_new*. On the 1$^{\text{st}}$ geometry the routine just saves the old *list_pr_pr* in the proper place so the logical *convert* is false. The outer loop runs over the gaussian shells of the entire system. The intersection of the old sparse list, *list_pr_pr_old*, and the new sparse list *list_pr_pr* for each shell is found with *merge*. The next loop then runs over the common shells and transfers the data into the new position in the matrix array.

Forces

The calculation of forces for an LCAO basis set is more complicated than with plane waves. With plane waves the basis set does not change with a movement of the atoms, and the Hellman-Feynman formula[10] is used. With LCAO, the basis set does change with atomic position. The gradient of the calculated total energy must now include contributions from the nuclear-displacement-related changes of the basis set. Pulay[11] has derived the generalization of the Hellman-Feynman formula for the case that basis orbitals do not change shape as they move with a nucleus. The adaptation of the forces from the pre-existing dense routines is quite simple. Once the sparse density matrix is obtained one just needs to access the information properly.

The force from the potential on the grid is calculated in *VSLOFRC_SP* with additions to the dense counterpart similarly as in *VSLOMAT_SP*. The projected density matrix is obtained as in *GRIDRHO_LIN* and the operation

$$\mathbf{WF}(\mathbf{r})\ \mathbf{P}_{\mathrm{sm}} = \mathbf{WKBI} \quad (4.11)$$

gives the matrix **WKBI** which is then used with gradient information of the gaussian orbitals to calculate stress and force terms.

FRC2CTR_SP calculates the force and stress from the overlap and kinetic energy terms. The routine loops through the atoms(*iatm*), its shells(*ishell*), and then through another list of atoms(*jatm*) to *iatm*, neighboring unit cells, and its shells(*jshell*). If *jshell* and *ishell* have a significant interaction, the proper elements of the density matrix and the energy matrix defined by

$$\mathbf{EMAT} = \Phi(\Phi^{\dagger}\mathbf{S}\Phi)^{-1}\Phi^{\dagger}\mathbf{H}\Phi\ (\Phi^{\dagger}\mathbf{S}\Phi)^{-1}\Phi^{\dagger} \quad (4.12)$$

are accessed. *Locate* is again used to find the pointer to the proper data. Only the lower triangular part of these symmetric matrices is accessed. Since only the Γ-point, hence only real numbers, is used two multiplies the off-diagonal elements of the density matrix. The contribution of the density matrix for *NLOCFRC_SP* and *VLOCFRC* is accessed in exactly this same method.

P. Schultz and P. J. Feibelman, to be published.

N. Troullier and J. L. Martins, Phys. Rev. B **43**, 1993 (1991).

D. R. Hamann, Phys. Rev. B **40**, 2980 (1989).

P.J. Feibelman, J. Chem. Phys. **81**, 5864 (1984)

G. B. Bachelet, D. R. Hamann, and M. Schluter, Phys. Rev. B **26**, 4199 (1982).

J. J. More and D. J. Thuente (1990), "On line search algorithms with guaranteed sufficient decrease", Mathematics and Computer Science Division Preprint MCS-P153-0590, Argonne NAtional Labaoratory (Argonne, IL)

M. C. Strain, G. E. Scuseria, and M. J. Frisch, Science **271**, 51 (1996)

P. J. Feibelman, Phys. Rev. B **33**, 719 (1986)

W. Yang and T. S. Lee, J. Chem. Phys. **103**, 5674 (1995)

H. Hellman, *Einfuhrung in Die Quantenchemie* (Deuticke, Leipzig, 1937); R. P. Feynman, Phys. Rev. **56**, 340 (1939)

P. Pulay, in *Modern Theoretical Chemistry,* edited by H. F. Schaefer (Plenum, New York, 1977), Vol. 4, pp. 153-185; Mol. Phys. **17**, 197(1985)

APPLICATION

First-principles calculations have impacted and will continue to impact the role of the theoretical scientist. DFT has been proven to be a reliable tool for studying many simple idealized materials of technological interest. The materials are simple in the sense that the primitive cell typically involve few (10∼100) atoms. The calculations are idealized in the sense that they usually assume a perfect crystal with no defects, although first principles DFT calculations are becoming more commonplace for defects. In order to handle more realistic systems instead of the usual idealized systems, more complex and larger systems are needed. Calculations have been limited in the number of atoms in computationally demanding quantum mechanical methods, such as DFT. This has been due mostly to a large *primitive* space, e.g. plane waves usually chosen by physicists. As mentioned previously in this thesis, the calculating of the Hamiltonian matrix elements and its diagonalization also has limited the system size.

Parallel computers and code that is explicitly a plane wave basis or implicitly very similar, i.e. a real space uniform mesh, have made great strides in handling large systems. The codes implement a real space mesh, resulting in sparse matrices, by use of an FFT[1] or finite-difference.[2,3] The availability of these machines is limited though. An alternative to plane waves is the use of a linear combination of atomic orbital (LCAO) representation[4] with a comparatively small number of basis functions. This representation also results in sparse matrices. Physicists have typically shied away from this representation due to the non-orthogonality and non-completeness of the typical basis sets.

Methods[5−7] for the calculation of the Hamiltonian matrix elements in a LCAO basis have been shown to asymptotically scale linearly with system size. In many ways, these methods are more straightforward than the methods to achieve linear scaling for the calculation of the *occupied* space. Linear Scaling algorithms have promise to be used as an alternative to diagonalization in DFT and Hartree-Fock. Several papers have demonstrated linear scaling behavior on test systems, predominantly carbon,[8] silicon[9,10] and water clusters.[11,12] One code[13] has been used on many important systems with mostly an emphasis on qualitative results. There exists a need for more testing of the accuracy in real problems to find out at what level the algorithms can quantitatively contribute.

Issues such as desired accuracy and computational effort need to be addressed in order to ascertain the usefulness of the codes as well as some overall familiarity of operational settings. Overall for linear scaling methods to become an integral part of the arsenal of the theoretical physicist, one needs to feel as comfortable with their reliability as with plane waves. We wanted to test the ability of our occupied subspace optimization code to relax atoms in complex systems and to resolve the energies of different configurations of the system.

The linear scaling techniques that employ an atomic orbital basis, solely used in chemistry methods, first need to establish the accuracy of the basis set for a particular problem. The accuracy could be tested by comparison to converged plane wave results. This is not usually done in the chemistry community where the DZP basis is considered as the standard for DFT and Hartree-Fock. The other approximation that needs to be addressed is the restriction of the nonorthogonal occupied orbitals to have a non-zero contribution from only selected basis functions, gaussians in this paper.

The localization is at the heart of this Chapter. This restriction to a smaller vector space is an approximation that one needs to gain experience in using. One of the hardest steps in using an algorithm that restricts the orbital solutions is how to know *a priori* an acceptable localization region for the orbitals. The acceptability is dependent on the necessary accuracy of the current problem. The accuracy could be a given error in relaxed atomic positions or relative energies. Once the desired accuracy is known, the result is converged or compared against diagonalization to determine the localization region. After one has gained experience in using localization, only then can one use previous data to justify the use of a localization region for a given problem.

Relative Energy Calculations in Silicon Carbide

Initially, we test the accuracy of the GCG algorithm to resolve relative energies between different systems with different localization regions. Silicon carbide, a wide gap semiconductor that can be operated at high temperature and high pressure, was chosen for its technological importance[14] and for its multiple crystal phases. The latter provides a stringent test on the accuracy of the total energy calculations. The relative energy differences between the 3C (cubic) and the 2H and 4H (hcp) phases[15] are used for this purpose.

Dense Calculations

First, we check converged calculations using diagonalization within the local density approximation (LDA) against experimental[16] results and recent plane wave[15] (LDA) and LMTO[17] (GGA) calculations. For the two-atom 3C phase, we use up to a 12x12x12 Monkorst-Pack[18] mesh. We include the Γ point, $\mathbf{k} = 0$, for the $\mathbf{k}$-point sampling, in order to obtain the total energy for a Γ point sampling of the Brillouin zone for the large unit cells. As the unit cell increases, the $\mathbf{k}$-points of the primitive cell fold into each other and all eventually fold into the Γ point. For example, a calculation using a 5x5x5 mesh including the Γ point of the two-atom primitive cell of 3C SiC gives the same energy/atom as a 250-atom Γ point calculation.

For the 4-atom 2H and 8-atom 4H, we used up to a 9x14x14 and a 3x10x10 mesh, respectively. A grid spacing of 0.356 Bohr was used for the calculation of the potential from the residual charge density (see Sec. 4.4) via Poisson's equation. We relaxed the internal positions for the hcp phases. Table 1A compares our structural results with experiment and our energetic values with other theoretical results. All structural values are within 1% of experiment. Our relative energies give the same ordering of crystal phases as other theoretical results. Table 1B gives the energy/atom for the different phases at different system sizes obtained from $\mathbf{k}$-point sampling as mentioned above.

	a (Bohr)	C/a	energy (Ry)
Cubic	4.0950		9.7035238
	4.1196[a]		
2H	5.7807	1.642	9.7032725
	5.8127[a]	1.641[a]	
4H	5.785	3.275	9.7035581
	5.807[a]	3.271[a]	

Table 1A. Structural and energetic results with diagonalization respectively compared to experimental and theoretical values.
[a]reference 16 (Expt.)
[b]reference 17 (LMTO)
[c]reference 15 (PW)
[d]Present work (LCAO)

		Ry/atom	
# of atoms	Cubic	2H	4H
64	9.69864	9.70785	9.69860
128	9.69876	9.70090	9.70086
200		9.70268	9.70266
250	9.70219		
288		9.70316	9.70314
432	9.70312		
686	9.70339		
1024	9.70348	9.70326	
1176			9.70349
1458	9.70352		

Table 1B. Γ point energies derived from **k**-point sampling of the primitive unit cells.

Sparse Calculations for 64-atom Unit Cell

Now imposing varying localization, we investigate the accuracy, convergence, and scaling of computational effort with system size for Γ point calculations of 64 to 1458 atoms. The occupied orbitals are initially formed from a linear combination of the single zeta gaussians. The determination of the localization region for required accuracy, e.g. the energy differences to be resolved, is logically the first step in these calculations. Generally, if the localization region is increased, higher accuracy is obtained. The localization also affects directly the convergence and computational effort. The desired accuracy also determines acceptable convergence criteria and the growth of **S**Φ, which also affects computational effort.

We start with the 64-atom system. Energies are calculated with the relaxed geometry from the converged **k**-point calculations. **S**Φ had full growth. Table 2 gives the energies for the different phases for a given localization including the diagonalization result for reference and the corresponding energy difference with the 2H phase. The localization is given as the cutoff radius for the single zeta, double zeta, and polarization shells. A "b" corresponds to an orbital whose localization region is centered between the two atoms and "a" is for the orbital centered on the carbon. For example, a 5,4,5-b setting is centered at the center of the bond with the radii of the single zeta and polarization functions being longer then the radius for the double zeta. Four atom-centered orbitals on carbon were used for the larger localization region as the code takes advantage of shared sparsity patterns among orbitals to use less memory and run faster. At a 7 bohr radius cutoff, the code was more accurate and efficient for atom centered orbitals.

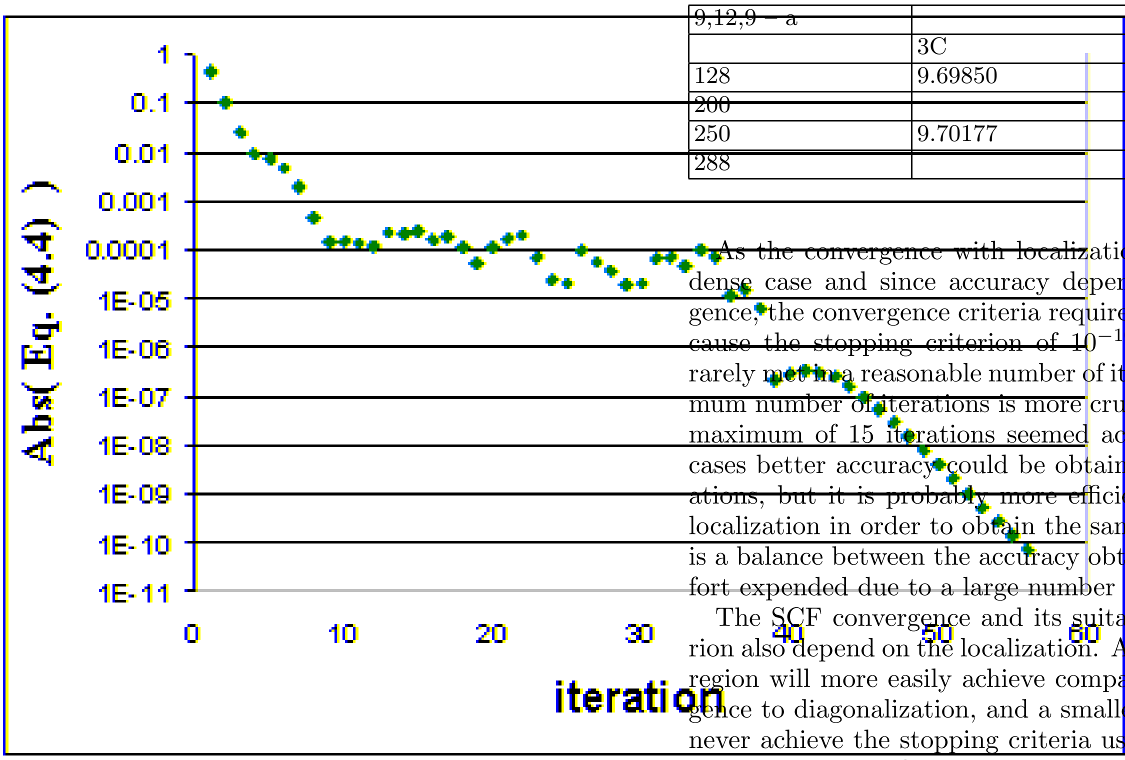
			Energy/atom
	3C	Δ E(2H – 3C)	2H
Diagonalization	9.69864	0.00920	9.70785
5,4,5 – b	9.69065	0.01083	9.70149
4,5,5 – b	9.67891	0.01192	9.69084
5,5,5 – b	9.69208	0.01171	9.70378
7,5,7 – b	9.69343	0.01200	9.70543
5,7,7 – b	9.69364	0.01133	9.70527
7,7,7 – b	9.69418	0.01165	9.70583
7,7,7 – a	9.69706	0.00983	9.70690
9,7,9 – a	9.69797	0.00932	9.70729
7,9,9 – a	9.69795	0.00926	9.70721
9,9,9 – a	9.69824	0.00917	9.70741
9,11,9 – a	9.69838	0.00918	9.70756
11,9,9 – a	9.69831	0.00921	9.70752
9,12,9 – a	9.69860	0.00905	9.70765
12,9,9 – a	9.69843	0.00911	9.70756
11,11,11 – a	9.69843	0.00921	9.70765
12,12,12 – a	9.69863	0.00914	9.70777

Table 2. Energy in Ry/atom for varying localization regions for 64-atom SiC

In order to have an equitable comparison with diagonalization and varying localization, the energy values in Table 2 are given after 4 SCF iterations and a maximum of 15 GCG iterations with a stopping criterion at 10^{-10} for Eq. (4.4). After 4 SCF iterations, the diagonalization calculations were converged to within 0.1mRy/atom. In order to take advantage of the smallest localization region for a desired accuracy, we looked at the effect that different gaussian shells had on the energy. Near the center of the orbital, the single zeta functions have a larger impact on the accuracy of the total and relative energies than the more diffuse double zeta functions. The single zeta functions give more variational freedom where the orbital has the highest probability and the strongest interactions, i.e. near the center of the bond.

With the localization radius between 5-9 bohr, having the single zeta with a larger radius than the double zeta gave lower energies energy for the 3C and 2H structures, but higher for 4H. In this range, a larger radius for double zeta gave better accuracy for all energy differences. Above 9 bohr, the double zeta gave lower energies for all structures, but the accuracy was mixed. Accuracy of less than 0.2 mRy/atom, which is on the order of the energy differences for the converged results, was obtained when the radius was 9 bohr for each shell. The 9 bohr radius corresponds to the extent of the orbitals of silicon determined by the cutoffs already in Seqquest, which give an accuracy of 1 μRy/atom. A radius of 12 bohr gave an accuracy of less than 0.1 mRy/atom, which also was obtained with a 9 bohr radius but only after 10 SCF iterations. The SCF and GCG convergence with a larger radius for the double zeta did generally better.

With Eq. (26) as the measure of convergence for the GCG algorithm, we plot, in Fig. 1, the absolute value of Eq. (4.4) for the 5,4,5-b setting of the 64-atom system for the first SCF iteration. Fig. 1 shows three distinct stages for the convergence of the GCG algorithm: (1) starts off well with convergence typical of a conjugate gradient algorithm, (2) hits a region of flat convergence,



and (3) then ends well. This behavior is exhibited in all of the localization regions. The duration of each stage and the corresponding transition points vary with localization radius. The good convergence at the end is generally accompanied by Eq. (4.4) being negative at some steps. In Fig. 1, Eq. (4.4) becomes and stays negative at iteration number 39. This behavior occurs for subsequent cycles and is explainable by the following.

Fig. 1 **G** •$dE_B/d\Phi$ vs. iteration # for the 5,4,5 – b setting of the 64 atom unit cell of SiC.

At the beginning, the largest gradient values are for the gaussians near the center of the localized orbital, which are within the localization region. This corresponds to the good initial convergence that results from the orbitals being allowed to change in the direction (gaussians) of steepest descent. As the gradient gets small, the size of the gradient for the gaussians within the localization region becomes comparable to the size for gaussians outside of the localization region. The error in the gradient due to localization is now of the same order of magnitude as its length. Thus, the GCG convergence stalls, as orbitals are not allowed to move in directions that would give a significant decrease in the energy. Localization has effectively cut off the bottom of the bowl that makes the minimum. The orbitals wander around the edge of this cut, as the direction that would take Φ to the bottom is no longer allowed. At the end, the sharp drop is most likely due to the two vectors **G** and $dE_B/d\Phi$ becoming perpendicular. **Z** and $dE_B/d\Phi$ also become perpendicular at this stage.

9,12,9 – a		Energy/atom
	3C	2H
128	9.69850	9.70053
200		9.70222
250	9.70177	
288		9.70269

As the convergence with localization differs from the dense case and since accuracy depends on the convergence, the convergence criteria require investigation. Because the stopping criterion of 10^{-10} for Eq. (4.4) is rarely met in a reasonable number of iterations, the maximum number of iterations is more crucial. Generally the maximum of 15 iterations seemed acceptable. In some cases better accuracy could be obtained with more iterations, but it is probably more efficient to use a larger localization in order to obtain the same accuracy. There is a balance between the accuracy obtainable and the effort expended due to a large number of GCG iterations.

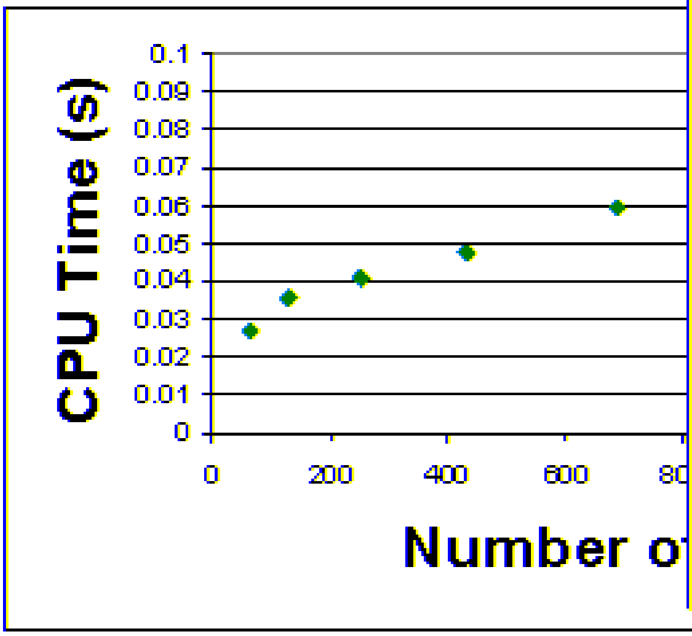
The SCF convergence and its suitable stopping criterion also depend on the localization. A larger localization region will more easily achieve comparable SCF convergence to diagonalization, and a smaller localization may never achieve the stopping criteria used for diagonalization calculations. One solution to this problem is to know the accuracy obtainable for a given localization. Since less accuracy is expected from smaller localization regions, the SCF convergence can be stopped for a larger energy difference than for diagonalization and larger localization regions. For example, one can consider the 5,4,5–b converged once the energy change is less than 1 mRy/atom as higher accuracy is not expected. This was achieved in the 4 SCF iterations. The 9,12,9-a converged to within 0.1mRy/atom after 4 SCF iterations. If the SCF procedure takes too long for the desired convergence, then one needs a bigger localization.

Sparse Calculation of Larger Systems

We now turn our attention to *cut_gro* (see Sec. 4.1.2). For the 250-atom cubic unit cell using the 5,4,5-b setting, a significant drop in total energy was seen above a value of $5x10^{-3}$ for *cut_gro*. For the accuracy expected here, the growth cutoffs for the first cycle were not crucial. For the 9,12,9-a localization we looked at several system sizes. We wanted to determine the growth value necessary to keep the same accuracy and the proper energetic ordering of the crystal phases. For as large a system as we could do, 10^{-3} was sufficient. The growth for the first cycle was 12,15,12 using the naming convention for the localization region. The results are presented in Table 3.

Table 3. setting of 9,12,9 – a with growth of 12,15,12 for 1st SCF iteration and 10^{-3} for subsequent iterations.

5,4,5 – a		Energy/atom
	3C	2H
128	9.69078	9.69089
200		9.69162
250	9.69296	
288		9.69199
432	9.69270	
686	9.69244	
1024	9.69349	9.69159
1176		
1458	9.69277	



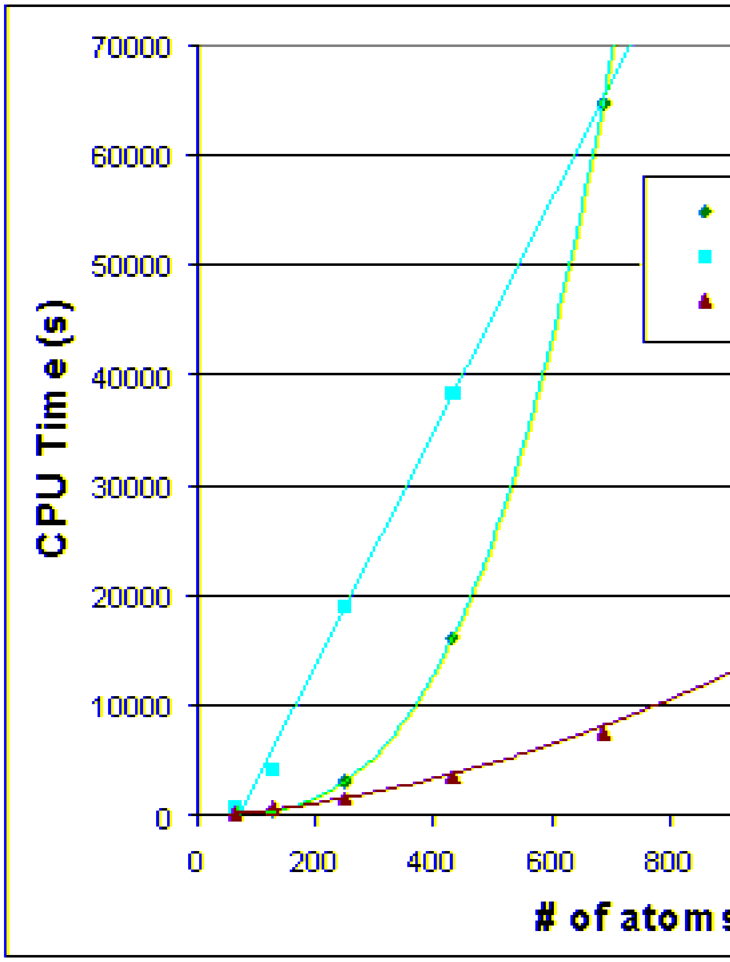


Fig 2. CPU time for one GCG step per orbital for the 5,4,5-b setting Vs. # of atoms.

The accuracy for the 5,4,5-b localization is presented in Table 4. This localization did not obtain the proper ordering at every system size but did succeed with the proper ordering for the largest systems done.

Table 4. setting of 5,4,5 – a with growth of 10,10,10 for 1st SCF cycle and $5x10^{-3}$ for subsequent cycles

As a test of the scaling for the 5,4,5-b setting, we present in Fig. 2, the timing for 1 GCG step per orbital vs. number of atoms. Perfect linear scaling would give a horizontal line. The N^3 scaling of the inversion of the occupied space, $\Phi^{\dagger}\mathbf{S}\Phi$**,** begins to be substantial around 1000 atoms. The timing for the 1458-atom system is smaller than expected. The impact of the N^3 parts should cause increasing deviation from a horizontal line. We attribute this discrepancy to fluctuations in timing due to running on a non-dedicated machine. For the 1458 atom system, 11.6% of the elements of $\Phi^{\dagger}\mathbf{S}\Phi$ are non-zero so only at this point might an asymptotically linear scaling approach be more efficient.

In Fig 3, we show a timing plot of diagonalization and optimization with two localization regions (5,4,5-b and 9,12,9-a) vs. time (s). This displays the large difference in crossover that occurs as the localization is increased. Crossover can be obtained early, but at a cost of accuracy. These numbers are very important when deciding if a linear scaling algorithm is suitable for a certain problem. The Lapack routine *DSYGVX*, which calculates a given number of lowest eigenvalues and eigenvectors, from the optimized library *DXML* was used. All of the times are for a non-dedicated serial 440 Mhz DEC workstation. The user time from the function DTIME was used.

Fig. 3 Timing for 4 SCF cycles for diagonalization, 9,12,9-a, and 5,4,5-b.

Forces, Relaxation, and Formation Energies of Yttria-Stabilized Zirconia

Forces are an integral part of electronic structure calculations. Whereas absolute energies for a given geometry cannot be compared between different methods, the value of the forces can. Accurate forces are necessary to calculate the ground-state structural properties of a system. For first-principles molecular dynamics, which I don't present here, validity of a reaction path or of a statistical ensemble depends on the accuracy of the forces.

Forces are also necessary when coupled with minimization methods to relax the atomic positions to get a local minimum of a system. The ground-state geometry is important for comparing relative energies. To have an accurate comparison, the geometries and possibly the unit cell also must be optimized. In a few cases, we optimize the unit cell by minimizing the stress on it. The calculation of formation energies also requires the ground-state atomic positions and unit cell shape.

Physical System

We are primarily interested in zirconia (ZrO_2) in the fluorite (FCC) phase.[19,20] Zirconia is only stable in the fluorite phase at high temperatures or at room temperature with approximately a 4% -30 % mol[21] doping of yttria (Y_2O_3). Structures with this doping concentration are termed yttria-stabilized zirconia (YSZ). YSZ is used as an oxygen detector and in fuel cells due to its high oxygen diffusion. The oxygen diffusion is at the highest around 8 % mol[21] of Yttria.

The fluorite phase of Zirconia has three atoms in the unit cell. Zr lies in the center of a cube with the oxygen atoms at the corners making it 8-fold coordinated. The oxygen is four-fold coordinated with all Zr neighbors. Yttria has the bixbyite[22] structure as its lowest energy crystal phase. It is a 40-atom BCC unit cell. It can be viewed as a distorted fluorite lattice with the Y being 6-fold coordinated. Each Y lies in the middle of a cube with the oxygen in close proximity to the corners of a cube, hence a distorted lattice.

Essentially Yttria (Y_2O_3) in the bixbyite structure[22] is mixed into Zirconia (ZrO_2) in the fluorite phase. A single defect is defined as 2Zr $\rightarrow$ 2Y and an O $\rightarrow$ vacancy. Around the defect, the atoms will relax with the most movement around the vacancy. Relaxation plays a major role in the relative energies; therefore, it is a good system to test the forces. For example, the energy for 2c configuration (see Sec.5.1.3.2) changes by 0.833 Ry from the ideal fluorite positions to the relaxed geometry. The many configurations allow for testing of relative and formation energies, and the movement of the atoms in response to the defect is a good test for the accuracy of the relaxation.

	VASP	Seqquest
E_{Tetrag}- $E_{fluorite}$	-36	-27
$E_{Monoclinic}$- $E_{fluorite}$	-81	-6
$E_{bixbyite}$ - $E_{\alpha-alumina}$	119	137

Basis Set Accuracy

a Zirconia and Ytrria In order to test the accuracy of the basis, we compared against plane-wave results, VASP,[23−26] using ultra-soft pseudopotentials.[27,28] and the LDA approximation. An energy cutoff of 495eV was used. For VASP and SEQQUEST, the **k**-point meshes are 4x4x4 for all zirconia phases and 2x2x2 for the bixbyite phase and 4x4x4 for the α-alumina phase. The lattice vectors all lie within 1% of experiment for zirconia and yttria. Table 5 gives the relative energies of different phases of Zirconia and Yttria. For Zirconia, we compare the monoclinic and tetragonal phases to the fluorite structure. For yttria, we compare the α-alumina phase to the bixbyite structure.

Table 5 Relative energy for ZrO_2(compared to the fluorite phase) and Y_2O_3 (compared to the bixbyite phase) in units of meV/cation.

The biggest discrepancy is the energy of the monoclinic phase of Zirconia. This could be attributable to the 7-fold coordination of the cations in the monoclinic phase whereas tetragonal and fluorite phases have 8-fold coordination. With an added polarization function on oxygen, we do obtain the monoclinic phase slightly lower than the tetragonal phase. Since YSZ stays in the fluorite structure or undergoes only tetragonal distortions, we believe the energy discrepancy for the monoclinic phase is not a major problem.

b Formation Energy of Composite Structures As an additional check, we also looked at the formation energy of some structures with 40% or higher % mol doping. These doping concentrations are higher than that of YSZ, but we used these structures, as plane wave results were readily available. The accuracy of SEQQUEST for these very similar configurations should translate over to the lower concentrations of yttria in YSZ. In the course of calculations looking at different doping concentrations of Y_2O_3, we will want to compare the energetics of systems with a different number of atoms. This prevents a direct comparison of the total energies, E(YSZ), for determining which configuration is more energetically favorable. The following formula allows the calculation of the formation energy (enthalpy) to compare energetics of systems with

VASP Seqquest lattice vectors
$Zr_3Y_4O_{12}$ 108 101 w/ V a = 9.82 4.93 4.89
95 w/o V b = -4.85 9.80 4.82
c = -4.92 -5.10 9.98
$Zr_2Y_2O_7$–V2a 471 491 a = -4.84 0.00 4.84
497 b = 4.84 –4.84 0.00
c = 10.53 15.38 15.38
$Zr_2Y_2O_7$–W2a 51 50 a = 4.72 –4.72 0.00
53 b = 0.45 14.63 -4.74
c = 5.15 5.15 9.87
$Zr_2Y_2O_7$–Y2a 121 114 a = 10.02 0.04 0.04
116 b = 0.00 4.87 -4.87
c = 0.08 9.85 9.85
$Zr_2Y_2O_7$–Z2a 600 605 a = 4.95 4.95 0.00
608 b = -4.95 4.95 0.00
c = 0.00 0.00 19.86

a different number of atoms using the energy per ZrO_2 constituent in the fluorite phase (E_f) and the energy per Y_2O_3constituent in the bixbyite phase (E_{BX}):

E_H = (E(YSZ) - (# of ZrO_2)*(E_f) - (# of Y_2O_3)*(E_{BX}))/ (# cations). (5.1)

In Table 6, we have calculated the formation energy of a $Zr_3Y_4O_{12}$ structure, the same stoichiometry as the yttria-zirconia δ-phase.[29,30] We also give the formation energy of four $Zr_2Y_2O_7$ structures along with the lattice vectors of the structures. The sets of two values from SEQQUEST correspond to the use of an oxygen basis at the vacancy site and the other without a basis set. We obtained a slightly better accuracy for a basis set at the site of the vacancy. The difference in formation energy is not significant; but in order to have uniformity in the functional space, we chose to use vacancy orbitals. The formation energies from SEQQUEST lie very close to the VASP values and substantiate the use of a gaussian LCAO basis set for this system.

Table 6 Formation energy of YSZ structures in units of meV/cat and lattice vectors in a.u.

The VASP calculations used an energy cutoff of 495 eV. We use a 4x4x4 **k**-point mesh for the $Zr_3Y_4O_{12}$ structure and an equivalent[31] 4x4x4 **k**-point mesh to the primitive fluorite unit cell for the four $Zr_2Y_2O_7$ structures. The SEQQUEST runs used a 4x4x4 **k**-point mesh for the $Zr_3Y_4O_{12}$ structure, 6x6x2 for the V2a, 6x3x4 for the W2a, 5x6x3 for the Y2a, and 6x6x4 for the Z2a.

Accuracy of Localization

c Forces Initially, we wanted to look at the accuracy of the forces for different localization regions. We started with a 24-atom unit cell obtained from doubling along each lattice vector the 3 atom primitive cell of Zirconia in the fluorite structure. One Yttria defect was inserted with the Y nearest neighbors (n.n.) to the vacancy. We centered the occupied orbitals on the atoms with four on oxygen and three on the cations. The initial guess for the four orbitals on oxygen was strictly the single zeta s and p, and for the three orbitals on the cations was the single zeta p.

In Fig. 4A, we compare the magnitude of the rms error of the vector difference of
the forces obtained with the optimization with the forces from diagonalization on this defect system at the initial geometry of the ideal fluorite crystal. The horizontal axis is an approximate radius since the multiple radii for different basis functions prevents a pure definition of the localization radius. This plot shows the nice qualitative feature of the convergence of the forces with increasing radius. Numerically the only information from this plot is that the smallest localization regions are unusable. For more numerical insight and quantitative results, Fig. 4B plots the natural logarithm of the maximum error in the vector force difference. Even though we used an insufficient k-point sampling (Γ point) for these calculations, it is acceptable as we are interested only the comparison with diagonalization, which we also used only the Γ point.

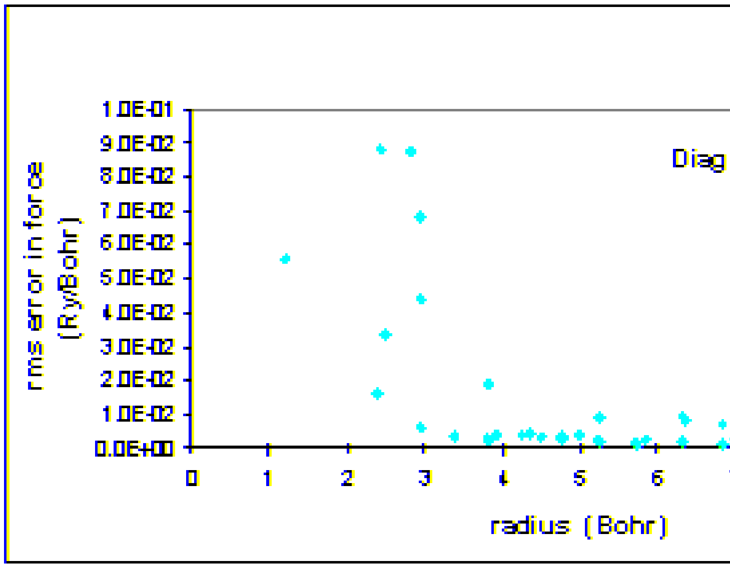


Figure 4A For 4 atom Zirconia unit cell with one defect, the rms error in force vs. approximate localization radius

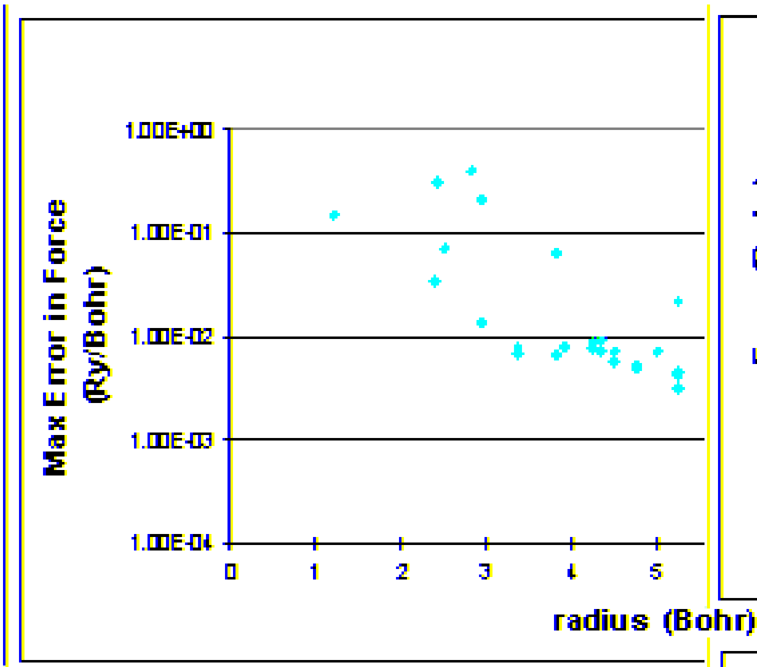


Figure 4B For 4 atom Zirconia unit cell with one defect, the maximum error in force vs. approximate localization radius

From these results, we were able obtain insight for acceptable settings of the localization radius. The localization region of the orbitals on the cations can be smaller as these are semi-core states. For the orbitals on the oxygen atoms the inclusion of the polarization function on the nearest oxygen atoms is crucial. The orbitals on the cations require either the single or double zeta functions on the n.n. atoms, i.e. oxygen. One does not gain much in accuracy by increasing the localization region for the orbitals on the oxygen if the localization regions for the orbitals on the cations are too short and vice versa.

d Relaxation We next check the accuracy of a geometry relaxation of the better settings of the localization region. We use an 81-atom, 3x3x3 generated unit cell. The initial geometry has the atoms in the ideal fluorite positions. The starting bond length is 4.15 Bohr. The number of geometry steps for a relaxation is typically about 8-10. The criterion for self-consistency is a change in the energy of 5μRy/atom.

For the configuration (1a) defined as a single defect with both Y n.n. to the vacancy, Fig 5A plots the magnitude of the vector difference of the final atomic positions vs. the fraction of sparsity. The localization setting of Zr-5,0,0; Y-0,5,5; and O-4.4,4.4,5.4 is the smallest region to achieve full relaxation.

Fig. 5B plots the same data for an 81-atom system at the more appropriate 8 mol % doping, i.e. 2 Y_2O_3 defects. This configuration, labeled (2a), has 2 vacancies separated by a Zr and no Y n.n. to vacancies. A configuration is considered relaxed if the magnitude of the largest force on any atom is below 2x10^{-3} Ry/Bohr.

.

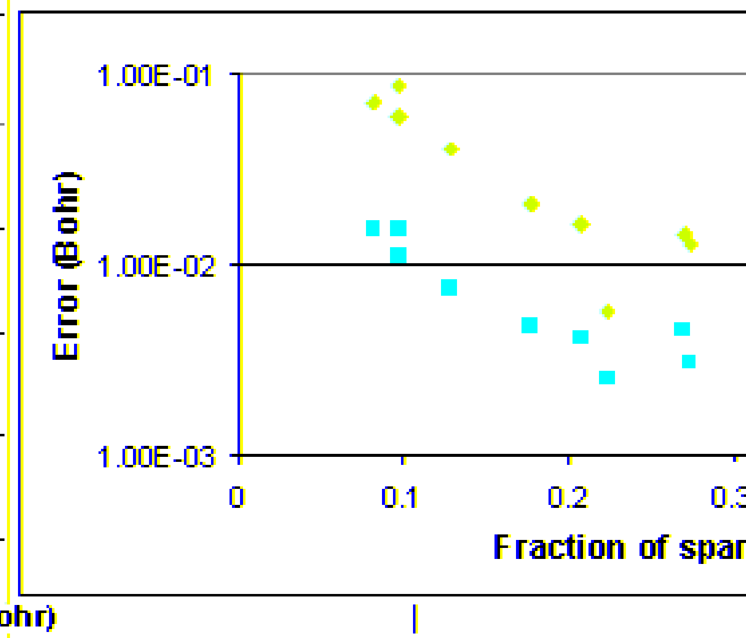


Figure 5A error in final relaxed positions vs. fraction of sparsity for configuration 1a

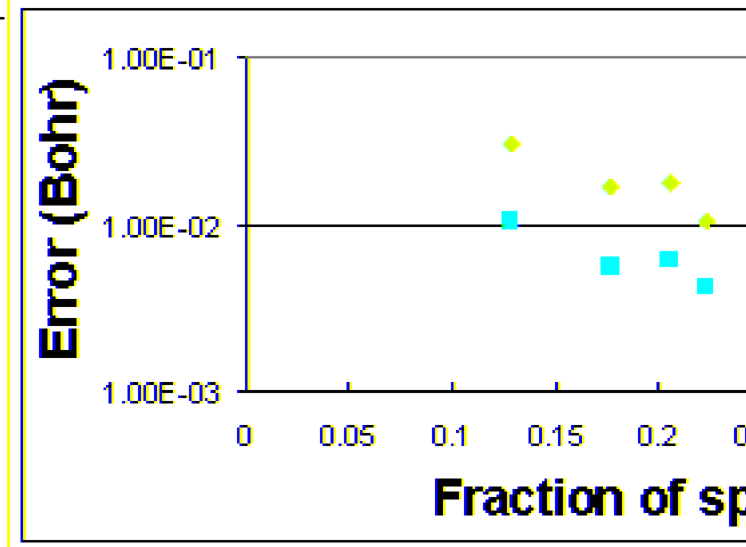


Figure 5B error in final relaxed positions vs. fraction of sparsity for configuration 1b.

For the (2a) configuration, the code using diagonalization has problems relaxing if the vacancy basis functions are allowed to move. The vacancy functions started to move appreciably just before the relaxed geometry was achieved and the forces never reached convergence. If the vacancy functions are not allowed to move, a relaxed geometry is achieved. The calculations with the GCG

Diag	0.081	0.127	0.206	0.418	0.272	0.222	
1a	31.6	5.6	17.7	16.6	22.9	13.1	10.0

	1a	1b	2a	2b	2c
LR1	17.7	9.7	78.5	43.5	-13.9
Diag	31.6	24.2	95.2	58.6	-0.9

algorithm did not have this problem. A localization region labeled (LR1), incorporating all gaussian functions up the second n.n. of the O orbitals and the first n.n. on the cations, was necessary in order to relax the (2a) configuration. Note that this is larger than the necessary localization region for the (1a) configuration.

For perspective, we compared the final atomic positions for two calculations starting with different initial positions. We use the ideal fluorite positions and the arbitrary choice of the positions of the 11^{th} geometry step in the unsuccessful run with diagonalization when vacancy functions were allowed to move. The vacancy functions were fixed to the starting positions of the fluorite crystal. The final energies were the same within 0.5 mRy, but the final positions differed by a rms of 0.0103 Bohr and a maximum of 0.025 Bohr. This result suggests that localization described above is sufficient given the inherent error with diagonalization.

e Relative and Formation Energies Formation energies are important to calculate the energetically favorable configurations and doping concentrations. In order to use Eq. (5.1), we need to calculate the energy of ZrO_2 in the fluorite and Y_2O_3 in the bixbyite for each localization setting. From **k**-point calculations of the primitive unit cells, we determine the Γ point to be sufficiently converged for 192 and 320 atoms respectively, and we use these system sizes to calculate the constituent zirconia and yttria energies. A growth parameter of $5\text{x}10^{-4}$ was sufficient in that a change of only 0.5 meV/cation was seen compared to full growth.

Table 7A gives the formation energy (row2) for diagonalization and for several localization settings (row 1), given by fraction of sparsity, for the 1a configuration. The FCC lattice vectors were slightly enlarged over the pure zirconia values. For the 81-atom unit cell the pure zirconia lattice vectors would have a length of 20.259 Bohr. The stress relaxed length is 20.314 Bohr. There is a concern that the energies do not approach the diagonalization monotonically. This is not a breakdown of the variational principle, but a result of all of the energies in Eq. (5.1) monotonically increasing at different rates.

Table 7A Formation energy for configuation (1a) in meV/cation.

In Table 7b, we give the formation energies for several more configurations with the localization region (LR1). The additional configurations as follows: (1b) – 1 defect with 1 Y n.n. (2b) – vacancies as in 1a and no Y n.n. to vacancy, and (3) 2c – vacancies with more separation and no Y nn to vacancies. The error from diagonalization is approximately the same for each configuration. This consistency reflects accurate relative energetics. We get similar consistency for other localization regions, which were done for a few of the structures. The same size cell was used for these additional structures as for the 1a structure.

Table 7B comparison of formation energy (meV/cation) for several configurations

With localization region of Zr-5,0,0; Y-0,5,5; and O-5.4,4.4,5.4, the formation energies for the 192 and 375 are –8 meV/cation and -14.7 meV/cation. Given that the 81-atom case is off by 26 meV/cation these could be scaled to 18 and 11.3. Physically this is reasonable that the formation energy should decrease as the doping concentration decreases. The energy/cation tends toward the value of zirconia thus making the formation go toward zero in the limit of one defect in an infinite zirconia solid.

f Computational effort With localization region of Zr-5,0,0; Y-0,5,5; and O-5.4,4.4,5.4, the subspace optimization uses less memory at 192 atoms than diagonalization. At 375 atoms, there is a three-fold savings: Optimization ∼ .5Gb, Diagonalization ∼ 1.5Gb. Using 15 GCG iterations for 375 atoms, the subspace optimization has a two-fold savings in CPU time: Subspace ∼ 2,500 seconds, Diagonalization ∼ 5,000 seconds. The total time for the relaxation, 10 geometry steps, of the 375 atom relaxation was 162 hours. We used 15 GCG iterations, but possibly less might be acceptable. This is investigated more in the next section.

Physical Comments

Looking at the formation energies of the configurations, it appears that the Y prefer to be far from the vacancies. This is counter-intuitive by a simple charge analysis. Formally a Y is a point defect with a charge of –1, and the vacancy has charge +2. So one would expect them to prefer to be close. This is true for the energy in the ideal fluorite positions. With relaxation, the cations move away from the vacancy allowing the oxygen and its electrons to occupy the vacant space. If a Y is next to the vacancy, its attraction to the +2 charge makes it move less farther away than a Zr from the ideal fluorite positions. Since the Y moves less the O can't move as much into the vacancy and thus not allowing for a lower energy.

Deep Level defects of N for Si in SiC

Linear scaling algorithms and occupied subspace optimizations incorporating localization typically allow only for full occupancy of the orbitals. This is the result of not using eigenfunctions that are necessary once partial occupancies are involved. We present in this section an algorithm to handle partial occupancy. A deep level defect is a good starting point to generalizing these algorithms to partial occupancies. In metallic states many states are extended. In a deep level defect only the defect-state is extended compared to the localized orbitals in the rest of a semiconductor or insulator. As only one state is extended the rest of the system can be handled with reasonably localized states.

Defects of materials such as silicon carbide are important to understanding real world properties of non-pure materials. They are also important in understanding the properties of doped materials. In both cases a large number of atoms are necessary to investigate defect–defect interactions and many different doping concentrations. Therefore it is desirable to adapt recent algorithmic developments, previously restricted to systems with a gap. We use a nitrogen substitution of Si in SiC as a system to test the algorithm.

Algorithm for Partial Occupancies

We give a brief review of the general algorithm given in Chapter 3. For each self-consistent iteration, instead of solving iteratively or exactly the generalized eigenvalue equation,

$\mathbf{H}\Psi=\mathbf{S}\Psi\mathbf{E}$, (5.2)

the trace of a generalized Rayleigh quotient is minimized

$\mathrm{Tr}[\ (\Phi^\dagger\mathbf{S}\Phi)^{-1}\ \Phi^\dagger\mathbf{H}\Phi\ ]$. (5.3)

The orbital matrix, Φ, is an M by N matrix of the coefficients for the N localized nonorthogonal occupied orbitals. We implement a Grassmann conjugate gradient algorithm to minimize this trace. For systems which have all eigenvectors fully occupied the density matrix is simply calculated by using

$\mathbf{P} = \Phi(\Phi^\dagger\mathbf{S}\Phi)^{-1}\Phi^\dagger$. (5.4)

For partially occupied systems the eigenvectors near the gap have to be found. Let us now assume that the N lowest eigenvectors are fully occupied and the N+1 eigenvector is singly occupied. We accomplish this by solving the smaller generalized eigenvalue equation in the space of K localized non-orthogonal orbitals where K $\geq$ N+1.

$(\Phi^\dagger\mathbf{H}\Phi)\Psi_K = (\Phi^\dagger\mathbf{S}\Phi)\Psi_K\mathbf{E}_K$. (5.5)

The density matrix is now given by

$\mathbf{P} = \sum_{i=1}^{K} f_i\Psi_i\Psi_i^\dagger$ (5.6)

with f_i defining the occupation of the i^{th} eigenvector. The value of f_i is handled in straightforward manner.

This algorithm adds an additional $O(N^3)$ step to the two previous $O(N^3)$ steps of calculating $(\Phi^\dagger\mathbf{S}\Phi)^{-1}$ and the multiplication of $(\Phi^\dagger\mathbf{S}\Phi)^{-1}\ (\Phi^\dagger\mathbf{H}\Phi)$. This is not a concern for system sizes up to 1,000 atoms. Once the system size demands a truly linear scaling algorithm this method for calculating singly occupied states must be modified. Alternatives are to solve for the space of the lowest N eigenvectors by a the linear scaling algorithm, and then find the N+1 eigenvector by minimizing

$\Psi\ \ {}^\dagger\mathbf{H}\Psi$, (5.7)

while keeping it orthogonal to the space of the N localized orbitals. Another alternative is to solve for the space of K localized orbitals, and then find the largest eigenvectors of this space. These eigenvectors are weighted properly and their contribution to the density matrix is subtracted from the total density matrix given in Eq. (5.4).

Dense Calculations

If the system is started in the T_d symmetry then the final state is basically triply degenerate. One p-state has an eigenvalue of only 10^{-5} Ry lower than the other 2. If the system is perturbed into a C_{3v} symmetry then the energy gap of the lower p state becomes more pronounced at 5.4 mRy. For a 64-atom unit cell the C_{3v} symmetry is 2mRy lower.

We looked at two configurations of a nitrogen atom for Si defect in SiC for a 64-atom system. In configuration (C1), nitrogen directly takes the position of a Si atom at the origin. In (C2), the nitrogen atom is placed as in (C1) and then switched with the furthest n.n. C in the C_{3v} symmetry. The starting geometry for both configurations is the C_{3v} symmetry. In (C1), the nitrogen is moved -0.720 Bohr in the −1,-1,-1 direction, the n.n. C in 1,1,1 direction is moved 0.401 in the −1,-1,-1 direction, and the other n.n. C are moved 0.459 closer to the N. In (C2), the N is moved -0.945 Bohr in the −1,-1,-1 direction, the n.n. C in 1,1,1 direction is moved 0.487 in the −1,-1,-1 direction, and the other n.n. C are moved 0.457 closer to the N.

With diagonalization, Table 8 gives the difference in the energy in the relaxed geometry. The lattice vectors remained cubic and were optimized to 23.052 Bohr from the 23.165 Bohr. This was an optimized unit cell for the 1^{st} configuration. The stress on the unit cell for the 2^{nd} configuration was small for the diagonal elements, and tolerable for the off-diagonal elements. As we are mostly interested in comparing the diagonalization and the optimization method, just using identical geometries is most important.

E_{C1}-E_{C2} (Ry)
64 atoms 0.2833
216 atoms 0.2545

Table 8 Relative energy difference between the two configurations with diagonalization

We next look at a 216-atom system with the single defects described above. We use the unit cell of the ideal SiC crystal. The differing relative energy values for the two system sizes in Table 8 are caused partly by our not using an optimized unit cell for all structures. The starting geometry corresponds to the 64-atom case. The calculation with diagonalization showed that the gap in the eigenvalues between the occupied p-state and the unoccupied p-states shrinks compared to the 64-atom system. In the first configuration the gap is 1.6 mRy and for the 2^{nd} configuration the gap shrinks from 37 mRy to 17 mRy.

We looked at the effects of scaling back the cutoffs. By multiplying 100 times the default cutoffs for the calculation of the Hamiltonian matrix elements, we had problems converging (C1). We also tried increasing the mesh spacing for the solution of the Poisson's equation. We have previously used 0.273 Bohr and scaled back to 0.4095. We also scaled back the cutoff for inclusion of a gaussian function onto a box in the grid from the default of 10^{-8} to 10^{-6}. The relative energy accuracy was comparable at 0.253 Ry.

Sparse Calculations

For the sparse calculations on the 64-atom unit cell, we put 4 orbitals on the nitrogen atom and 4 on the carbon atoms creating a space of 132 orbitals in which we diagonalize. We looked at having one orbital on the nitrogen but found this to be inferior in a calculation with no localization. Table 9 gives the relative energies for localization energies that gave relaxed geometries. Our starting geometry was the same as for the calculations using diagonalization. 8-12 geometry steps was typically sufficient to obtain a magnitude of any force smaller than $2x10^{-3}$.

For (C1), we first tried bond-centered orbitals including all gaussians from the closest 8 atoms. This setting did not relax the atom positions. We then tried atom-centered orbitals including all gaussians within 7 Bohr. We were able to relax the structure, but the error in the energy was greater than desired, about 1 mRy. A setting with the nitrogen orbitals occupying the full variational space gave a good relaxed geometry for (C1). This setting for the orbitals did not give as accurate a geometry for (C2). The inaccurate geometry and the smaller relative energy suggested that (C2) needed more variational flexibility near the defect.

64 atom E_{C1}-E_{C2} (Ry)
S1 – all 7 Bohr 0.2661
S2 – S1 + N full 0.2146
S3 – S2 + n.n. C full 0.2837
S4 – S1 + N and 1 n.n. C 9 Bohr 0.2830
S5 – S4 + 2 n.n. C 9 Bohr 0.2822

216 atom E_{C1}-E_{C2} (Ry)
S6 – S1 + N and n.n. C 13 bohr 0.2534
(C1) and (C2) near minimum
S4 - (C1) and (C2) near minimum 0.2346
S4 – (C1) near minimum and 0.2764
(C2) original positions

With extended orbitals on the n.n. C also, both configurations had accurate final geometries and the relative energy was within our desired accuracy. We also tried a localization of 9 Bohr on the n.n. C and on the second n.n. C. For (C2) using bond-centered orbitals for the nitrogen and n.n. C fully extended and the other C with a radius of 5 Bohr for all shells, the time was longer than for the comparable run with atom-centered orbitals with a 7 Bohr radius. So bond-centered orbitals did not seem to be a benefit for this system,

Table 9. Relative energy difference between the two configurations with different localization region.

Typically 15 GCG iterations has been used in our previous use of the algorithm. In an effort to achieve more efficiency, we tried 10 iterations. This setting required significantly more SCF iterations on the 1^{st} and 2^{nd} geometries, but the same for the subsequent geometries. The increase in SCF iterations was about 50% which, neglecting for time in other areas of a SCF cycle, was exactly compensated by the fewer GCG iterations. With the S3 localization setting, the difference in total energy for the two structures is less than 1 mRy, and the relative energy was 0.2820 Ry.

Table 10 Relative energies for the 216-atom unit cell. Growth parameter $5x10^{-4}$.

Table 10 gives the relative energies between relaxed 216-atom unit cells of the two defect configurations. In order to decrease the computational effort, we started from the positions from the diagonalization calculations, labeled "near minimum". The S6 localization region gives a value within our desired accuracy of 1mRy. The S4 localization region, which gave the desired accuracy for the 64-atom unit cell, did not achieve the accuracy for two combinations of starting atomic positions. This loss in accuracy may be due to the initial positions of the atoms in both configurations not being consistent (see section 5.4).

A value of 5x10^{-4} for *cut_gro* was used for all of the equations in Table 10. A calculation with all of the default cutoffs and the 1^{st} localization on configuration (1) with a growth parameter of 10^{-3}gave forces that did not monotonically decrease. The energy decreased, but as the forces jumped around this was seen as an inferior setting to 5x10^{-4} , which gave monotonically decreasing forces.

For the 216-atom unit cell, we also looked at scaling back the cutoffs in order to increase efficiency. We use: (1) a grid spacing of 0.4095 Bohr for the solution of Poisson's equation and 10^{-6} for *convgr* (see Sec. 4.4.2), (2) a factor of 100 larger cutoff criteria for the setup of **S** and **H**, (3) 10^{-3} for *cut_gro*, and (4) the S6 localization setting. With these settings, the relative energy is 0.2650 Ry. The benefit of increasing the setup cutoffs is mainly in the set up time, about 2 times faster. The matrices are slightly more sparse, but not significantly so to shorten the time for the sparse multiplication. The time spent on the grid is about 4.25 times faster. The overall saving was about a speedup 1.5. These faster cutoffs were not extremely advantageous, as the optimization of the ground state is the biggest consumer of time so it was not very beneficial to decrease the other parts of the code.

For configuration (C2), the effects of the number of the maximum GCG iterations for these cutoffs showed less than 1 mRy change in energy for a value of 10 to 15. Below 10, we saw a 6 mRy drop. For these cutoffs we also looked the relaxation with a localization of 14 Bohr on the N and n.n. C and had problems relaxing. We also had problems relaxing for a localization region of 15 Bohr with the most accurate settings and cutoffs.

Issues for Obtaining Quantitative Accuracy

False Minima

Other papers have reported finding local minima[32,33] for the minimization of E_B. This was stated as the result of the localization. The local minima are not found when the full vector space is used for the orbitals, i.e. no localization. A solution[34] was presented in which a larger *occupied* subspace was calculated and then the occupations within this space were resolved. This was different in technique but similar in spirit to the method presented in section 5.3.

What we typically find is a stalling of the self-consistent minimization procedure. For some calculations, a local minimum should not be assumed, as self-consistency has not been achieved by an analysis of the maximum change in any self-consistent potential matrix element even though from an energetic point of view a local minimum has been found. The GCG algorithm has stalled for one or more SCF steps.

YSZ (1a) Energy (Ry)
LR1 Relaxed -3651.1443
LR1 Single energy at relaxed geometry -3651.1318
Nitrogen defect 64-atom (C2)
S4 Relaxed -632.5009
S4 Single energy at relaxed geometry -632.4862

Table 11. Energies at final positions w/ and w/o relaxation from an initial geometry.

In Table 11, we compare the energy obtained at the end of a relaxation to the energy obtained with a calculation at the final geometry with no relaxation. The energies are lower for the relaxation as the minimization procedure has started at later geometries with a Φ closer to the electronic energy minimum. A path for Φ is followed with the atomic movements allowing the GCG algorithm to break free from what could be called a local minimum. If the minimum is found too quickly, e.g. a single calculation at the final geometry, the GCG minimization is not allowed the time to slowly find the minimum or the ability to evade the " local minima". Depending on how far two configurations start from the final positions, the final relative energies may be inaccurate. This may manifest itself in the low formation energies in Table 7B, which include a calculation of a single geometry for the reference energy of fluorite and bixbyite.

If the SCF procedure is restarted with the old guess for Φ, the energy can be lowered. This method may allow one to escape a local minimum. These results are in Table 12. The self-consistent potential can be started from the atomic reference or from the last SCF iteration. A minimum is only assumed if self-consistency has been achieved for the self-consistent matrix elements. For some calculations, a local minimum with self-consistency is still found. A maximum change of 10^{-4} for any self-consistent matrix element is sufficient. The energy difference is smaller than in the case of the stalling with no self-consistency.

YSZ (1a) Energy (Ry)
LR1 Final geometry with Self-consistency -3651.1378
LR1 Final geometry after several re-input of Φ -3651.1412
Nitrogen defect (C2)
S4 Final geometry after 3 inputs of Φ -632.5019

Table 12. Energies at final positions with different starting values for Φ.

Relaxations

Typically one may not have enough time to fully relax a system during a run. This may be due to time constraints on the system that one is running. As mentioned above one should save Φ from the previous run. This will facilitate the removing of the local minima or critical slowdown in the GCG algorithm. The previous Φ will be used if the variable *dophi* in *INIT_PHI* and consequently inputted in *INIT_PHI.DAT* is set to *false*. One also has to be careful that the localization radius gives the same sparsity pattern. Since the atoms have moved the same values may not give the same sparsity pattern. At some point the code should be made to use the old sparsity pattern also. This will entail saving *nhbr_occ_max* and *nhbr_pr_max* as well as altering *init_phi*.

Consistency

LCAO calculations inherently require more consistency in the calculation as the basis sets are usually not as converged as in plane wave calculations. Since we are looking at relative convergence the two calculations must be very similar for a cancellation of errors. This is also true for plane wave calculations with regards to using equivalent **k**-points. Efficiency can be made here as the **k**-point sampling is not necessarily fully converged, but the **k-**space integration is equally represented in the two calculations.

Now with localization, one has to pay even more attention to consistency than in the typical LCAO calculation. The localization regions must be consistent. This is not straightforward for systems that have different atomic geometries. Ideally one should keep the same number basis functions for each type of orbital. This may require different localization radii. If there is a drastic difference in atomic positions this may not be the best course of action. One may just want to keep the localization radii the same.

One also now has to worry about the path to the electronic ground state. One has to ensure that the initial geometries are comparable so that a similar number of geometry steps are made. This will allow equal time for the GCG algorithm to overcome any critical slowdowns or local minima. One could also implement a self-consistent procedure by restarting with the final Φ once the final geometry is found.

1 B. G. Pfrommer, S. G. Louie, and H. Simon, Proc. Eighth SIAM

Conf. on Parallel Processing for Sci. Comp., 8 (1997)

2 J. R. Chelikowsky, J. Phys. D **33**, R33 (2000)

3 J. Bernholc et. al., Phys. Stat. Solidi B **217**, 685 (2000)

D. Porezag and M. R. Pederson, Phys. Rev. A **60**, 2840 (1999)

P. Schultz and P. J. Feibelman, to be published.

C. Ochsenfeld, C. A. White, and M. Head-Gordon, J. Chem. Phys. **109**, 1663 (1998).

K. N. Kudin and G. E. Scuseria, Chem. Phys. Lett. **289**, 611 (1998).

P. Ordejón, D. A. Drabold, R. M. Martin, et al., Phys. Rev. B **51**, 1456 (1995).

E. Tsuchida and M. Tsukada, J. Phys. Soc. Japan **67**, 3844 (1998).

D. R. Bowler and M. J. Gillan, Comp. Phys. Com. **120**, 95 (1999).

J. M. Millam and G. E. Scuseria, J. Chem. Phys. **106**, 5569 (1997).

M. Challacombe, J. Chem. Phys. **110**, 2332 (1999).

P. Ordejon, Phys. Stat. Sol. B **217**, 335 (2000).

V. A. Gubanov and C. Y. Fong, Appl. Phys Lett. **75**, 88 (1999).

P. Käckell, B. Wenzien, and F. Bechstedt, Phys. Rev. B **50**, 17 037 (1994).

C. H. Park, B.-H. Cheong, K.-H. Lee, et al., Phys. Rev. B **49**, 4485 (1994).

S. Limpijumnong and W. R. L. Lambrecht, Phys. Rev. B **57**, 12 017 (1998).

H. J. Monkhorst and J. D. Pack, Phys. Rev. B **13**, 5188 (1976)

G. Stapper, M. Bernasconi, N. Nicoloso, et al., Phys. Rev. B **59**, 797 (1999).

G. Jomard, T. Petit, A. Pasturel, et al., Phys. Rev. B **59**, 4044 (1999).

J. P. Goff, W. Hayes, S. Hull, et al., Phys. Rev. B **59**, 14 202 (1999).

B. H. O'Conner and T. M. Valentine, Acta Cryst. **B25**, 2140 (1969).

G. Kresse and J. Hafner, Phys. Rev. B **47**, 558 (1993).

G. Kresse, Thesis, Technische Universit at Wien 1993.

G. Kresse and J. Furthmuller, Comput. Mat. Sci. **6**, 15 (1996).

G. Kresse and J. Furthmuller, Phys. Rev. B **54**, 11169 (1996).

G. Kresse and J. Hafner, J. Phys. Condens. Matter **6**, 8245 (1994).

D. Vanderbilt, Phys. Rev. B **41**, 7892 (1990).

H. G. Scott, Acta Cryst. **B33**, 281 (1977).

S. P. Ray, V. S. Stubican, and D. E. Cox, Mat. Res. Bull. **15**, 1419 (1980).

S. Froyen, Phys. Rev. B **39**, 3168 (1989).

F. Mauri and G. Galli, Phys. Rev. B **50**, 4316 (1994).

P. Ordejón, D. A. Drabold, R. M. Martin, et al., Phys. Rev. B **51**, 1456 (1995).

J. Kim, F. Mauri, and G. Galli, Phys. Rev. B **52**, 1640 (1995).

SUMMARY

We have presented an approach for an occupied subspace optimization (minimization) to calculate the electronic ground state of a condensed matter system. We have used Density Functional Theory (DFT)[1,2] as our quantum mechanical method with a gaussian basis and pseudopotentials.[3,4] The minimization algorithm is a Grassmann conjugate gradient[5] (GCG) method. The resulting wavefunctions of the minimization are localized non-orthogonal orbitals. Taking advantage of this localization, we have made the approximation of strict localization by limiting orbitals to have a non-zero contribution from selected gaussian functions within a localization region. The result is an algorithm that has the dominant parts scale as O(N), for N the number of electronic states. The algorithm has shown to be useful in calculations on complex systems with quantitative accuracy.

In Chapter 1, we motivated the use of computer simulations for scientific applications and the usefulness of new algorithms. We briefly introduce some computational aspects of solving the equations of DFT and the traditional method of solution involving eigenfunctions. We then introduce the concept of localized orbitals as an alternative to eigenfunctions. The essential point is that for certain applications localized orbitals are superior over eigenfunctions. We then added the property of "nearsightedness"[6], resulting in Linear Scaling, as the benefit of localized orbitals that we utilize.

In Chapter 2 as a background analysis of quantum mechanical calculations, we describe the many approximations used in modern electronic structure calculations. We then lead from these many approximations to the approximation of strict localization that is at the heart of Linear Scaling. There are a great many approximations in these calculations and is not trivial to incorporate all properly. It is easy to accept some approximations as being exact or completely converged because of their ubiquitous use. It is very helpful for even the most experienced user of code to remind oneself of the approximations so that the inherent inaccuracy of calculations is not forgotten.

As much of this thesis revolves around the minimization of the electronic energy, a section on general minimization concepts was presented at the beginning of Chapter 3. We emphasize the familiar relationship that diagonalization is identical to a constrained minimization. We also present an unconstrained minimization technique of a generalized Rayleigh quotient to be equivalent to the diagonalization of a Hamiltonian, for a system with a gap between the occupied and unoccupied eigenvalues. With a basic understanding of the vector spaces in place, we then introduce the concepts and issues in Linear Scaling. One such issue was the multiple length scales on which localization occurs. This analysis sheds light on the decisions that were made for our approach to obtaining the *occupied* space, the vector space spanned by the occupied eigenfunctions.

Based upon recent mathematical work[5,7], we introduce the concept of a constraint manifold, the Grassmann manifold. This manifold is inherent in any method to obtain the *occupied* space within an orbital or density matrix formulation. It is the adhering to this manifold which is the difficult goal of variational Linear Scaling algorithms. The Grassmann manifold is described as having relevance even in our unconstrained minimization algorithm.

With the knowledge of the Grassmann manifold in place, we proceeded to describe the GCG algorithm. We detailed the properties of a gradient, an obvious necessity for a conjugate gradient algorithm, with emphasis on it residing in the hyper-plane tangent to the Grassmann manifold. Our definition of the inner product that must accompany the definition of a gradient is given as the only tensorally correct formula. A distinction between a differential and a gradient for representations of nonorthogonal basis functions is remarked from the outset.

The formulas for the gradient and differential are simple formulas in terms of matrices. It is evident from these formulas that the differential does not satisfy the requirements of a gradient, which are satisfied by our definition of the gradient. The formulas for the creation of search direction and an initial guess for the step size are also given. The concept of parallel transport for keeping the gradient and the search direction in the tangent plane after the orbitals have been updated is briefly discussed with the corresponding formulas.

Having presented the conceptual framework for the minimization algorithm, we then proceeded to discuss our implementation with strict localization into a serial code, SEQQUEST. This is mostly a detailed discussion of the code designed for those who wish understand the specific steps and for those who wish to adapt the code for future use. The result of sparsity due to localization is discussed. The sparse storage format and method for accessing elements is described with direct reference to subroutine and variable names.

Once we described the basic elements of the sparse implementation of the GCG algorithm, we added some changes to the GCG algorithm necessary for sparsity and hence Linear Scaling. We addressed fundamental changes in the algorithm and some numerical issues and problems. The multiplication of sparse matrices resulting in sparse matrices came next as an important and vital step in Linear Scaling. The technique and our rational for the numerical algorithms are given, again with explicit reference to subroutines and variable names. We have attempted to use optimized libraries of Basic Linear Algebra Subprograms (BLAS) whenever possible.

We also detailed modifications to existing routines for the calculation of the charge density, self-consistent matrix elements, and the forces. In these discussions, we addressed the operation count as we have with other routines, which is at minimum required to ensure linear scaling. The calculation of the density matrix, which is used in the calculation of the charge density among other calculations, was also included.

Having presented the theory and specifics of implementation, we proceeded to apply the code to several systems of interest. The accuracy to resolve the relative energies of 64-atom unit cells of the 3C, 2H, and 4H crystal phases of silicon carbide (SiC) was mapped out for many localization regions. From this information, we explain that the desired accuracy will dictate which localization region to use. Then by looking at timings for different localization regions vs. number of atoms, one will know whether the optimization with localization is more efficient than diagonalization.

Wanting to test the algorithm for the accuracy of forces and on a more complex material, we moved onto to investigating yttria-stabilized zirconia (YSZ). We analyze the accuracy of forces, atomic positions after relaxation, relative energies of several configurations, and the formation energies of these configurations. These results show that our method and code show promise to be used for quantitative accuracy. For YSZ, the code has benefits over diagonalization at around 192 atoms. Due to the added complications of the implementation of Linear Scaling though, the code probably should not be used until diagonalization becomes impractical or impossible with one's computer resources, typically around 400-500 atoms.

Finally we adapt the algorithm to handle partial occupancies. Our specific choice was a singly occupied state deep level defect state. The system was a nitrogen atom substituting for a silicon atom in SiC. We found that the localization region for the orbitals near the defect needed to be larger than the localization region for the other orbitals of the system. We also noticed that the final energy was different for different starting atomic positions. Addressing this appearance of local minima, we detailed issues to confront for getting the proper quantitative results.

Having investigated the geometrical aspects of the constraint (Grassmann) manifold and multiple length scales, our algorithm has proven to have quantitative application to system sizes between 300-1500 atoms. There are still issues concerning convergence and local minima with localization that need to be answered. Hopefully the mathematical community might help with this endeavor. Other possible future work is to develop an asymptotically O(N) method that adheres to the Grassmann manifold after every update of the orbitals.

P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).

W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).

D. R. Hamann, Phys. Rev. B **40**, 2980 (1989).

N. Troullier and J. L. Martins, Phys. Rev. B **43**, 1993 (1991).

A. Edelman, T. A. Arias, and S. T. Smith, SIAM J. on Matrix Anal. Appl. **20**, 303 (1998).

W. Kohn, Phys. Rev. Lett. **76**, 3168 (1996).

R. A. Lippert and M. P. Sears, Report SAND99-2986, Sandia National Laboratories, Abuquerque, NM USA, (1999).