Real-Space Localization Methods for Minimizing the Kohn-Sham Energy

by

Marc Millstone

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctorate of Philosphy
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
May 2011

Michael Overton

To Anne: your love, kindness and support have made all of this possible

Abstract

The combination of ever increasing computational power and new mathematical models has fundamentally changed the field of computational chemistry. One example of this is the use of new algorithms for computing the charge density of a molecular system from which one can predict many physical properties of the system.

This thesis presents two new algorithms for minimizing the Kohn-Sham energy, which is used to describe a system of non-interacting electrons through a set of single-particle wavefunctions. By exploiting a known localization region of the wavefunctions, each algorithm evaluates the Kohn-Sham energy function and gradient at a set of iterates that have a special sparsity structure. We have chosen to represent the problem in real-space using finite-differences, allowing us to efficiently evaluate the energy function and gradient using sparse linear algebra. Detailed numerical experiments are provided on a set of representative molecules demonstrating the performance and robustness of these methods.

ACKNOWLEDGMENTS

The number of people people that have helped me achieve my academic goals is large and any finite list is bound to be incomplete. First, I want to thank Andy Hicks for opening my eyes to interesting computational problems from optics as well as being a good friend throughout my entire education.

To my advisor, Michael Overton, I thank you for teaching me the joys of optimization, linear algebra and good writing. To both Juan Meza and Chao Yang, I thank you for taking a risk on a young graduate student in search of a thesis topic. Your patience and time has been and will always be appreciated.

Finally, I wish to thank my wonderful wife Anne. Your support and guidance have made me both a better researcher and a better person. Without you, none of this would have been possible (or as enjoyable).

Table of Contents

List of Figures

List of Tables

List of Algorithms

1

Electronic Structure and the Kohn-Sham energy

Proposed in 1926, the many-electron Schrödinger equation describes the fundamental behavior of non-relativistic electrons. Although there exist special cases where its solution can be obtained analytically, most interesting problems require the numerical solution of the large eigenvalue problem

$$\mathcal{H}\Psi(r_1, r_2, \ldots, r_{n_e}) = \lambda\Psi(r_1, r_2, \ldots, r_{n_e}) \tag{1.1}$$

where $\mathcal{H}$ is the many-body Schrödinger Hamiltonian and $\Psi(r_1, r_2, \ldots, r_{n_e})$ is the many-body wavefunction, that is a function of the position, $r_i \in \mathbb{R}^3$, of the n_e electrons. Additionally, because of physical constraints, these wavefunctions must be orthogonal and antisymmetric. Mathematically, this is equivalent to

$$\int \Psi^*\Psi = 1, \tag{1.2}$$

$$\Psi(r_1, \ldots, r_i, \ldots, r_j, \ldots, r_{n_e}) = -\Psi(r_1, \ldots, r_j, \ldots, r_i, \ldots, r_{n_e}) \tag{1.3}$$

for $1 \leq i \leq j \leq n_e$. This many-body wavefunction describes the probabilistic properties of the system. For example, $\|\Psi(r_1, r_2, \ldots, r_{n_e})\|^2 dr_1 dr2 \ldots dr_{n_e}$ represents the probability of finding the first electron in a small volume in r_1, the second electron around a small volume around r_2, *etc.*

We apply the *Born-Oppenheimer* approximation and assume that the nuclei of the molecule are fixed in space at locations $\hat{r}_j, j = 1, 2, \ldots, n_u$ where n_u is the number of nuclei. The Hamiltonian $\mathcal{H}$ is then defined as:

$$\mathcal{H} = -\frac{1}{2}\sum_{1}^{n_e}\Delta_{r_i} - \sum_{j=1}^{n_u}\sum_{i=1}^{n_e}\frac{z_j}{||r_i - \hat{r}_j||} + \sum_{1\le i,j\le n_e}\frac{1}{\|r_i - r_j\|}, \tag{1.4}$$

where z_j is the charge of the jth nucleus and Δ_{r_i} is the Laplacian associated with the ith electron. Observe that the dimension of $\mathcal{H}$ is large even for trivial systems consisting of fewer than five electrons, meaning a direct solution to the eigenvalue problem (1.1) is computationally impractical. A direct calculation by Yang et al. [44] shows that equation (1.1) can be viewed as the first-order necessary conditions to the minimization problem

$$\min \int \Psi^* \mathcal{H} \Psi, \tag{1.5}$$

subject to both the orthogonality and antisymmetry constraints described in conditions (1.3).

Since 1950, two different approaches have been used to allow practical solutions to system (1.1) with great success. Originally, scientists followed the *Hartree-Fock* approach, as it described the chemical bonds of the molecule to reasonable accuracy. A more recent approach is based on *Density Functional Theory* (DFT). DFT is an exact approach if one knew the exact functional. It, however, relies fundamentally upon the *exchange-correlation energy* that attempts to model the complicated, many-body electron effects in a single term. Although this term is known to exist, the proof is non-constructive and no simple, exact expression is yet known.

1.1 A quick tour of Density Functional Theory

In their influential work, Hohenberg and Kohn [14] showed that, at ground state, the total energy of an electronic system can be described not by the many-body wavefunction, but instead solely by the electron charge density $\rho(r)$ defined as

$$\rho(r) = n_e \int \Psi^*(r, r_2, \ldots, r_{n_e}) \Psi(r, r_2, \ldots, r_{n_e}) dr_2 dr_3 \ldots dr_{n_e}. \tag{1.6}$$

Physically, this term describes the probability of finding an electron at a point in space near r.

Later, Kohn and Sham [21] proposed a practical formulation for the total energy of the system proven to exist earlier. This energy functional approximates the total energy of the system as a function of n_e single-particle wave functions ψ_i that *do not interact*. The charge density is then computed as

$$\rho(r) = \sum_{1}^{n_e} \psi_i^*(r) \psi_i(r). \tag{1.7}$$

Next, the Local Density Approximation (LDA) is used to approximate the exchange-correlation energy

$$E_{xc}(\rho) = \int \rho(r) \epsilon_{xc}[\rho(r)] dr, \tag{1.8}$$

where $\epsilon[\rho]$ computes the exchange-correlation energy per particle in a uniform gas of density ρ. The term, $\epsilon[\rho]$ approximates the electron-electron interactions and various approximations are used in practice [32]. The Kohn-Sham energy is then defined as a function of n_e orthogonal, single-particle wavefunctions

$$E_{KS}[\{\psi_j\}] = \sum_1^{n_e} \int \psi_j(-\frac{1}{2}\Delta\psi_j)dr + \int V_{ion}(r)\rho(r)dr + \frac{1}{2}\int\int \frac{\rho(r)\rho(r')}{||r-r'||}drdr' + E_{xc}(\rho). \tag{1.9}$$

The function $V_{\text{ion}} = \sum_1^{n_u} \frac{z_j}{\|r-\hat{r}_j\|}$ denotes the ionic potential induced by the nuclei. The optimization problem associated with equation (1.9) is given by

$$\begin{aligned} &\min && E_{KS}(\{\psi_j\}) \\ &\text{s.t.} && \int \psi_i^*\psi_j = \delta_{i,j} \text{ for } i = 1, 2, \ldots, n_e. \end{aligned} \tag{1.10}$$

The first-order necessary conditions for this minimization problem give the *Kohn-Sham equations*

$$\mathcal{H}_{KS}(\rho)\psi_i = \lambda_i\psi_i, \tag{1.11}$$

$$\int \psi_i^*\psi_j = \delta_{i,j}, \text{ for } i = 1, 2, \ldots, n_e. \tag{1.12}$$

The Kohn-Sham Hamiltonian $\mathcal{H}_{KS}$ is given by

$$\mathcal{H}_{KS} = -\frac{1}{2}\Delta + V_{ion}(r) + \rho(r) * \frac{1}{||r||} + V_{xc}(\rho), \tag{1.13}$$

where $*$ denotes the convolution operator and V_{xc} is the derivative of the right-side of equation (1.8) with respect to ρ. As ρ depends on the wavefunctions ψ_i, the Kohn-Sham equations can be viewed as a nonlinear eigenvalue problem.

1.1.1 Removing the orthogonality constraint

The orthogonality constraint in the optimization problem (1.10) can be removed through a modification of the energy function and charge density. Let S

denote the overlap matrix for nonorthogonal, single-particle wavefunctions $\{\psi_i\}$ defined by

$$S_{ij} = \int \psi_i^* \psi_j dr. \tag{1.14}$$

We extend the Kohn-Sham energy to non-orthogonal wavefunctions by letting

$$\rho(r) = \sum_j \sum_k \psi_j(r)^* (S^{-1})_{jk} \psi_k(r) \tag{1.15}$$

and rewrite the energy as

$$\begin{aligned} E(\{\psi_j\}) = &\sum_j \sum_k (S^{-1})_{jk} \int \psi_j(-\frac{1}{2}\Delta\psi_k) dr + \int V_{ion}(r)\rho(r) dr \\ &+ \frac{1}{2} \int \int \frac{\rho(r)\rho(r')}{||r - r'||} dr dr' + E_{xc}(\rho). \end{aligned} \tag{1.16}$$

A direct calculation shows that this form of the energy is *independent* of the basis chosen for the subspace spanned by $\{\psi_i\}$, i.e., given two sets of wavefunctions, $\{\psi_i\}$ and $\{\hat{\psi}_i\}$ such that

$$\text{span}\{\psi_i\} = \text{span}\{\hat{\psi}_i\} \tag{1.17}$$

it follows that

$$E(\{\psi_i\}) = E(\{\hat{\psi}_i\}) \tag{1.18}$$

We will refer to the formulations of the Kohn-Sham energy given by equations (1.9) and (1.16) as the *orthogonal Kohn-Sham energy* and the *nonorthogonal Kohn-Sham energy* respectively. The formulations of both the orthogonal and nonorthogonal Kohn-Sham energies will be analyzed in Chapter 2.

Although the extension of the Kohn-Sham energy to a nonorthogonal basis increases the complexity of the objective function, it allows the exploitation of *locality* in evaluating the function and computing a solution. When using

an orthogonal basis, the influence of a particular wavefunction covers the entire domain. For many molecular systems, however, it is known that the set of wavefunctions can be represented in a basis such that, each vector in the basis influences only a small region of space. Exploiting this property allows the design of new algorithms that scale better as the number of atoms in the system is increased. Specifically, each wavefunction can be assumed to have support only on a small, finite region of the domain. This locality, referred to as “nearsightedness" in [22] is not reflected in the standard electronic structure calculations as most diagonalization approaches explicitly generate orthogonal iterates.

In this work, locality is understood in the following terms; let $r \in \mathbb{R}^3$ be a point in space. The charge density $\rho(r)$ will only be influenced by nearby atoms in a region of some support radius.

For a complete overview of the material in this section see Goedecker [13], Kaxiras [17], Payne et al. [31].

2

Approaches to solving the Electronic Structure Problem

No matter the method, the first step in solving either the energy minimization problems in Chapter 1 or the related Kohn-Sham eigenvalue problems involves choosing a discretization. Many schemes exist, from standard wavelet [3], finite-difference [2, 4, 8] and finite-element [30, 39] techniques to the more physically realistic planewave [23, 24, 31] and Wannier function [25] approach to discretization. In any case, the discretization generates a finite basis in which the method is able to represent the wavefunctions, Laplacian operator, etc. Two standard discretizations, planewaves and finite-differences, will be described in the Sections 2.1.1 and 2.1.2 respectively.

After the discretization is chosen, one must solve either the Kohn-Sham eigenvalue problem or directly minimize the Kohn-Sham energy. General approaches to solving these problems will be described in Section 2.2 and Section 2.3.

2.1 Choosing a discretization

2.1.1 Planewaves

The central idea in using a plane wave discretization is to represent the wavefunction $\psi(r)$ as a linear combination of terms of the form $e^{ig_j^T r}$ where $r \in \mathbb{R}^3$

and each $g_j \in \mathbb{R}^3$ is a frequency vector. The main advantage of such a planewave discretization is that the necessary energy functions are straightforward to implement and their computation can take advantage of the Fast Fourier Transform (FFT).

For periodic systems, such as solids, the planewave basis is a natural choice. For non-periodic systems, planewaves can be used by embedding the structure in a fictitious supercell that is then periodically extended throughout the entire domain [31].

A wavefunction is represented by the Fourier series

$$\psi(r) = \sum_{j=-\infty}^{\infty} c_j e^{ig_j^T r} \tag{2.1}$$

where c_j denotes the Fourier coefficient

$$c_j = \int_{R/2}^{R/2} \psi(r) e^{-ig_j^T r} dr$$

and R is the period of the single-particle wavefunction.

As described, the Fourier series is infinite, so to compute the values numerically, one must truncate the series to allow only a finite number of terms. If care is not taken, this will lead to a large number of terms as $V_{ion}(r)$ contains a singularity at the position of each nuclei $\hat{r}_j$. These singularities require high frequency planewaves to approximate the wavefunction closely. However, inner electrons—electrons close to the nuclei—are not active in chemical reactions between atoms. The outer electrons, or *valence* electrons, are responsible for the majority of physical properties present in an atomic system. The wavefunctions of these valence electrons can be represented using relatively few planewaves.

The *pseudopotential* approximation [34, 35, 46], is a method of formalizing this observation. It attempts to model the inner electrons as part of the ionic core, while allowing the valence electrons to be treated separately. A detailed overview of different pseudopotentials and their properties is well beyond this dissertation, however, their use will allow

- V_{ion} to have no singularities, which in turn allows a small number of planewaves (or in the next section, a larger mesh-size) in representing a wavefunction;

- the reduction of the number of electrons in a prescribed system to only the valence electrons [44].

Using the pseudopotential then allows $\psi(r)$ to be represented by the finite sum

$$\psi(r) = \sum_{j=1}^{n_g} c_j e^{ig_j^T r}, \tag{2.2}$$

for some small constant n_g that depends only on the parameters of the system and the property of the system one wishes to study.

The Kohn-Sham energy is then written as

$$E_{KS}(X) = E_{kineticIonic}(X) + E_{Hartree}(X) + E_{xc}(X), \tag{2.3}$$

where

$$\begin{aligned} E_{kineticIonic} &= \text{trace}\left(X^*\left(-\frac{1}{2}L + V_{ion}\right)X\right), \\ E_{Hartree} &= \frac{1}{2}\rho^T L^\dagger \rho, \\ E_{xc} &= \rho^T \epsilon_{xc}[\rho], \\ \rho(X) &= \text{diag}\left[XX^*\right]. \end{aligned}$$

The discretized wavefunctions are naturally orthogonal given their construction.

The discretized Laplacian operator, L, can be written as

$$L = F^* D_g F, \tag{2.4}$$

where F is the Discrete Fourier Transform matrix and D_g is a diagonal matrix with elements $\|g_j\|^2$ on the diagonal. Due to the assumed periodic boundary conditions, the Laplacian is singular. We replace the inverse of L with a pseudo-inverse defined by

$$L^\dagger = F^* D_g^\dagger F, \tag{2.5}$$

where the diagonal elements of $D_g^\dagger$ are $\|g_j\|^{-2}$ when $g_j \neq 0$ and 0 otherwise.

2.1.2 Finite differences

We first consider a one-dimensional domain of radius r centered about the origin. This interval is discretized with a uniform mesh size of h. A function $\psi(r)$ is represented as a value $\psi(x_i)$ for each grid point $x_i \in [-r, r]$. We can expand this function to the points $x_{i+1} = x_i + h$ and $x_{i-1} = x_i - h$ via the Taylor series as follows:

$$\psi(x_{i+1}) = \psi(x_i) + \psi^1 h + \frac{1}{2}\psi^2(x_i)h^2 + \frac{1}{6}\psi^3(x_i)h^3 + \frac{1}{24}\psi^4(x_i)h^4 + \ldots \quad (2.6)$$

$$\psi(x_{i-1}) = \psi(x_i) - \psi^1 h + \frac{1}{2}\psi^2(x_i)h^2 - \frac{1}{6}\psi^3(x_i)h^3 + \frac{1}{24}\psi^4(x_i)h^4 + \ldots \quad (2.7)$$

where ψ^k denotes the kth derivative of the function ψ. Adding these two equations and rearranging the terms we obtain the well-known second-order, finite-difference approximation for the Laplacian

$$\psi^2(x_i) = \frac{1}{h^2}(\psi(x_{i-1}) - 2\psi(x_i) + \psi(x_{i+1})) + O(h^2). \quad (2.8)$$

By collecting the values of $\psi(x_i)$ into a vector, we obtain the well-known discrete Laplacian matrix, L,

$$\frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \end{pmatrix} \quad (2.9)$$

As described above, the finite-difference error is $O(h^2)$. This error can be decreased by using a higher-order approximation

$$\psi^2(x_i) = \sum_{j=-M}^{M} C_j \psi(x_i + jh) + O(h^{2(M+1)}). \quad (2.10)$$

where the sum bounds M and the coefficients C_n are defined in [11]. Unless otherwise specified, this work uses eighth-order finite-difference approximations, as is standard in other real-space methods, such as RSDFT described in Chapter 5. The use of a higher-order Laplacian will result in a slightly denser (but still sparse) Laplacian matrix; however, the number of grid points required for a given accuracy will be fewer.

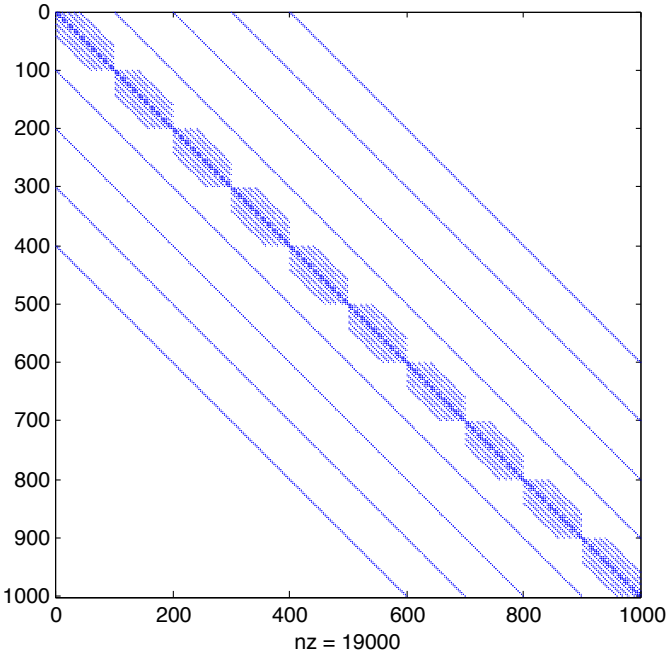


Figure 2.1: Eighth-order Laplacian sparsity structure

The finite-difference approximation can be extended to three dimensions by discretizing along each coordinate direction and then summing the one-dimensional case along each axis. Although, no longer tridiagonal, the Laplacian matrix is sparse and banded once a consistent ordering of the grid points

is determined. Figure 2.1 shows the sparsity pattern of the three-dimensional eighth-order Laplacian matrix. A negative aspect of using a finite-difference discretization is that this discretization is not *variational*; as the mesh size decreases the errors in approximating the true wavefunctions can be positive or negative.

After choosing an ordering of grid points, each wave function x can be collected into the matrix $X = [x_1, x_2, \ldots, x_{n_e}]$.

Combining the above, the Kohn-Sham energy is written

$$E(X) = E_{kineticIonic}(X) + E_{Hartree}(X) + E_{xc}(X), \tag{2.11}$$

where

$$\begin{aligned} E_{kineticIonic} &= \text{trace}\left(X^*X)^{-1}X^*\left(-\frac{1}{2}L + V_{ion}\right)X\right), \\ E_{Hartree} &= \frac{1}{2}\rho^T L^\dagger \rho, \\ E_{xc} &= \rho^T \epsilon_{xc}[\rho], \\ \rho(X) &= \text{diag}\left[X(X^*X)^{-1}X^*\right]. \end{aligned}$$

Observe that the matrix of wave functions, X, need not be orthogonal.

As stated above, L is the Laplacian matrix represented using finite-differences. The Hartree potential $L^\dagger\rho$ is computed iteratively via the solution to the Poisson problem

$$Lv = 4\pi\rho, \tag{2.12}$$

with periodic boundary condition, where ρ is evaluated at the current set of wavefunctions X.

A direct computation shows that the Kohn-Sham energy is independent of the basis: we have the relationship

$$E(X) = E(XG) \tag{2.13}$$

where $G \in \mathbb{R}^{n_e \times n_e}$ is an invertible matrix.(See equation (1.18). Similarly, the charge-density, ρ, is also independent of the basis chosen.

For completeness, the gradient of each term in the nonorthogonal Kohn-Sham energy (2.11) is given by

$$\begin{aligned}
\nabla E_{kineticIonic} &= -XS^{-1}X^*(L + V_{ion})XS^{-1} + (L + V_{ion})XS^{-1}, \\
\nabla E_{Hartree} &= \operatorname{Diag}(L^\dagger \rho)XS^{-1} - XS^{-1}X^* \operatorname{Diag}(L^\dagger \rho)XS^{-1}, \\
\nabla E_{xc} &= \operatorname{Diag}(V_{xc}XS^{-1} - X * (S^{-1} * X^* \operatorname{Diag}(V_{xc}XS^{-1})).
\end{aligned}$$

Observe that the gradient requires computing (or applying) the inverse of the overlap matrix, S. Implementation details regarding the computation of this term are provided in Section 5.1.

2.2 Solving the Kohn-Sham Eigenvalue Problem

After choosing a discretization, one standard approach to computing solutions to the Kohn-Sham eigenvalue problem defined by equations (1.12) is the Self-Consistent Field (SCF) iteration. The SCF algorithm is a fixed-point algorithm and is described in detail in Algorithm 2.1. In a basic implementation, each iteration requires a full orthogonalization of the Hamiltonian. Moreover, convergence to a fixed-point is not guaranteed. Yang et al. [43] showed that, for a simple model problem, the SCF iteration can generate a sequence containing

two limit points, neither of which is self-consistent. Heuristics, such as *charge mixing*, are used in practice, making the algorithm more reliable and rapidly convergent (Johnson [16], Kerker [19], Pulay [36], Raczkowski et al. [37]).

Algorithm 2.1 The SCF iteration

function SCF($X_0 \in \mathbb{R}^{n \times n_e}$)
    Let $X_k = X_0$
    repeat
        Compute the charge-density $\rho_k = \rho(X_k)$
        Form the Hamiltonian $H^{(k)} = H(\rho_k)$
        Solve the Kohn-Sham eigenvalue problem by computing an X_{k+1} such that

$$H^{(k)} X_{k+1} = X_{k+1} \Lambda$$

        where Λ contains the n_e smallest eigenvalues of $H^{(k)}$.
    until $\|\rho_k - \rho_{k-1}\|$ is sufficiently small
end function

In Gao and E [12], the authors propose a novel method for solving the Kohn-Sham eigenvalue problem. The algorithm replaces the expensive eigen-decomposition with a direct minimization of the Rayleigh quotient

$$R(X) = \operatorname{trace}\left((X^* X)^{-1} X^* H(\rho) X \right), \tag{2.14}$$

where $H(\rho)$ is the Kohn-Sham Hamiltonian evaluated at the previous charge density ρ. The authors then propose a modification of steepest descent and conjugate gradients that includes an explicit localization procedure that attempts to minimize $R(X)$ over sparse matrices X that contain a number of

non-zero entries that is linearly proportional to the number of wavefunctions in the molecule. By evaluating the function and gradient at only these sparse iterates, the algorithm obtains better scaling than if the matrix was fully dense. Section 2.4 provides an overview of this procedure.

2.3 Directly minimizing the Kohn-Sham Energy

A different approach is to minimize Kohn-Sham energy directly. These methods differ in whether they handle the orthogonality constraint explicitly—by choosing an orthogonal basis [26, 28], optimizing over the manifold of orthogonal matrices [9] or solving a constrained optimization problem [1, 42]— or implicitly—by minimizing a form of the non-orthogonal Kohn-Sham energy [5, 33]. We provide a brief overview of methods directly related to this thesis.

Pfrommer et al. [33] present an unconstrained optimization approach to minimizing the Kohn-Sham energy. In this work, the authors apply a standard nonlinear conjugate gradient method to minimize the nonorthogonal energy function (1.16). Additionally, they use a Taylor approximation to the matrix $(X^*X)^{-1}$ to speed up the function evaluations within the linesearch.

In Adhikari and Baer [1], the authors propose an augmented Lagrangian method that attempts to minimize an energy functional, E_f, as a function of the density matrix, $D = XX^*$, as opposed to the matrix of wave functions. The density matrix is related to the charge density via the relationship

$$\rho = \text{diag}(D). \tag{2.15}$$

The optimization problem is written as

$$\begin{aligned} \min \quad & E_f(D) \\ \text{s.t.} \quad & \operatorname{trace}(D) = n_e \\ & D^2 = D. \end{aligned} \tag{2.16}$$

For an overview of density-matrix based methods for solving the Kohn-Sham energy, see the references listed at the end of Chapter 1. Observe that in a finite-difference representation of the Kohn-Sham energy, the matrix D is a sparse $N \times N$ matrix where N is the number of grid points.

2.4 Exploiting locality to gain sparsity

When working with a finite-difference representation, locality, described in Section 1.1.1, can be directly exploited by allowing each wave function to have non-zero support on a constant number of gridpoints. When working with a finite-difference representation of the Kohn-Sham energy, the matrix of wavefunctions $X = [x_1, x_2, \ldots, x_{n_e}]$ is a sparse matrix with $O(n_e)$ nonzero elements as opposed to a dense matrix $N * n_e$, where N is the number of grid points in the entire domain. An example of a such a sparse matrix X in one dimension is depicted in Figure 2.2.

Choosing the support region, location and size for each wavefunction is molecule dependent; however, a rough starting point is to place a neighborhood about the midpoint of an atom and each of its nearest neighbors. When choosing the size of this region, a trade-off must be made between the computational efficiency of the method and the error allowed in the final solution

quality. Figure 2.3 demonstrates one particular choice of localization region for methane, CH_4.

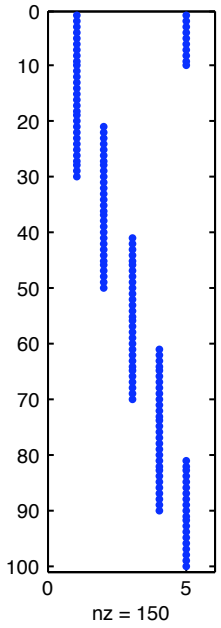


Figure 2.2: Sparsity structure of X: Support region size 30, $n = 100$, $n_e = 5$

An overview of notation for truncation

Let $M = [m_1, m_2, \ldots, m_k]$ be a rectangular matrix with dimensions $n \times k$ where $n > k$. Let S be $0/1$-rectangular matrix of the same dimensions as M. We define the *truncation* of the matrix M onto S as

$$(M)_T(i,j) = \begin{cases} M(i,j) & \text{if } S(i,j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.17}$$

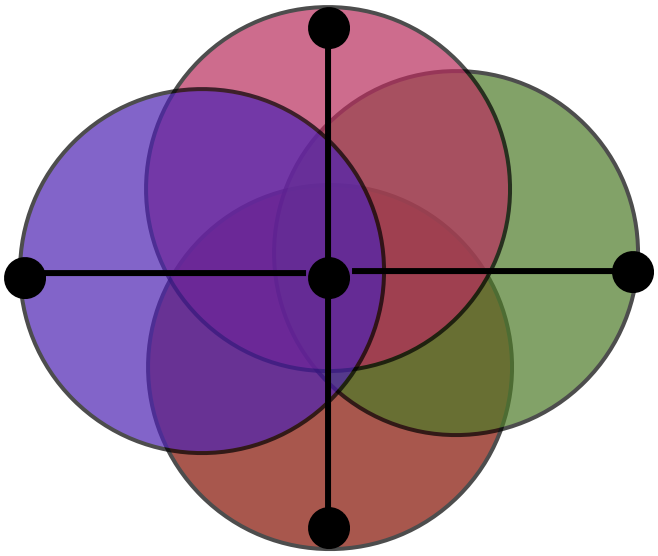


Figure 2.3: A localization region for CH_4. Each colored circle depicts a region of space where a particular wavefunction is allowed to have support.

where $M(i,j)$ and $S(i,j)$ denote the (i,j)-entry of each respective matrix. The *support region* of a column m_i is given by the corresponding column of the matrix S. If we assume each entry of the column m_i corresponds to a point $(x,y,z) \in \mathbb{R}^3$ (as when M comes from the finite difference discretization) the set of points $S_i = \{j : S(i,j) = 1\}$ can be viewed as a region of space known as the *localization* region of the column m_i. This duality between the sparsity structure of the matrix M and the localization region of each column will be critical when M is not a general matrix, but a matrix of *wavefunctions*.

Finally, it will be necessary to truncate all columns of the matrix M onto a

particular support region S_k. Specifically, we define

$$(M)_{T_k}(i,j) = \begin{cases} M(i,j) & \text{if } j \in S_k \\ 0 & \text{otherwise.} \end{cases} \tag{2.18}$$

2.4.1 A method for localization

Equation (2.13) describes a condition on X stating that the Kohn-Sham energy is invariant with respect to the basis in which X is represented. Gao and E [12] exploit this condition by computing the closest "localized" subspace so that the Kohn-Sham energy can be efficiently computed at a *sparse* matrix.

Given a dense matrix $X = [x_1, x_2, \ldots, x_{n_e}]$ corresponding to the finite difference discretizations of the wavefunctions ψ_i and a fixed support region S, it will be necessary to compute a sparse, localized matrix $\tilde{X}$ such that the spans of the two are equal (or nearly equal).

The *localization matrix*, G is then computed as the minimizer of the problem

$$||XG - (XG)_T||_F \tag{2.19}$$

under the condition that G is invertible. If the support region S is chosen appropriately, computational experiments show that

$$E(XG) \approx E((XG)_T). \tag{2.20}$$

The matrix G can be computed column-by-column in the following manner: First let $Y_i = X - (X)_{T_i}$. Letting $G = [g_1, g_2, \ldots, g_{n_e}]$, where g_i denotes the ith column of the matrix G, the problem reduces to finding solutions, g_i, to the n_e minimization problems

$$\min \; ||Y_i g_i||_2. \tag{2.21}$$

To ensure that the problem gives non-trivial solutions, we must also add an additional constraint on the columns of G. The form of this constraint leads to different solutions. For example, if it is specified that $g_i^T(1,1,\ldots,1) = 1$, then the problem reduces to the standard least-squares problem. Another choice would be to specify that each column of G satisfies $\|g_i\|_2 = 1$. In this case, the solution is given by a singular vector corresponding to the minimum singular value of Y_i. In this work, we use the first method in our localization procedure, as it is computationally more efficient.

It should be noted that the above solution replaces the global invertibility condition required for G with a column-wise constraint on the norm. Therefore, solutions of these problems do not necessarily guarantee the invertibility of the full matrix G nor that it is well-conditioned. For example, if the support regions for two columns are identical (or have much overlap), the corresponding two columns of G could be identical (or very similar). In such cases, localization may not be appropriate.

3

Localized Optimization

As larger molecular systems are analyzed computationally, the inherent scaling found in either evaluating the Kohn-Sham energy and gradient functions (in the case of optimization-based methods) or in solving the Kohn-Sham eigenvalue problem (in the case of SCF) quickly limits the size of systems able to be considered. Recently, much work has been published in constructing *linear-scaling* methods, i.e. methods that scale linearly in the number of electrons of the system (See Goedecker [13] for an overview of such methods). These linear-scaling methods fundamentally exploit the localizability of the wavefunctions under consideration. It should be clear that in the methods described in the following two chapters, no claims of linear-scaling are made; however, these methods are shown to be efficient experimentally. As opposed to evaluating the energy and gradient function a dense $n \times n_e$ matrix, both methods evaluate these functions at a sparse matrix with a number of non-zero entries that are linearly proportional to the number of wavefunctions of the system under consideration. This is a necessary first-step in the design of future linear-scaling methods using the techniques described in the subsequent chapters.

In this chapter, we consider the unconstrained, nonorthogonal Kohn-Sham energy defined in equation (2.11). The goal is to minimize this energy directly over the matrix of wavefunctions $X \in \mathbb{R}^{n \times n_e}$; however, to achieve better scaling, we wish to only evaluate the energy function and gradient at sparse, localized iterates. First, given a sparse iterate X_k, an unmodified optimization algorithm

cannot be expected to generate a new iterate X_{k+1} that is also sparse. For example, if each new iterate is computed by the steepest descent update

$$X_{k+1} = X_k - \alpha_k \nabla E(X_k), \tag{3.1}$$

where α is a scalar, the gradient $\nabla E(X_k)$ cannot be expected to have the same sparsity structure as the iterate X_k.

Simple experiments show that direct truncation onto a given support region—setting $X_{k+1} = ((X_{k+1})_T)$—can lead to local solutions far from the global optimum, especially if care is not taken in choosing a good initial set of wavefunctions[20]. Gao and E [12] demonstrate a method of integrating the previously described localization procedure directly into the optimization algorithm by localizing each subsequent iterate. However, as described in their paper, even though a given iterate may be sparse, the computed search direction may be dense, and the linesearch will be required to evaluate the energy function at non-sparse iterates so increases in scaling are hard to observe. The increased density of the search direction stems from the gradient of the nonorthogonal Kohn-Sham energy, given at the end of Section 2.1.2, which is itself dense. This chapter describes a new localization-based algorithm that overcomes this issue.

In the following sections, the optimization algorithm will not be specified, *i.e.* the choice of search direction P_k is only required to be a descent direction. Possible choices include computing the search direction based on a conjugate gradient type method, a quasi-Newton approach or a direction based on the full Newton direction. In our computational experiments reported in Chapter 5, we use a a search direction based on the (limited-memory) BFGS method and

compute a step length, α, that satisfies the weak-Wolfe conditions:

Armijo condition:

$$E(X_k + \alpha P_k) \leq E(X_k) + c_1 \alpha \nabla(E(X_k))^T P_k, \tag{3.2}$$

curvature condition:

$$\nabla(E(X_k + \alpha X_k))^T P_k \leq c_2 \nabla(E(X_k))^T P_k. \tag{3.3}$$

where $0 < c_1 < c_2 < 1$ are the Wolfe parameters.

For an overview of the optimization methods, see Nocedal and Wright [27].

3.1 Projected Localization

Instead of localizing each iterate, we will use localization to determine our search direction. The inspiration and intuition for this approach stems from the projected gradient methods described in [18]. For clarity, if a matrix has a known, exploitable sparsity structure, we will denote it with a tilde, as in $\tilde{M}$. A matrix without a tilde may or may not be be sparse.

Let $\tilde{X}^k$ be our current, sparse, iterate and P^k be the possibly dense search direction. We define the new, sparse, search direction as

$$\tilde{P} = \left((\tilde{X}^k + P^k)G^k\right)_T - \tilde{X}^k, \tag{3.4}$$

where

$$G^k = \arg\min \| \left(\tilde{X}^k + P^k\right) G - ((\tilde{X}^k + P^k)G)_T \|_F \tag{3.5}$$

as described in Section 2.4. For efficiency, we use the variation based upon a least-squares method. Visually, this procedure is described in Figure 3.1.

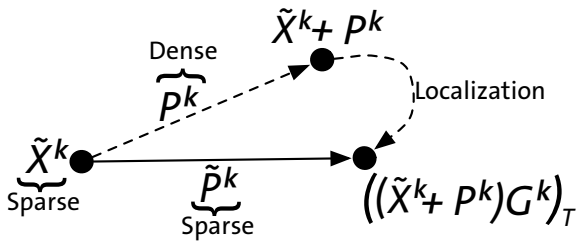


Figure 3.1: Procedure to generate new search direction

As the new search direction and the iterate share the same sparsity pattern, we ensure that the energy is only evaluated at sparse iterates. The complete algorithm is given in Algorithm 3.1.

The first question that needs exploring is whether this procedure maintains the descent condition of the search direction. Mathematically, this is equivalent to

$$\text{trace}\left(\tilde{P}^* \nabla E(X)\right) < 0. \tag{3.6}$$

Although theoretically this condition need not be maintained, practice shows that a loss of descent is rarely observed. To provide computational evidence that this procedure is robust in maintaining the descent property of the search direction, we apply this localization procedure to random iterates using the steepest descent direction and verify if descent is maintained. Specifically, we generate a random matrix of wave functions X, apply the localization procedure to $P = -\nabla(E(X))$ and compute the descent condition given in equation (3.6). Table 3.1 displays this experiment for two molecules C_2H_6 and $C_{12}H_{26}$.

Molecule	N	n_e	nnz in support	nnz/N	$nnz/(N * n_e)$	trials	%success
C_2H_6	46656	7	80532	.2466	1.726	280	100
$C_{12}H_{26}$	405224	37	1,069,140	2.63	.0713	21	100

Table 3.1: Experiment demonstrating that localization maintains descent. N is the number of grid points, n_e is the number of electrons. nnz is the number of non-zero elements in the specific support. %success is the percentage of trials that maintain descent.

Even given the evidence that localization maintains descent, care must be taken if this is not the case. One seemingly successful heuristic is to ignore the localization procedure completely and simply set the sparse search direction as the truncated search direction. If this truncation step is not required too often, the advantages of the localization procedure early in the optimization algorithms progress will not be lost in avoiding local, spurious minima. Additionally, experiments have shown the stopping the localization procedure after a constant number of iterations and simply truncating the search direction onto the provided support does not affect the solution quality and increases the rate of convergence of the algorithm. The intuitive reasoning behind this is that once the algorithm reaches an iterate sufficiently close to the final solution, the fill-in or off-support entries in the search direction P will be sufficiently small, leading to an ill-conditioned localization step. A typical value for this constant is $maxLocalizationIter = 20$.

The main workload of each iteration is divided between the necessary function and gradient evaluations required by the linesearch and the localization procedure required to compute the new search direction. When evaluating the

energy and gradient functions, the overlap matrix, S must be inverted as well as the necessary matrix multiplications must be performed. Theoretically, the matrix multiplication computations can be performed in time proportional to the number of non-zero entries of the given matrices; however, as described in Chapter 5, the actual implementation is not optimal. Moreover, care must be taken to maintain proper scaling when inverting the overlap matrix, S. Section 5.1 provides specific implementation details as well as references to computing this term efficiently. The localization procedure used requires solving n_e least-square problems each of size proportional to the number of elements *outside* of the support of each column. Given the expensive nature of this computation, future work must be performed to make the localization procedure more tractable.

Algorithm 3.1 KSLocOpt: Kohn-Sham Localized Optimization

function KSLocOpt(Initial iterate $\tilde{X}_0 \in \mathbb{R}^{n \times n_e}$ with support region S)
  Let $\tilde{X}_k = \tilde{X}_0$
  Compute the initial search direction, P_k
  repeat
    if current iteration $\leq maxLocalizationIter$ **then**
      Let $Y_k = \tilde{X}_k + P_k$
      Set $\tilde{Y}_k = (Y_k G)_T$ where G is computed as in equation (3.5)
      Set the new search direction as $\tilde{P}_k = \tilde{Y}_k - \tilde{X}_k$
    else
      $\tilde{P}_k = (P_k)_T$
    end if
    Choose α_k from a weak-Wolfe linesearch on $E(\tilde{X}_k + \alpha_k \tilde{P}_k)$ guaranteeing that conditions (3.2) and (3.3) hold.
    Set $\tilde{X}_{k+1} = \tilde{X}_k + \alpha_k \tilde{P}_k$
    Compute a new search direction P_{k+1}.
  until $\|\nabla(E(X_k))\|_F$ is sufficiently small
end function

4

A sparse Augmented Lagrangian Algorithm

In Chapter 3, an unconstrained optimization algorithm was described that minimized the nonorthogonal Kohn-Sham energy. Ignoring orthogonality entirely, however, can lead to scaling issues as different columns of the matrix X may have drastically different scales. This bad scaling can be seen by observing that $E(XG) = E(X)$ for all invertible G. In particular, by choosing G as the diagonal matrix $\operatorname{diag}(\varepsilon, 1, 1, \ldots, 1, \frac{1}{\varepsilon})$ the scaling of XG can be arbitrarily bad.

We will first describe the augmented Lagrangian equations specialized for the Kohn-Sham energy function. Subsequent sections will describe how this approach can then be extended to exploit locality to ensure the complex energy function is only evaluated at sparse iterates.

4.1 A dense augmented Lagrangian algorithm

The specific optimization problem we consider is given by

$$\begin{aligned} \min \quad & E(X) \\ \text{s.t.} \quad & X^*X - I = 0, \end{aligned} \tag{4.1}$$

where E is the nonorthogonal Kohn-Sham energy discretized using a finite-difference representation. Observe the constraint $X^*X - I = 0$ provides only

$\frac{n_e(n_e+1)}{2}$ independent constraints. Without loss of generality, we consider only the lower triangular portion of this matrix, denoted by $O_l(X) = \text{tril}(X^*X - I)$ (As a mnemonic, the O is for *orthogonality* and the 'l' is for lower triangular).

The augmented Lagrangian function is written in terms of the matrix X and the matrix of constraints $X^*X - I$ by observing that, for compatible matrices U and V

$$\text{trace}(U^*V) = \text{vec}(U)^* \text{vec}(V), \tag{4.2}$$

where $\text{vec}(M)$ is the vector consisting of the stacked columns of matrix M. This observation simplifies the necessary gradient computations by avoiding the otherwise necessary Kronecker products.

The augmented Lagrangian function is then given, in matrix form, by

$$\mathcal{L}_\mu^{KS}(X, \Lambda) = E(X) + \text{trace}(\Lambda^* O_l(X))) + \frac{1}{2\mu}\text{trace}(O_l(X)^* O_l(X)), \tag{4.3}$$

where Λ is a lower-triangular matrix approximating the Lagrange multipliers. The gradient is given by

$$\nabla\mathcal{L}_\mu^{KS}(X, \Lambda) = \nabla E(X) + X(V + V^*), \tag{4.4}$$

where

$$V = \text{tril}(\Lambda + \frac{1}{\mu}(X^*X - I)). \tag{4.5}$$

The augmented Lagrangian algorithm is given in Algorithm 4.1. Additional implementation details are provided in Chapter 5. The inner minimization problem found on line 4.1.3 of Algorithm 4.1 can be solved using a standard unconstrained minimization algorithm. In our computational experiments, LMBFGS is used (for a detailed description of the implementation of LMBFGS used as well as other implementation details, see Section 5.2).

Algorithm 4.1 The augmented Lagrangian method

function AUGMENTED LAGRANGIAN(X_0, μ_0, Λ_0)
    repeat
        $X_{k+1} = \arg\min \mathcal{L}_{\mu_k}(X, \Lambda_k)$
        Update μ_{k+1} according to Algorithm 5.2
        Update $\Lambda_{k+1} = \Lambda_k + \frac{1}{\mu_{k+1}} O_l(X_{k+1})$
    until Convergence criteria are met
end function

The main advantages of the augmented Lagrangian approach to minimizing the Kohn-Sham energy is that it directly handles the orthogonality constraint while avoiding computing the full Jacobian of the constraint matrix $X^*X - I$ as required by other constrained optimization algorithms. For completeness, this Jacobian is given by

$$\frac{\partial}{\partial X}(X^*X - I) = (I_{n^2} - P)(I_n \otimes X^*), \tag{4.6}$$

where I_n and I_{n^2} are the $n \times n$ and $n^2 \times n^2$ identity matrices respectively, P is the $n^2 \times n^2$ matrix that maps $\text{vec}(X) \rightarrow \text{vec}(X')$ and $\otimes$ is the standard Kronecker product. The matrix multiplication $X(V + V^*)$ is still required.

4.1.1 An aside on minimizing the augmented Lagrangian subproblem in the specific case of the Rayleigh quotient

Let $H \in \mathbb{R}^{n \times n}$ be a symmetric, positive-definite matrix. It is well-known that the solution of the minimization problem

$$\begin{aligned} \min_{X \in \mathbb{R}^{n \times k}} \quad & \text{trace}(X^*HX) \\ \text{s.t.} \quad & X^*X = I \end{aligned} \tag{4.7}$$

is given by

$$\sum_{1}^{k} \lambda_i, \tag{4.8}$$

where $\lambda_1 \leq \lambda_2 \ldots \lambda_k$ are the k smallest eigenvalue of H (Fan [10]).

As described above, the augmented Lagrangian method solves a sequence of unconstrained optimization problems of the form

$$\min_{X \in \mathbb{R}^{n \times n_e}} \quad \mathcal{L}_\mu(X, \Lambda), \tag{4.9}$$

where

$$\mathcal{L}_\mu(X, \Lambda) = \text{trace}(X^* H X) + \text{trace}(\Lambda^* O_l(X))) + \frac{1}{2\mu}\text{trace}(O_l(X)^* O_l(X)). \tag{4.10}$$

One approach to solving this inner minimization problem is Newton's method, where the search direction is computed as the solution to the system

$$\nabla^2 \mathcal{L}_\mu(X_k, \Lambda) p_k = -\nabla \mathcal{L}_\mu(X_k, \Lambda). \tag{4.11}$$

The Hessian $\nabla^2 \mathcal{L}_\mu(X_k, \Lambda)$ has dimension $(nk \times nk)$ so for large systems it will not be computationally feasible to form and store this matrix. However, a variation of Newton's method, the *truncated Newton method* solves system (4.11) approximately using the conjugate-gradient method [27, 38]. By using conjugate gradients, only *Hessian-vector* products are required and the full Hessian need never be formed. Let D be an arbitrary $n \times k$ matrix. We can compute the Hessian-vector product by first defining

$$\Phi(X, D) = \text{trace}(\nabla \mathcal{L}_\mu(X_k, \Lambda)^* D). \tag{4.12}$$

The gradient of the Hessian applied to the vector $\text{vec}D$ is then computed as

$$\frac{\partial \Phi(X, D)}{\partial X} = \nabla^2 \mathcal{L}_\mu(X_k, \Lambda)\ \text{vec}(D). \tag{4.13}$$

What is important to note is that this can be computed without forming the Hessian itself. In the case of equation (4.10), equation (4.13) is given by

$$\mathrm{mat}(\nabla^2 \mathcal{L}_\mu(X_k, \Lambda)\ \mathrm{vec}(D)) = F_1(X, D) + F_2(X, D) + F_3(X, D), \tag{4.14}$$

where

$$\begin{aligned} F_1(X, D) &= HD + DV, \\ F_2(X, D) &= \frac{1}{\mu} X \left(\mathrm{tril}(X^* D) + \mathrm{tril}(X^* D)^*\right), \\ F_3(X, D) &= \frac{1}{\mu} X \left(\mathrm{tril}(D^* X) + \mathrm{tril}(D^* X)^*\right), \end{aligned}$$

and V is defined in equation (4.5).

Initial experiments show that this approach will converge to a solution with the expected objective value requiring between 4-50 conjugate gradient steps to compute the search direction. Future work will explore both the application of this technique to solve sequences of trace minimization problems, for example, in solving the eigenvalue subproblem in the SCF iteration as well as using such techniques for directly minimizing the Kohn-Sham energy. A key component of this exploration will be proper preconditioning of the Hessian when only Hessian-vector products are available to reduce the required number of conjugate gradient iterations.

4.2 Introducing locality to the augmented Lagrangian

The augmented Lagrangian algorithm as explained above requires multiple dense matrix multiplications as the orthogonality condition is a global con-

straint affecting the entire domain under consideration. Locality is introduced into the optimization problem by assuming that wavefunctions only interact with other nearby wavefunctions over a limited volume of the domain. Orthogonality will then be enforced locally, only among groups of wavefunctions. As with the choice of localization region, the choice of clusters will be dependent on the molecule in consideration. By allowing some wavefunctions to belong to multiple clusters of wavefunctions, it is expected that proper scaling and conditioning will be maintained throughout the entire system. Such "mutual orthogonality" is difficult to apply in other methods and is often handled via similar tricks of choosing overlapping regions. See Wang et al. [41] for the use of such ideas when working with a plane-wave basis on a local scale.

To represent this new problem, let C_i denote the indices of the ith cluster of wavefunctions. The matrix X_{C_i} denotes the columns of X at indices C_i. If there is only one cluster $C_0 = \{1, 2, 3, \ldots, n_e\}$ then the problem reduces to the situation described above. Similarly, if the only specified cluster C_0 is empty, the formulation reduces to the unconstrained, non-orthogonal Kohn-Sham energy minimization problem described in Section 2.3.

After choosing the necessary clusters and support regions, the new optimization problem becomes

$$\begin{aligned} \min \quad & E(X_T) \\ \text{s.t.} \quad & X_{C_i}^* X_{C_i} - I = 0 \quad \text{for } i = 1, 2, \ldots k \end{aligned} \tag{4.15}$$

where k is the number of clusters and the corresponding augmented Lagrangian

is

$$\begin{aligned}\mathcal{L}_{\mu}^{KS}(X,\Lambda) =& E(X_T) + \sum_i \text{trace}(\Lambda_i^* O_l(X_{C_i})) \\ &+ \frac{1}{2\mu}\sum_i \text{trace}(O_l(X_{C_i})^* O_l(X_{C_i})).\end{aligned} \tag{4.16}$$

The nonorthogonal Kohn-Sham energy must be used in equation (4.15) as although individual subsets of columns of X will be orthogonal, the entire matrix will not be. Observe that the matrix X is always truncated onto its support region before any function, gradient or constraint evaluations. The complete algorithm is described in Algorithm 4.2 with implementation details given in Section 5.2.

The gradient computation is straightforward as above, although some care must be taken when accounting for wavefunctions that appear in multiple clusters.

The main computation time of each inner iteration is the energy and gradient function evaluation. As in the previous chapter, these evaluations require inverting the overlap matrix, S. The rest of the workload only requires sparse or small matrix-matrix multiplications when computing the energy and gradient functions or orthogonality constraint terms in the augmented Lagrangian respectively.

A key computational advantage of KSSPALOPT over KSLOCOPT, is that elements outside the specified support region need not be stored nor computed. The localization procedure used in KSLOCOPT requires accessing elements outside of the specified support when computing the required localization matrix.

Algorithm 4.2 KSSPALOPT: Kohn-Sham Sparse augmented Lagrangian method

1: **function** KSSPALOPT(X_0,Support S, Clusters C_i,μ_0, Λ_0)
2: **repeat**
3: $X_{k+1} = \arg\min \mathcal{L}_{\mu_k}(X_T, \Lambda_k)$
4: Update μ_{k+1} according to Algorithm 5.2
5: Update $\Lambda_{k+1} = \Lambda_k + \frac{1}{\mu_{k+1}} O_l(X_{k+1})$
6: **until** Convergence criteria are met
7: **end function**

5

Implementation Details and Numerical Experiments

All algorithms are implemented in MATLAB and run using MATLAB version R2010a. The implementations focus on algorithm experimentation as opposed to speed. Methods for computing the Hartree potential and exchange-correlation energy terms, as well as the computing and loading of molecule dependent data such as domain size into the system are borrowed from the MATLAB package RSDFT [7, 47, 48]. RSDFT is a real-space, finite-difference based code that solves the Kohn-Sham eigenvalue problem using a variation of SCF. In addition to code reuse, using RSDFT as a basis for energy calculations allows the solutions generated by our new algorithms to be compared directly to a known solution.

In all implementations, the sparsity structure is not exploited explicitly; instead the MATLAB sparse matrix functionality is used. Future implementations will allow for better performance, as the sparsity structure on the wavefunctions X can be exploited directly without the overhead needed for representing general sparse matrices. Moreover, both implementations only support basic parallelization as provided by MATLAB.

The experiments described in Sections 5.5, 5.6 and 5.8 were conducted on a Macbook Pro with a 2.66Ghz Intel 513 Core i7 processor and 4GB of RAM. The larger experiments described in Sections 5.7 and 5.9 were run on a Intel

Xeon 659 node with eight 2.33GHz cpus and 32 GB of RAM.

For reference, the algorithm names are given in Table: 5.1.

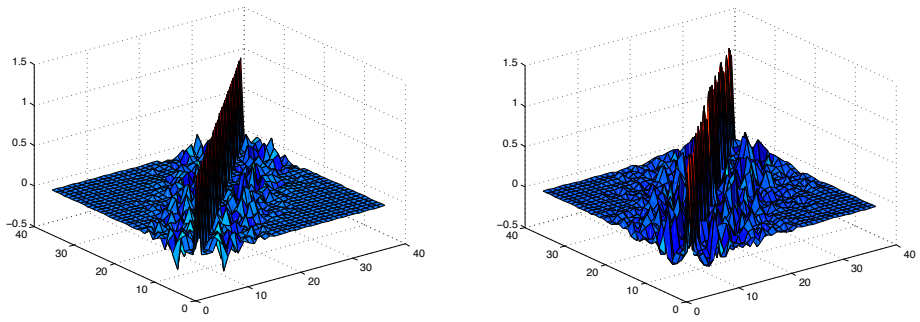
Algorithm	Meaning	Reference
RSDFT	Real-Space Density Functional Theory	[7, 47, 48]
KSLOCOPT	Kohn-Sham Localized Optimization	Section 3.1
KSALOPT	Kohn-Sham Augmented Lagrangian Optimization	Section 4.1
KSSPALOPT	Kohn-Sham Sparse Augmented Lagrangian Optimization	Section 4.2

Table 5.1: Algorithm names

5.1 Some comments regarding the overlap matrix $S = X^*X$ and its inverse

In computing both the nonorthogonal Kohn-Sham energy and its gradient, a key step is both the computation and factorization of the matrix $S = X^*X$. In general, computing S^{-1} would require $O(n_e^3)$ operations. However, given the supplied structure on the matrix X, the matrix S is sparse and diagonally dominant. Although not true in general, the inverse of this matrix is observed computationally to also be diagonally dominant. Figures 5.1(a) and 5.1(b) shows surface plots depicting the magnitude of each element of the matrices S and S^{-1} for a specific molecule.

Given this structure, techniques exist that can compute the inverse of the overlap matrix with scaling linear in n_e. See Jansík et al. [15] for one such approach. In this work, different approaches using the incomplete Cholesky factorization (see Challacombe [6]) were implemented and profiled; however,



(a) S (b) S^{-1}

Figure 5.1: Surface plot of $(i, j, S(i, j))$ and $(i, j, S^{-1}(i, j))$ for $C_{12}H_{26}$.

given the relatively small size of the matrix S, the straightforward use of the MATLAB backslash operator, '\' , proved to be the fastest means to compute the required inverse. In future work, as these ideas are extended to larger systems, this decision will need to be revisited.

5.2 Details involving the optimization algorithm implementations

In both Chapter 3 and Chapter 4, the algorithms presented require solving an unconstrained minimization problem. A full Newton algorithm or quasi-Newton method is not practical for this problem as the required Hessian or quasi-Newton matrices are both expensive to compute and too large to store. Therefore, either the Limited-Memory BFGS (LMBFGS) method or a conjugate gradient-based method seem appropriate. Given that most CG methods require

the stricter, strong-Wolfe linesearch, whereas LMBFGS requires only a weak-Wolfe linesearch, the latter is used. For a complete overview of this material see Nocedal and Wright [27].

The LMBFGS details are as follows:

Algorithm: LMBFGS

Linesearch: generates an iterate satisfying the weak Wolfe conditions. Uses bisection to compute the step length as fewer function evaluations are required. The chosen linesearch parameters are $c_1 = .01$, $c_2 = .99$. The choice seeks to minimize the number of function evaluations required by each linesearch. The code used is a slightly modified version of the code released as part of the HANSO package (Overton [29]).

Algorithm specific options:

History length: The LMBFGS update is computed using the three pairs

$$\{(s_{k-2}, y_{k-2}), (s_{k-1}, y_{k-1}), (s_k, y_k)\}.$$

Sigma update formula: When computing the LMBFGS search direction, an initial approximation to the inverse Hessian matrix is required. We follow the standard procedure of choosing a scaled identity matrix σI where σ is computed as

$$\sigma = \frac{1}{2}\left(\frac{s_k' y_k}{y_k' y_k} + \frac{s_k' s_k}{s_k' y_k}\right)$$

and $s_k = X_{k+1} - X_k$ and $y_k = \nabla E(X_{k+1}) - \nabla E(X_k)$.

In addition to the options above, the implementation uses a heuristic that replaces σ_k with $\nu\sigma_k$, attempting to generate future search directions that lead

to the linesearch taking a step of length one. The damping parameter ν is initially set to 1 and updated according to Algorithm 5.1.

Algorithm 5.1 Heuristic for updating the damping parameter ν

if $\alpha_k < 1$ **then**

$\nu = \max\left(10^{-5}, \frac{1}{2}\nu\right)$

else

$\nu = \min(1, 2 * \nu)$

end if

Both the heuristic for updating ν and the update formula for σ were suggested by Waechter [40].

The augmented Lagrangian based methods details are as follows:

Algorithms: KSALOPTand KSSPALOPT

Initial Lagrange multipliers: All multipliers are initialized to 0

Initial penalty parameter: $\mu_0 = 1$

Penalty parameter update: The penalty parameter μ is updated using the heuristic described in Algorithm 5.2.

Termination Conditions for inner, optimization problem: LMBFGS is run for a maximum of 50 iterations. LMBFGS will terminate early if it generates an iterate satisfying

$$\|\nabla(E(X_k))\|_F \leq \max(10^{-4}, 10^{-i})$$

where i is the current outer iteration number.

Termination Conditions for outer loop: Each method is run for a specified maximum number of outer iterations, typically 10. The outer loop exits early if the inner optimization method generates a solution, X^*, satisfying

$$\|\nabla(E(X^*))\|_F \leq 10^{-4}$$

and with maximum infeasibility less than 10^{-4}, i.e.,

$$\max|\mathcal{O}_l(X^*)| \leq 10^{-4}$$

where $\mathcal{O}_l(X)$ is defined as in Section 4.1.

Algorithm 5.2 Heuristic for updating the penalty parameter μ

if $\max|O_l(X_{k+1})| \geq .75\max|O_l(X_k)|$ **then**

$\mu_{k+1} = \max\left(10^{-5}, \frac{1}{2}\mu_k\right)$

else

$\mu_{k+1} = \mu_k$

end if

5.3 Choice of the initial set of wavefunctions

It is well known that the choice of initial iterate is crucial to fast convergence for most optimization algorithms. We compute the initial iterate based on physical considerations as follows: For a given wavefunction corresponding to an atom pair (a column of the matrix X), the modified Gaussian

$$e^{-\left((x-p_x)^2+(y-p_y)^2+(z-p_z)^2\right)}$$

is centered about the midpoint $p = (p_x, p_y, p_z)$.

In the case of the projected Localization algorithm, the algorithm is started at the localized iterate $Y_0 = (X_0 G)_T$ where G is the computed localization matrix. For the sparse augmented Lagrangian method, the initial set of wavefunctions, X_0, is first orthogonalized via the (thin) QR decomposition, $X_0 = Q_0 R_0$, and the initial iterate is set as $Y_0 = (Q_0)_T$.

Another, more expensive option is to run the SCF iteration for exactly one iteration. This approach may be useful when the collection of midpoints may not be easily determined due to lack of *a priori* knowledge of the system's structure.

5.4 Experiment Overview

The following sections compare the solutions generated by each optimization algorithms described in this dissertation with that of RSDFT . The mesh size, h, and domain size are determined by RSDFT and based on the molecule under consideration. The goal is to compute a sparse solution with approximately the same energy and charge density at the solution. The trade-off between sparsity (and scaling) and solution quality requires careful consideration and is dependent on the system under consideration. A solution computed by RSDFT will be denoted X^*, with energy E^* and charge-density ρ^*. RSDFT uses its own starting point. In figures where multiple runs of KSLOCOPT and KSSPALOPT are compared, the energy computed by RSDFT is plotted repeatedly to better depict the variation and quality of the optimization based methods.

The figures depicting each molecule are in the public domain and found on Wikipedia.

5.5 Methane(CH_4)

Methane, shown in Figure 5.2, is the simplest natural gas, containing only 4 valence electrons in its outer shell.

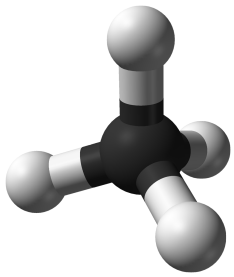


Figure 5.2: Methane(CH_4)

The domain used is a three-dimensional box with a radius of 8.25 atomic units (a.u.) that contains 34 grid points in each dimension (the mesh size is $h = \frac{1}{2}$ a.u.). The eighth-order Laplacian has dimension 39304×39304 with only 913,240 nonzero entries ($\approx$ .0006% sparsity). RSDFT computes an orthogonal solution X^* with Kohn-Sham energy -16.034 Rydbergs (Ry).

5.5.1 Choosing the localization region for KSLocOpt

The localization region is chosen as in Figure 5.3. For ease of computation, the region is chosen as a box, as opposed to a sphere, about the midpoint of each adjacent atom pair.

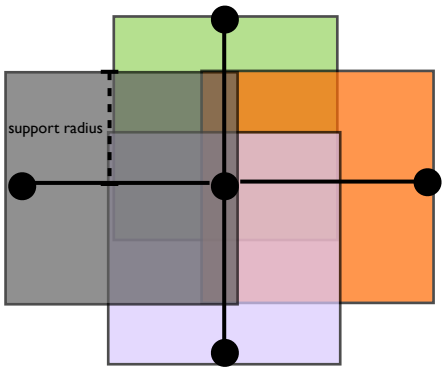


Figure 5.3: Support for KSLOCOPT for CH_4)

5.5.2 Choosing the localization region for KSSpALOpt

When using the sparse augmented Lagrangian, the support for each wavefunction is chosen as the region about the center Carbon atom as shown in Figure 5.4. Experiments have shown that for wavefunctions that are constrained to be orthogonal (in this case all of them), the support regions should be chosen to be identical to each other in order to meet this orthogonality constraint.

5.5.3 Computational Experiments

Given the relatively small size of matrices required for these computations a detailed analysis of both algorithms is presented for this molecule.

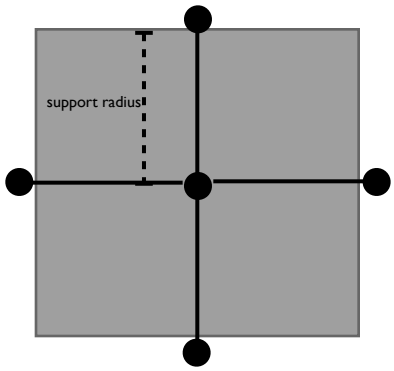


Figure 5.4: Support for KSSPALOPT for CH_4)

5.5.4 Avoiding spurious local minima

It is well-known that simple truncation of the wavefunctions onto their respective support regions leads to multiple local-minima [12, 20]. The first experiment exhibits the ability of both algorithms to compute solutions with similar Kohn-Sham energies. Each algorithm is run from the same random initial matrix of wavefunctions. The localization region is chosen to cover $\approx$ 45% of the domain (the support radius is 6.5 a.u.). It should be noted that comparing both methods with the same localization radius is not exactly fair. For a given radius, both methods will have slightly different localization regions and a different number and pattern of non-zero elements in the their respective support matrices. More specifically, given that the regions are staggered when choosing the support for KSLOCOPT, more of the domain is covered with at least one grid point.

Figure 5.5 compares the final energies computed by each method, with the energy computed by RSDFT as well as the energy at the solution from RSDFT *localized* onto the support region described in Section 5.5.1 (denoted by $E(X^*)$ and $E((X^*G)_T$ respectively). These values should provide an approximate lower bound or *goal* energy value for both KSSPALOPT and KSLOCOPT. (The bound is approximate, as RSDFT is an iterative algorithm with its own termination conditions).

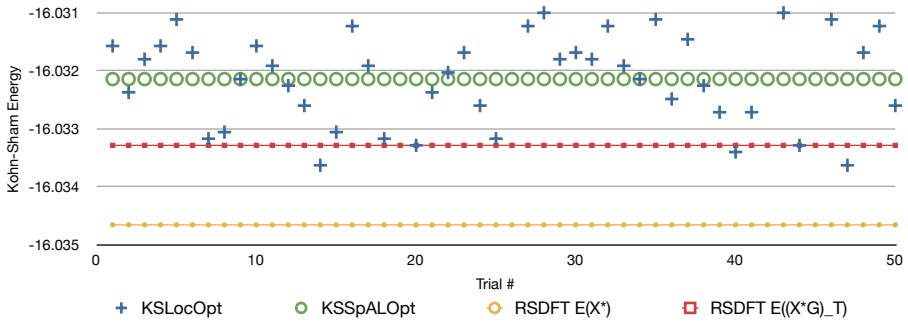


Figure 5.5: Methane (CH_4): Energy at the solution starting from 50 random initial iterates. The energy at the solution computed by RSDFT as well as the energy of the localized solution from RSDFT are repeated for convenience.

This experiment shows a common trend in comparing both new methods. KSLOCOPT *can* outperform KSSPALOPT (as well as RSDFT) for a given support region; however, the sparse augmented Lagrangian method tends to be more consistent in solution quality.

The central tenet of density functional theory is that at the solution, the properties of a molecule can be derived from the charge density. We compare the final charge density, ρ^*, computed by RSDFT with the final charge density

computed by each method in Figure 5.6. In addition to the sparse augmented Lagrangian, we include the solution computed by augmented Lagrangian with complete support. Figure 5.6 demonstrates that the final charge density computed by each method is comparable to the charge density computed by RSDFT.

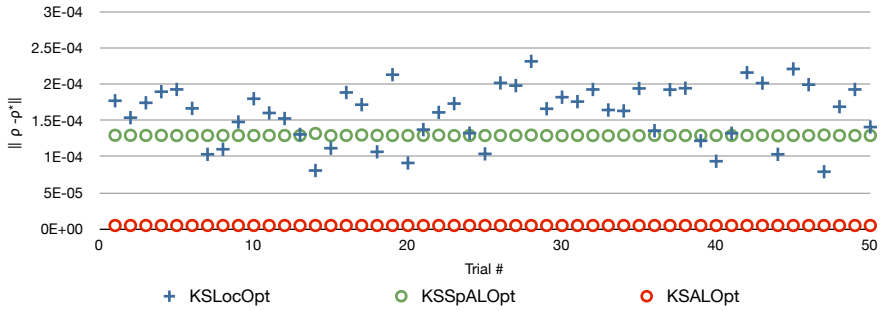


Figure 5.6: Methane (CH_4): $\|\rho - \rho^*\|_2$ at the final solution (log scale)

5.5.5 The effect of the support radius on solution quality

The next experiment compares the effect of the size of the support region on the quality of the computed solution. Each method is started from the same iterate, while increasing the support region until the support covers the entire domain. Figure 5.7 depicts the results for a particular run of each method. The number of iterations shown for KSSPALOPT is the total number of inner iterations summed over all outer iterations required to meet the given termination conditions.

For smaller support regions, KSLOCOPT computes a solution with a smaller energy than KSSPALOPT ; however, Table 5.2 shows that the projected Local-

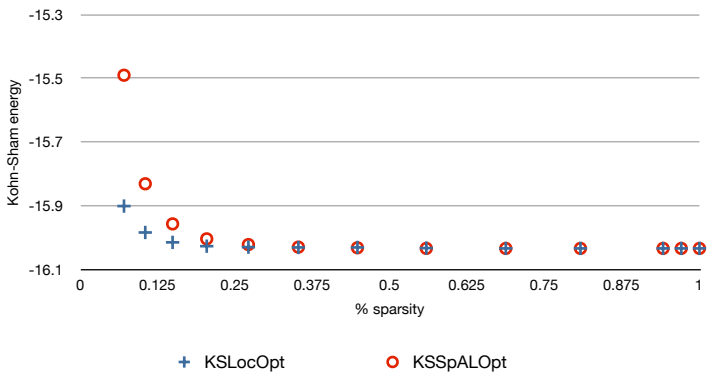


Figure 5.7: Methane (CH_4): Energy at the solution vs. the size of the support

ization method has a higher variability in the number of iterations required for convergence. Again, it should be noted that comparing the number of iterations is not completely fair as each method has slightly different termination conditions, as KSSPALOPT must meet both gradient norm and constraint tolerances. However, given that KSLOCOPT requires an additional localization step in each iteration, this seems to be the fairest metric for comparison.

5.5.6 Rate of convergence from a good starting point

We now run each optimization algorithm from the initial iterate described in Section 5.3. The support radius is chosen as 6.5 a.u.. The results are shown in Figure 5.8. The convergence to a matrix of wave functions with energy equal to that computed by RSDFT is more rapid with both methods reaching a stationary point in fewer iterations than from a random starting point.

Support %	KSLocOpt iters	KSSpALOpt iters
0.069815	74	102
0.10421	159	102
0.14838	337	117
0.20354	384	118
0.27091	272	119
0.35172	105	118
0.44718	125	116
0.55852	184	116
0.68695	202	120
0.80765	183	113
0.94118	145	120
0.97059	86	120

Table 5.2: Methane (CH_4): Size of support vs. the number of iterations to solution

5.5.7 Timing experiments from a good starting point

Comparing timings is subtle, given variations in machine workload, *etc.* However, it is still beneficial to compare the cost per iteration of both optimization-based approaches with RSDFT. It should be noted, that neither optimization algorithm described in this thesis nor RSDFT are implemented for peak performance, nor do they take advantage of parallelization.

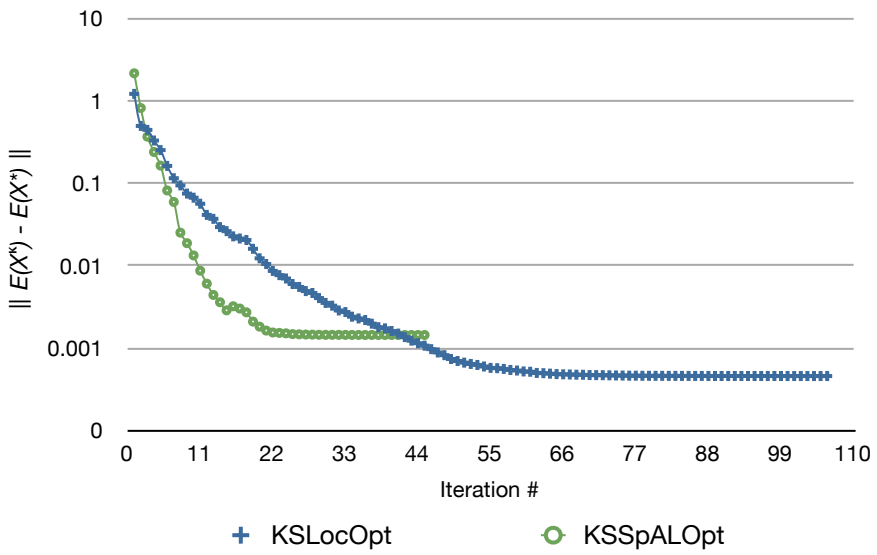


Figure 5.8: Methane (CH_4): iterations vs. $(E^k - E^*)$ from a good starting point (log scale)

	RSDFT	KSSPALOPT	KSLOCOPT
iters	11	45	107
mean(secs)	1.79	.56	.82
min(secs)	1.45	.47	.68
max(secs)	4.24	.67	1.78

Table 5.3: Timing Experiments for CH_4. Time is seconds per inner iteration

5.6 Ethane (C_2H_6)

Ethane (depicted in Figure 5.9) is a colorless, orderless gas and a byproduct of petroleum refinement and contains 7 valence electrons. RSDFT computes an

orthogonal solution, X^*, with Kohn-Sham energy $-29.4534Ry$. The domain is chosen as a box of radius 8.75 a.u. using a mesh size $h = \frac{1}{2}$ a.u..

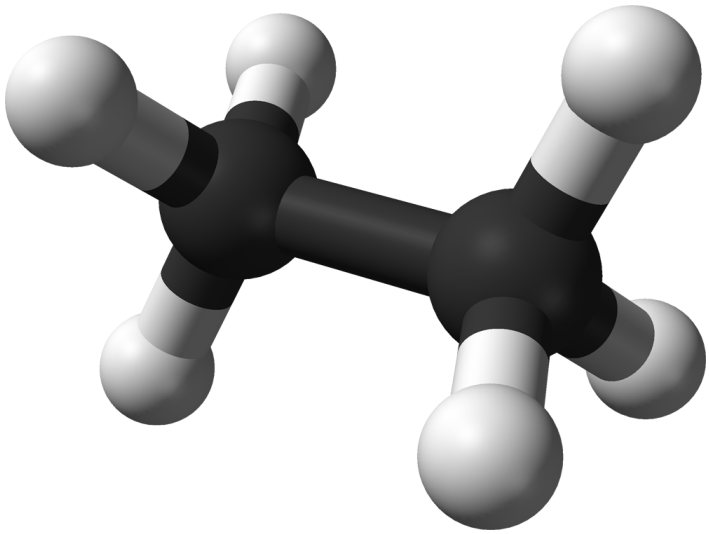


Figure 5.9: Ethane (C_2H_6)

5.6.1 Choosing the localization region for KSLocOpt and KSSpALOpt

In the case of KSOPTLOC, the support region is chosen as in methane: a box is placed around the midpoint of each atom-atom pair.

With larger molecules, the difference between the necessary support regions required by KSLOCOPT and KSSPALOPT become more pronounced. Ethane consists of a C-C bond and 6 C-H corresponding to the 7 wavefunctions. The support for the columns of X corresponding to the C-H bonds is chosen as a box about the nearest carbon atom. The support for the final C-C bond is chosen as the union of the support. Figure 5.10 demonstrates the structure on the matrix

of wavefunctions X. Observe that columns one to three (corresponding to the first set of C-H bonds) share the same support as do columns five to seven (corresponding to the second set of C-H bonds). Column 4, corresponding to the single C-C bond, has a support region that that is the union of both of these clusters.

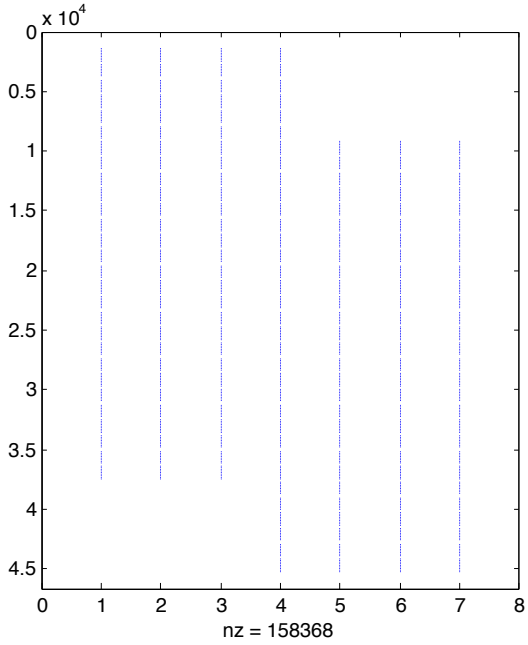


Figure 5.10: Ethane (C_2H_6): Wavefunction support matrix

5.6.2 The effect of the support radius on the solution quality

Figure 5.11 compares the energy at the computed solution as a function of increasing support radius (decreasing sparsity). As with the methane experiment, for smaller support regions, KSLocOpt outperforms KSSpALOpt. However,

when the support is chosen so that the matrix of wavefunctions X has sparsity of about 20%, the computed energy is nearly identical for both methods.

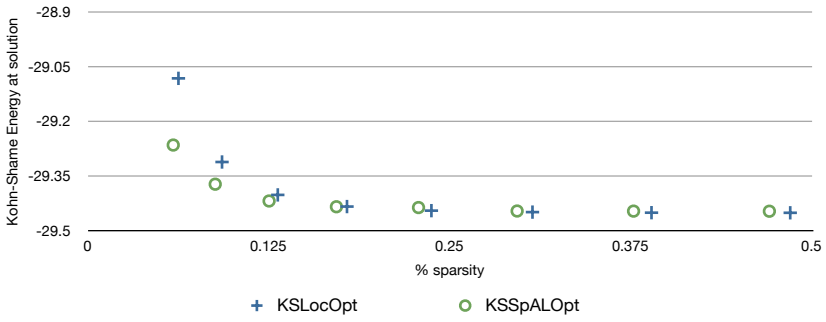


Figure 5.11: Ethane (C_2H_6): Size of the support versus energy at the solution

5.6.3 Rate of convergence from a good starting point

The support is chosen so that the matrix of wavefunctions has a sparsity of approximately 25% for each method. The difference between the energy at each iterate and the known energy E^* from RSDFT is shown in Figure 5.12. KSSPALOPT converges more quickly to a solution with correct energy in both time and the number of iterations required.

5.7 Dodecane ($C_{12}H_{26}$)

Dodecane ($C_{12}H_{26}$), shown in Figure 5.13, is the final molecule of the form C_nH_{2n+2} we will consider in this dissertation. It consists of 74 electrons corresponding to 37 wavefunctions. RSDFT computes an orthogonal solution

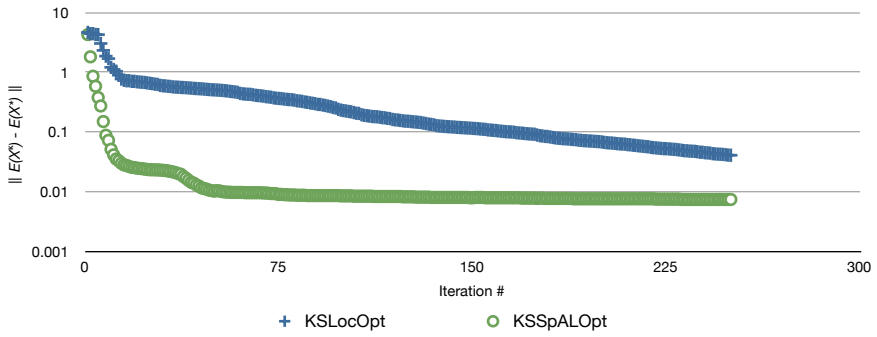


Figure 5.12: Ethane (C_2H_6): $(E^k - E^*)$ from a good starting point (log scale)

with energy -167.389 Ry. The domain is a box of radius 18.25 a.u. using mesh size of $h = \frac{1}{2}$ a.u.. The Laplacian has dimensions $405,224 \times 405,224$ with a sparsity of less than 6×10^{-5}%.

The localization regions are chosen in the same manner as for both methane and ethane. For KSLocOpt, the localization region is placed as a box about each atom-atom pair. In the case of KSSpALOpt, the regions are constructed as in Section 5.6.1, repeated about each of the 12 carbon atoms. The wavefunction corresponding to a C-C pair is constrained to be orthogonal to both the cluster to its left and the one to its right. For both methods, the support regions are chosen to have radius 7.5 a.u. giving a sparsity of approximately 11%. This support region was chosen so that the energy computed at the solution equals the energy computed by SCF.

Both KSLocOpt and KSSpALOpt are run from an initial iterate of wavefunctions described in Section 5.3. The convergence of the energy to the energy computed by RSDFT is shown in Figure 5.14. Figure 5.15 shows the isosurface

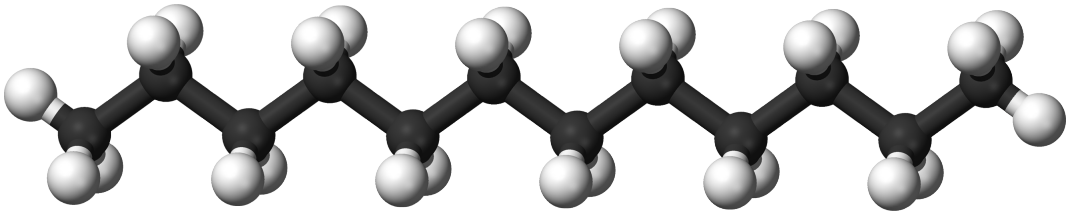


Figure 5.13: Dodecane ($C_{12}H_{26}$)

of the charge density at the solution computed by KSSPALOPT(the surface is computed at a value determined by Matlab). Table 5.4 details statistics as to the timing of each inner iteration. Observe that as the number of wavefunctions increase, the difference in timing between KSLOCOPT and KSSPALOPT increase. The localization procedure is an expensive operation.

Method	KSSPALOPT	KSLOCOPT	RSDFT
iter	118	500	14
Energy	−167.386	−167.382	−167.389
mean(secs)	24.7	32.3	72.69
min (secs)	22.1	24.7	60.18
max (secs)	25.6	39.0	206.71

Table 5.4: Timing experiments for Dodecane ($C_{12}H_{26}$)

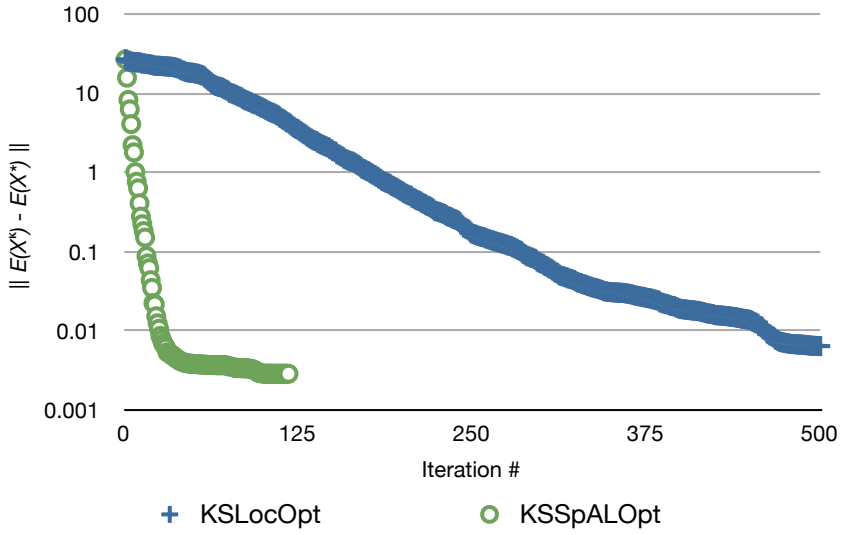


Figure 5.14: Dodecane ($C_{12}H_{26}$): $(E^k - E^*)$ (log scale)

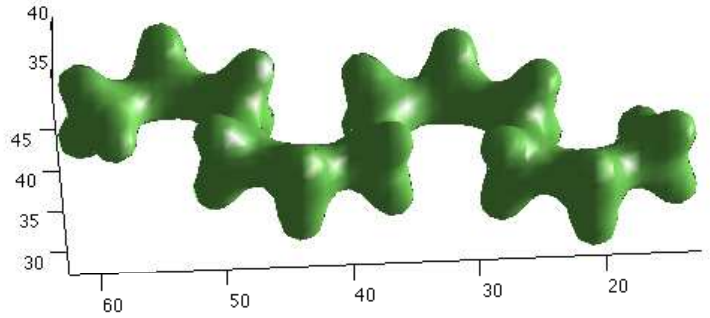


Figure 5.15: The computed charge density of the Dodecane ($C_{12}H_{26}$) molecule

5.8 Si_2H_4

The following experiment demonstrates that KSSPALOPT can be applied to molecules with double bonds where multiple wavefunctions will share support.

The molecule Si_2H_4 consists of four single Si-H bonds and one double S-S bond for a total of six wavefunctions. Due to this double bond, KSLOCOPT is not directly applicable, as two wavefunctions will have identical support. In future work, we will extend the localization procedure to handle such situations by introducing a local orthogonalization step [45]. However, the KSSPALOPT method can still be applied with the appropriate choice of clusters and support regions (see Section 4.2). Three clusters are chosen as to include the different atom pairs follows:

$$C_1 = \{\mathrm{Si}^1\text{-H}(1),\ \mathrm{Si}^1\text{-H}(2),\ \mathrm{Si}^1\text{-Si}^2(3)\}$$

$$C_2 = \{\mathrm{Si}^1\text{-Si}^2(3),\ \mathrm{Si}^2\text{-Si}^1(4)\}$$

$$C_3 = \{\mathrm{Si}^2\text{-H}(5),\ \mathrm{Si}^2\text{-H}(6),\ \mathrm{Si}^1\text{-Si}^2(4)\}$$

where Si^1 and Si^2 denote two different silicon atoms. The number in parenthesis denotes the column of the matrix X corresponding to the given atom-atom pair. Orthogonality between wavefunctions will only be enforced within each cluster. The support for the matrix of wavefunctions (shown in Figure 5.16) is chosen as follows:

- Columns 1-2 and 5-6 corresponding to the Si-H pairs have support in a box about the nearest Si atom;
- Column 3 corresponding to the first wavefunction of the double Si-Si bond has support about the midpoint of the atom pair unioned with the support of columns 1 and 2;
- Column 4 corresponding to the second wavefunction of the double Si-Si

bond has support that is the union of the region about the midpoint of the atom pair and the support of columns 5 and 6.

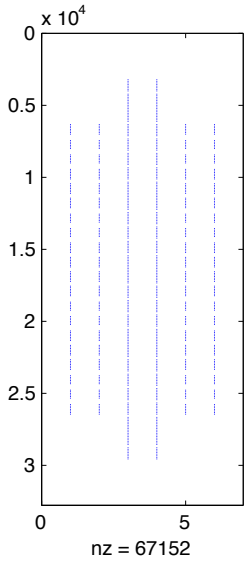


Figure 5.16: (Si_2H_4): Wavefunction support matrix

RSDFT computes an orthogonal solution with Kohn-Sham energy -12.316 Ry. The domain consists of a box of radius 9.3 a.u. using a mesh-size of $h = \frac{3}{5}$ a.u..

5.8.1 Rate of convergence from a good starting point

The starting point is chosen as described in Section 5.3 for the case of the sparse augmented Lagrangian method. Figure 5.17 shows the convergence behavior of both the energy and the charge density as a function of the iteration. As is often observed when running KSSPALOPT, the initial convergence is rapid,

with progress slowing down as the solution is approached. Figure 5.18 shows the isosurface of the charge density, ρ, at the solution computed by KSSPALOPT.

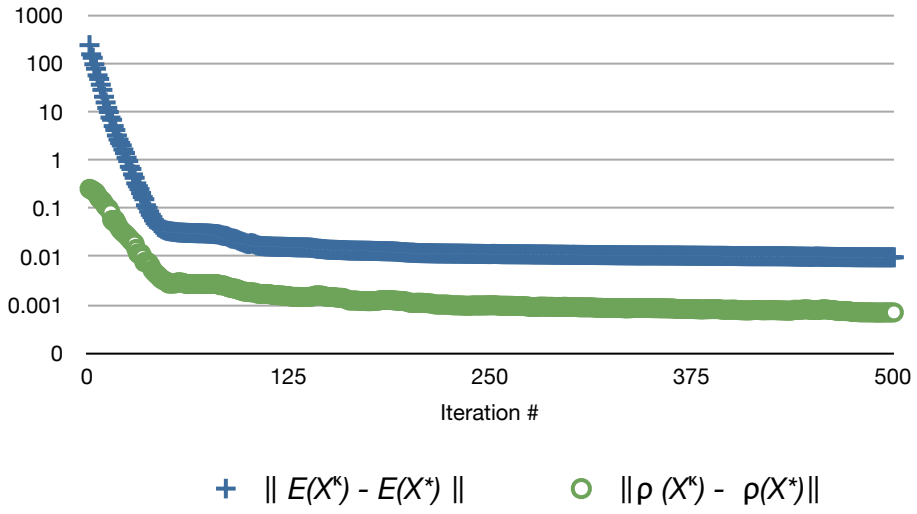


Figure 5.17: (Si_2H_4): $(E^k - E^*)$ and $\|\rho^k - \rho^*\|_2$ (log scale)

5.8.2 Timing experiments

Using the same machine as described in Section 5.5.7, we compare the time for an inner iteration of both RSDFT and KSSPALOPT. The latter consists of one iteration of LMBFGS, including function and gradient evaluations as part of the linesearch and the search-direction computation. Again, we stress that neither implementation explicitly exploits parallelism, nor is written with peak-performance as the main goal. RSDFT is run using two different options to handle the required diagonalization. The first method, referred to as RSDFT-

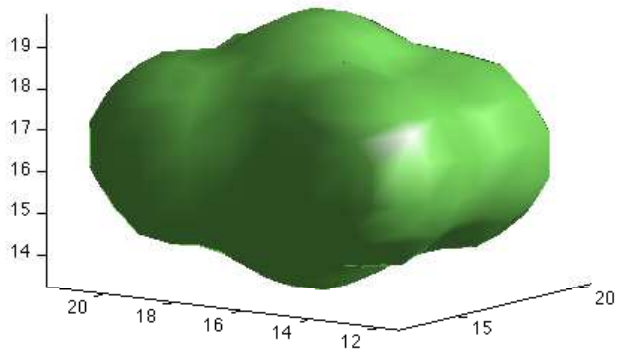


Figure 5.18: The computed charge density of the Si_2H_4 molecule

LANCZOS, applies the Lansczos method to solve the necessary eigenvalue problem required in each iteration of SCF. The second method, referred to as RSDFT-CHEBYSHEV, uses the Lanczos algorithm for the initial eigenvalue solve; however, each subsequent iteration uses the faster *Chebyshev-filtered subspace iteration* [47].

	RSDFT-LANCZOS	RSDFT-CHEBYSHEV	KSSPALOPT
iters	9	11	500
mean(secs)	5.87	1.62	.57
max(secs)	8.17	4.33	.93
min(secs)	5.38	1.29	.46

Table 5.5: Timing experiments for Si_2H_4

The time per iteration is much faster for KSSPALOPT; however, future work must be done to reduce the number of iterations required to reach convergence.

This can be accomplished via a better choice of convergence criteria or by determining a better initial iterate. Moreover, given that the bottleneck in this method is the function and gradient evaluation time, a new implementation that better exploits the structured sparsity of the matrices involved as well as parallelism will be able to reduce the time per iteration significantly.

5.9 C_{60}

Buckminsterfullerene, displayed in Figure 5.19, is a spherical molecule named after the spherical domes designed by Buckminster Fuller. Its structure is not obviously amenable to the localization approaches described in this thesis, but the full augmented Lagrangian method, KSALOPT, is still applicable and competitive with standard SCF computations.

C_{60} has 240 valence electrons corresponding to 120 unique wavefunctions. RSDFT computes an orthogonal solution of -687.504 Ry. The domain is a box with radius 12.75 a.u. using a mesh size of $h = \frac{1}{2}$ a.u..

KSALOPT is run twice, from two different initial matrices of wavefunctions. The first is a random matrix of the proper dimension. The second initial iterate is the matrix of wavefunctions computed after exactly one iteration of RSDFT. Both experiments give solutions with an energy slightly better that RSDFT (see Table 5.6). Figure 5.20 and Figure 5.21 show the rate of convergence of both methods to this better solution. RSDFT is run using only Chebyshev filtering. When using the Lanzcos method for diagonalization, the method did not converge. The isosurface of ρ at the solution computed by KSALOPT is shown in Figure 5.22.

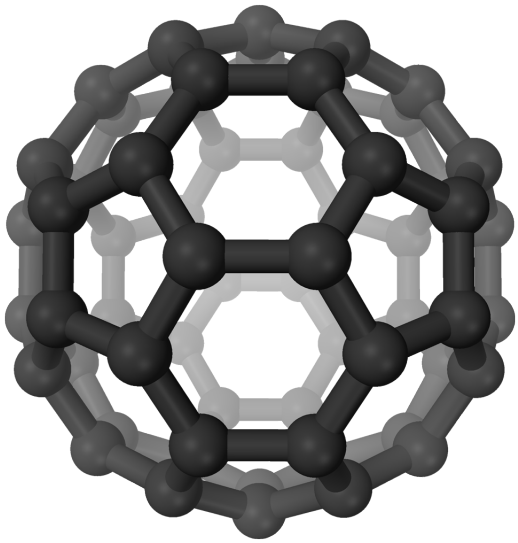


Figure 5.19: Buckminsterfullerene (C_{60})

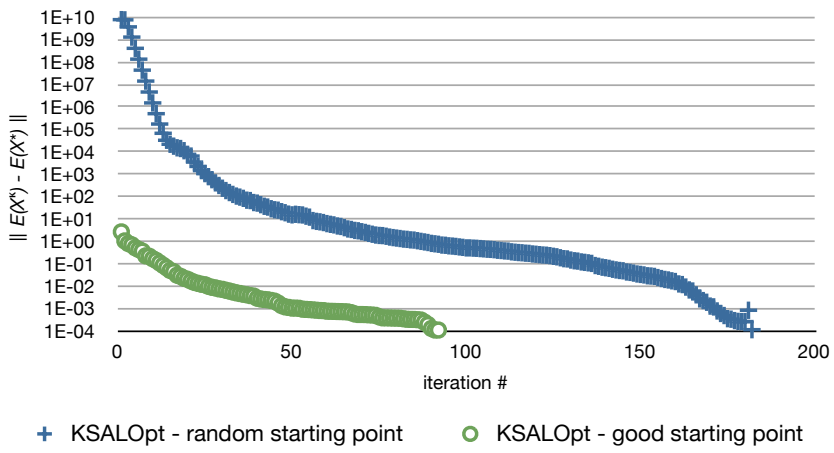


Figure 5.20: (C_{60}): $(E^k - E^*_{\text{KSALOpt}})$ (log scale)

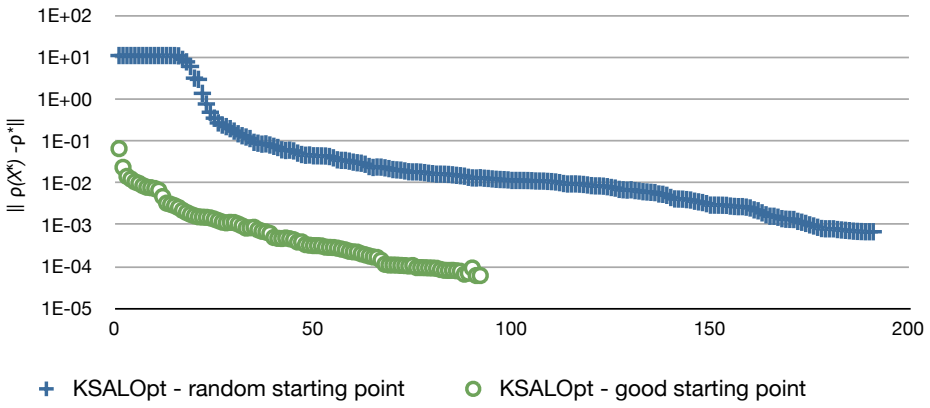


Figure 5.21: (C_{60}): $\|\rho^k - \rho^*_{\text{KSALOpt}}\|_2$ (log scale)

Method	KSALOPT random starting point	KSALOPT good starting point	RSDFT
iter	191	92	14
Energy	-687.5046	-687.5046	-687.5042
mean(secs)	20.7	20.5	61.35
min (secs)	15.2	15.6	52.7
max (secs)	21.48	21.3	168.7

Table 5.6: (C_{60}) Final energies and timings for each method

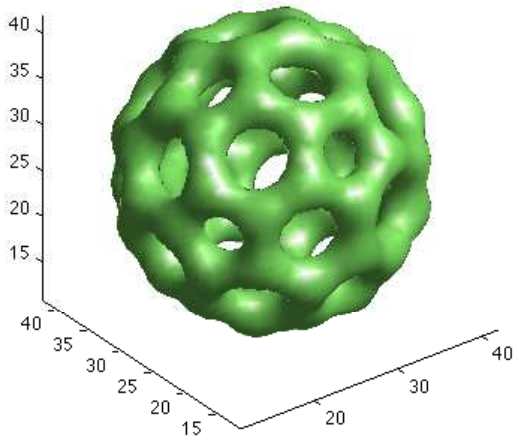


Figure 5.22: Buckminsterminsterfullerene: The computed charge density of the C_{60} molecule

6

CONCLUSION

This dissertation presents two new algorithms for minimizing the Kohn-Sham energy that exploit locality to evaluate the objective and gradient functions at sparse iterates. On a small, yet diverse set of test problems, both algorithms are demonstrated to converge to a solution with energies comparable to the energy at a known orthogonal solution computed by RSDFT. Critical to the performance of each method is the choice of the support or localization region. Experimentally, KSLOCOPT is shown to be able to converge to a solution with lower energy than KSSPALOPT when using a smaller specified support region; however, the convergence of KSLOCOPT is more dependent on the choice of the initial iterate and requires an expensive localization computation. When the localization region is not obvious *a priori*, KSALOPT is demonstrated to converge rapidly in both time and the number of iterations required even though the Kohn-Sham energy and gradient must be computed at dense iterates.

BIBLIOGRAPHY

[1] S. Adhikari and R. Baer. Augmented Lagrangian method for order-N electronic structure. *The Journal of Chemical Physics*, 115:11, 2001.

[2] M.M.G. Alemany, M. Jain, L. Kronik, and J.R. Chelikowsky. Real-space pseudopotential method for computing the electronic properties of periodic systems. *Physical Review B*, 69(7):75101, 2004. ISSN 1550-235X.

[3] T.A. Arias. Multiresolution analysis of electronic structure: Semicardinal and wavelet bases. *Reviews of Modern Physics*, 71(1):267–311, Jan 1999.

[4] T.L. Beck. Real-space mesh techniques in density-functional theory. *Reviews of Modern Physics*, 72(4):1041–1080, Jan 2000.

[5] D.R. Bowler, I.J. Bush, and M.J. Gillan. Practical methods for *ab initio* calculations on thousands of atoms. *International Journal of Quantum Chemistry*, 77(5):831–842, 2000. ISSN 1097-461X.

[6] M. Challacombe. A simplified density matrix minimization for linear scaling self-consistent field theory. *The Journal of Chemical Physics*, 110: 2332, 1999.

[7] J.R. Chelikowsky. *Introduction to the Quantum Theory of Atoms, Molecules, Clusters and Nanocrystals Using MATLAB*. Wiley, in Preparation.

[8] J.R. Chelikowsky, N. Troullier, and Y. Saad. Finite-difference-

pseudopotential method: Electronic structure calculations without a basis. *Physical review letters*, 72(8):1240–1243, 1994. ISSN 1079-7114.

[9] A. Edelman, T.A. Arias, and S.T. Smith. The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 20:303, 1998.

[10] K. Fan. On a theorem of Weyl concerning eigenvalues of linear transformations. *Proceedings of the National Academy of Sciences of the United States of America*, 35(11):652, 1949.

[11] B. Fornberg and D.M. Sloan. A review of pseudospectral methods for solving partial differential equations. *Acta Numerica*, 3:203–267, 1994. ISSN 0962-4929.

[12] W. Gao and Weinan E. Orbital minimization with localization. *Discrete Conliii. Dyn. Syst*, 23:249–264, 2009.

[13] S Goedecker. Linear scaling electronic structure methods. *Reviews of Modern Physics*, 71(4):1085, jan 1999.

[14] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev. B*, 136(3B):B864–B871, 1964.

[15] B. Jansík, S. Høst, P. Jørgensen, J. Olsen, and T. Helgaker. Linear-scaling symmetric square-root decomposition of the overlap matrix. *The Journal of chemical physics*, 126:124104, 2007.

[16] D. Johnson. Modified Broyden's method for accelerating convergence in self-consistent calculations. *Phys. Rev. B*, Jan 1988.

[17] Efthimios Kaxiras. *Atomic and Electronic Structure of Solids*. Cambridge University Press, 2003.

[18] C.T. Kelley. *Iterative methods for optimization*. Society for Industrial and Applied Mathematics, 1999. ISBN 0898714338.

[19] G.P. Kerker. Efficient iteration scheme for self-consistent pseudopotential calculations. *Phys Rev. B*, 23:3082–3084, 1981.

[20] J. Kim, F. Mauri, and G. Galli. Total-energy global optimizations using nonorthogonal localized orbitals. *Phys. Rev. B*, 52(3):1640–1648, Jan 1995.

[21] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4A):A1133–A11388, 1965.

[22] W. Kohn and A. Yaniv. Locality principle in wave mechanics. *Proceedings of the National Academy of Sciences of the United States of America*, 75 (11):5270, 1978.

[23] G. Kresse and J. Furthmuller. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comp Mater Sci*, 6(1):15–50, Jan 1996.

[24] G. Kresse and J. Furtthmüller. Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set. *Physical Review B*, 54(16):11169 – 11185, 1996.

[25] N. Marzari, I. Souza, and D. Vanderbilt. An introduction to maximally-localized wannier functions. *Psi-K Newsletter*, 57:129, 2003.

[26] F. Mauri, G. Galli, and R. Car. Orbital formulation for electronic-structure calculations with linear system-size scaling. *Physical Review B*, 47(15): 9973–9976, 1993. ISSN 1550-235X.

[27] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

[28] P. Ordejón, D. Drabold, R. Martin, and M. Grumbach. Linear system-size scaling methods for electronic-structure calculations. *Phys. Rev. B*, Jan 1995.

[29] Michael L. Overton. HANSO: Hybrid algorithm for non-smooth optimization. URL `http://www.cs.nyu.edu/faculty/overton/software/hanso/index.html`.

[30] J.E. Pask, B.M. Klein, C.Y. Fong, and P.A. Sterne. Real-space local polynomial basis for solid-state electronic-structure calculations: A finite-element approach. *Phys. Rev. B*, 59(19):12352–12358, Jan 1999.

[31] M. Payne, M. Teter, D. Allan, and T. Arias. Iterative minimization techniques for *ab initio* total energy calculation: Molecular dynamics and conjugate gradients. *Reviews of Modern Physics*, 64(4):1045–1097, Jan 1992.

[32] J. P. Perdew and Y. Wang. Accurate and simple analytic representation of the electron-gas correlation energy. *Phys. Rev. B*, 45:13244–13249, 1992.

[33] B.G. Pfrommer, J. Demmel, and H. Simon. Unconstrained Energy Func-

tionals for Electronic Structure Calculations. *Journal of Computational Physics*, 150(1):287–298, 1999. ISSN 0021-9991.

[34] J. C. Phillips. Energy-band interpolation scheme based on a pseudopotential. *Phys. Rev.*, 112(3):685–695, 1958.

[35] J. C. Phillips and L. Kleinman. New method for calculating wave functions in crystals and molecules. *Phys. Rev.*, 116(2):287–294, 1958.

[36] P. Pulay. Convergence acceleration of iterative sequences. The case of SCF iteration. *Chemical Physics Letters*, Jan 1980.

[37] D. Raczkowski, A. Canning, and Lin Wang. Thomas-fermi charge mixing for obtaining self-consistency in density functional calculations. *Phys. Rev. B*, 64(12):4, Sep 2001.

[38] T. Schlick and A. Fogelson. TNPACK-A truncated Newton minimization package for large-scale problems: I. Algorithm and usage. *ACM Transactions on Mathematical Software (TOMS)*, 18(1):46–70, 1992. ISSN 0098-3500.

[39] E. Tsuchida and M. Tsukada. Electronic-structure calculatins based on the finite-element method. *Phys. Rev. B*, 52(8):5573–5578, Jan 1995.

[40] Andreas Waechter. Private communication, 2010.

[41] L.-W. Wang, Z. Zhao, and J. Meza. Linear-scaling three-dimensional fragment method for large-scale electronic structure calculations. *Phys. Rev. B*, 77(16):5, Apr 2008.

[42] C. Yang, J. C. Meza, and L.-W. Wang. A constrained optimization algorithm for total energy minimization in electronic structure calculation. *Journal of Computational Physics*, 217:709–721, 2005.

[43] C. Yang, W. Gao, and J.C. Meza. On the Convergence of the Self-Consistent Field Iteration for a Class of Nonlinear Eigenvalue Problems. *SIAM Journal on Matrix Analysis and Applications*, 30:1773, 2009.

[44] C. Yang, J.C. Meza, B. Lee, and L.W. Wang. KSSOLVâĂŤa MATLAB toolbox for solving the Kohn-Sham equations. *ACM Transactions on Mathematical Software (TOMS)*, 36(2):1–35, 2009. ISSN 0098-3500.

[45] Chao Yang. Private communication, 2009.

[46] M. T. Yin and M. L. Cohen. Theory of *ab initio* pseudopotential calculations. *Phys. Rev. B*, 25(12):7403–7412, 1982.

[47] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky. Self-consistent field calculations using Chebyshev-filtered subspace iteration. *Journal of Computational Physics*, 219:172–184, 2006.

[48] Y. Zhou, Y. Saad, M.L. Tiago, and J.R. Chelikowsky. Parallel self-consistent-field calculations via Chebyshev-filtered subspace acceleration. *Physical Review E*, 74(6):66704, 2006. ISSN 1550-2376.