Robust Mixing for Ab-Initio Quantum Mechanical Calculations

L. D. Marks[1] and D. R. Luke[2, *]

[1] *Department of Materials Science and Engineering,*

Northwestern University, Evanston, IL 60201.

[2] *Department of Mathematical Sciences, University of Delaware, Newark DE 19716-2553, USA.*

We study the general problem of mixing for ab-initio quantum-mechanical problems. Guided by general mathematical principles and the underlying physics, we propose a multisecant form of Broyden's second method for solving the self-consistent field equations of Kohn-Sham density functional theory. The algorithm is robust, requires relatively little fine-tuning and appears to outperform the current state of the art, converging for cases that defeat many other methods. We compare our technique to the conventional methods for problems ranging from simple to nearly pathological.

PACS numbers: PACS: 71.15.-m, 02.70.-c, 31.15.-p, 31.15.ec

I.   INTRODUCTION

We consider the problem of determining the electron density ρ that satisfies the self-consistent field equations according to the Kohn-Sham density functional theory[1,2]:

$$(H_0 + V_\rho)\phi_i = \epsilon_i \phi_i \tag{I.1a}$$

$$\rho(x) = \sum_i \left(1 + e^{\beta(\epsilon_i - \mu)}\right)^{-1} |\phi_i(x)|^2. \tag{I.1b}$$

Here H_0 is the single-particle noninteracting Hamiltonian and V_ρ is an effective potential parameterized by the particle density ρ. The constant β is $1/kT$ where k is Boltzmann's constant and T is temperature. The term $(1 + e^{\beta(\epsilon_i - \mu)})^{-1}$ is the Fermi-Dirac occupation and the constant μ is determined by $\int \rho(x)dx = N$ for an N-body problem. Following[3] we let $H_\rho := H_0 + \lambda V_\rho{}^{60}$ denote the Kohn-Sham Hamiltonian and reformulate the above system of equations as a nonlinear fixed point problem: find ρ such that

$$F(\rho)(x) := \left(1 + e^{\beta(H_\rho - \mu)}\right)^{-1}(x, x) = \rho(x) \tag{I.2}$$

where μ is the unique solution to $N = trace(\left(1 + e^{\beta(H_\rho - \mu)}\right)^{-1})$. We refer to the operator F above as the self-consistent field (SCF) operator. We will not be concerned with the details of the SCF

operator or its approximations since these tend to be specific to the application. Also, we will work with the discretized version of the SCF operator, which we will call the SCF *mapping* since it is a real vector-valued mapping of the discretized density. Throughout this work, however, we will point to instances where the form of this mapping can cause problems for numerical procedures for solving Eq.(I.2).

Numerical algorithms for solving Eq.(I.2) abound – the representative examples we focus on here are[4,5,6,7,8,9,10,11]. These are iterative procedures and the process of determining the desired density ρ from previous estimates has come to be known as "mixing" in the physical literature. For ab-initio methods there is frequently a user-provided mixing term which, if it is improperly chosen, will lead to divergence of the iterations. In many cases the user has to learn by failure what is the correct value to use, expending a fair amount of computer resources in the process. We will show that many of the methods found in the physical literature have counterparts in the mathematical literature thus providing a more systematic approach to the choice of algorithm parameters. The goal of this work is the development of a method that does not require expert user input, is fast, and can handle many of the more complicated and poorly convergent problems such as metallic surfaces or heterostructures that can defeat a novice and sometimes an expert.

In the next section we provide detailed background both to the mathematical literature on methods for solving non-linear equations, as well the physical literature on mixing. There are two major "branches" of numerical techniques distinguished by the space in which they are derived. However we show that all of the different methods are generated by solving an optimization problem of the same form. In Section III we detail our proposed algorithm for solving Eq.(I.2). We show in Section III that most algorithms are predicated upon fixed point mappings with a great deal of regularity, in particular monotone mappings. Motivated by the prospect that most systems of physical interest do not lead to convex SCF mappings, the principal insight that yields our improvements is to treat the prior steps as random samples in a higher-dimensional space rather than deterministic steps in a path to a fixed point. In effect, the Kohn-Sham Hamiltonian evaluated at an electron density far from equilibrium can vary in a semirandom and perhaps chaotic fashion. This viewpoint leads to a natural interpretation of the updates in terms of predicted and unpredicted components. Controls on the algorithm are exerted through parameter choices that affect primarily the unpredicted step. The rest of the section involves technical considerations for controlling step sizes and safeguarding against instabilities. We introduce the idea of using proximal mappings[12] to account for problems associated with near-linear dependencies and, at the same time, to safeguard against uncertainties in the model analogous to a classic Wiener filter. This

regularization has the advantage that it also acts similarly to a classic trust region technique in numerical optimization. In Section IV we present numerical results for both very easy problems as well as semi-pathological cases. The new approach outperforms existing algorithms in most cases, and does significantly better with poorly constructed Kohn-Sham mappings. The algorithm is also relatively insensitive to user input. We conclude with a discussion of some of the open issues.

II.   ITERATIVE METHODS FOR SOLVING NONLINEAR EQUATIONS

We wish to determine the electron density ρ_* with fixed atom locations. The density is a real-valued vector with k elements. With an estimated density ρ_n at the n-th step of an iterative procedure for determining ρ_*, we check whether our estimate satisfies the ab-initio self-consistent field (SCF) equations given by Eq.(I.2). Evaluation of the SCF mapping returns a modified density $\rho'_n := F(\rho_n)$, another real-valued vector with k elements. The density we seek is a fixed point of F, i.e., we solve the system of non-linear equations

$$F(\rho_*) - \rho_* = 0. \tag{II.1}$$

This suggests the usual Newton algorithm as a possible numerical solution strategy.

A.   Newton's Method

Expressing the mapping in Eq.(II.1) by its Taylor series expansion centered on a fixed point ρ_* yields

$$F(\rho) - \rho = (J(\rho_*) - I)(\rho - \rho_*) + O(|\rho - \rho_*|^2) \tag{II.2}$$

where $J(\rho_*)$ is the Jacobian of F (supposing this is well-defined) at ρ_* and I is the identity mapping. The Jacobian J is a k-by-k real-valued matrix. Given a point ρ_n, Newton's method generates the next point ρ_{n+1} by

$$\rho_{n+1} = \rho_n - (J(\rho_n) - I)^{-1}(F(\rho_n) - \rho_n). \tag{II.3}$$

Under standard assumptions, this iteration can be shown to converge quadratically in a neighborhood of a local solution[13]. The computational cost of calculating the Jacobian and inverting is, however, prohibitive for high-dimensional problems such as density functional calculations.

B.   Matrix Secant Methods

A classical approach to avoid computing and inverting the Jacobian is via solutions to the matrix secant equation: $(J(\rho_n) - I) \approx B_n$ where

$$B_n(\rho_n - \rho_{n-1}) = \Big((F(\rho_n) - \rho_n) - (F(\rho_{n-1}) - \rho_{n-1})\Big) \tag{II.4}$$

Introducing new variables, a conventional reduction, this is compactly represented as

$$B_n s_{n-1} = y_{n-1} \tag{II.5}$$

or

$$H_n y_{n-1} = s_{n-1} \tag{II.6}$$

where $H_n = B_n^{-1}$ and

$$s_{n-1} = \rho_n - \rho_{n-1} \quad \text{and} \quad y_{n-1} = (F(\rho_n) - \rho_n) - (F(\rho_{n-1}) - \rho_{n-1}). \tag{II.7}$$

The next density ρ_{n+1} is then generated either by the recursion

$$\rho_{n+1} = \rho_n - B_n^{-1}(F(\rho_n) - \rho_n) \tag{II.8}$$

where B_n satisfies Eq.(II.5), or by

$$\rho_{n+1} = \rho_n - H_n(F(\rho_n) - \rho_n) \tag{II.9}$$

where H_n satisfies Eq.(II.6). Sequences generated by Eq.(II.8) or Eq.(II.9) are known as quasi-Newton methods. The unknowns in Eq.(II.5) and Eq.(II.6) are the matrices B_n and H_n, hence these are systems of k linear equations in k^2 unknowns. There are infinitely many solutions to the matrix secant equation and each different solution leads to a different numerical method for finding the fixed point of F. A common approach in the literature on density functional calculations, due to Srivastava[9], is based on the Broyden rank one matrix updates[14]. Other common updates in the optimization literature are the symmetric rank one (SR1) and the BFGS updates[15]. Our focus in this study is on improvements in the context of Broyden updates, however the basic principles outlined here extend more generally to other matrix secant methods.

C.   Rank One Updates

A new matrix B_{n+1} is obtained by updating in some fashion B_n using the new data pair (s_n, y_n) combined with the prior information $(s_0, y_0), (s_1, y_1), ..., (s_{n-1}, y_{n-1})$ subject to the constraint that

B_{n+1} satisfy Eq.(II.5). In his original paper Broyden[14] looked at two approaches. The first, normally referred to as Broyden's "good" method (GB), finds the nearest matrix to B_n with respect to the Frobenius norm (i.e. the square root of the sum of squares of the matrix entries) that solves Eq.(II.5) (this property, due to Dennis and Moré[16] is slightly different than Broyden's original interpretation). This is given explicitly by

$$B_{n+1} = B_n + \frac{(y_n - \beta_n B_n s_n)s_n^T}{\beta_n ||s_n||^2}. \tag{II.10}$$

Here and throughout this work the norm $||s|| = \sqrt{s^T s}$ is the Euclidean norm and a vector (understood to be a *column* vector) or matrix raised to the power ^{T} indicates the transpose. The scalar $\beta_n > 0$ is a step size parameter which has been retained for formal consistency although it is normally taken to be unity. This method has a simple recursion for its inverse (needed in Eq.(II.8)):

$$B_{n+1}^{-1} = B_n^{-1} + \frac{\left(\beta_n s_n - B_n^{-1} y_n\right) s_n^T B_n^{-1}}{\left(s_n^T B_n^{-1} y_n\right)}. \tag{II.11}$$

Broyden's first method is shown in[16] (Theorem 5.2) to converge locally superlinearly under the standard assumptions that the Jacobian exists, is nonsingular, and Lipschitz continuous at the solution.

The second method proposed by Broyden has the attractive feature that it is a recursion on the inverse Jacobian H_n and is given by

$$H_{n+1} = H_n + \frac{\left(\beta_n s_n - H_n y_n\right) y_n^T}{||y_n||^2}. \tag{II.12}$$

Note that our sign convention is different to the usual sign in the literature where $H_{n+1} = -B_{n+1}^{-1}$. Update Eq.(II.12) was not recommended by Broyden and subsequently became known as Broyden's "bad" method (BB) primarily, in our opinion, because the test problems to which the methods were applied favored Eq.(II.10). Analogously to GB, the BB update finds the nearest matrix to H_n with respect to the Frobenius norm that solves Eq.(II.6).

Generalizing the two methods suggested by Broyden one could consider updates of the form

$$B_{n+1} = B_n + \frac{(y_n - B_n s_n) v_n^T}{v_n^T s_n}. \tag{II.13}$$

Barnes[17] and Gay and Schnabel[18] proposed an update of this form where v_n is the projection of s_n onto the orthogonal complement of $[s_{n-m}, s_{n-m+1}, \ldots, s_n]$ $(0 \leq m < n)$. This strategy, known as *projected updates*, will be discussed in more detail in the next subsection.

A variation of Eq.(II.12) for the inverse recursion H_n was proposed by Martinez and Zambaldi[19] that takes the form

$$H_{n+1} = H_n + \frac{(s_n - H_n y_n)\, e_{n,j}^T}{e_{n,j}^T y_n} \tag{II.14}$$

where $e_{n,j}$ is the j'th column of the identity matrix, chosen so that $\left| e_{n,j}^T y_n \right| = \|y_n\|_\infty$ where $\|y_n\|_\infty$ is the component of y_n with largest absolute value. The matrix update then differs from the previous update only in the j'th column. Numerical experience with this update is very favorable[20,21,22].

Recall that the GB and BB updates are the matrices nearest to the previous matrices with repsect to the Frobenius norm that satisfy the matrix secant equation. The main difference between the GB and BB methods is the space in which the "nearest" criterion is applied[23]. For GB the criterion is applied in the domain of the mapping, while BB is applied in the range, where the domain of the mapping is the space of the density differences s_n and the range is the space of the residual differences y_n. We see from Eq.(II.8) that an ill-conditioned matrix update B_n will lead to a large and numerically unstable estimation of the step s_n since, to compute this one needs to invert B_n. On the other hand, from Eq.(II.9) it is clear that a least change criterion in the space of the residual differences y_n will lead to smaller steps that could slow progress for well-conditioned problems. In other words, the BB algorithm is inherently more conservative than the GB algorithm. The advantage of this property is not apparent until one is faced with ill-conditioned or ill-posed problems – see Section III for a discussion in the context of the physics of a DFT problem. In this respect, the distinction between Broyden's first and second methods is analogous to the distinction between backward, or implicit, and forward techniques for numerical solutions to stiff differential equations. We formulate these statements more precisely below.

The recursions Eq.(II.10)-(II.13) are all one-step recursions in terms of the most recent Jacobian estimate. Most practical implementations make use of multi-step recursions on the inverse Jacobian that allow one to avoid explicitly forming the matrix. In particular, the recursion for Eq.(II.11) with $\beta_j = 1$ $(j = 0, 1, 2, \dots)$, can be written as[24] (Theorem 6.2)

$$B_{n+1}^{-1} = B_0^{-1} - \left(B_0^{-1} Y_n - S_n \right) \left(L_n + S_n^T B_0^{-1} Y_n \right)^{-1} S_n^T B_0^{-1} \tag{II.15}$$

where

$$S_n := [s_0, s_1, s_2, \dots, s_n], \qquad Y_n := [y_0, y_1, y_2, \dots, y_n] \qquad (k\text{-by-}(n+1) \text{ matrices}) \tag{II.16}$$

and

$$(L_n)_{i,j} := \begin{cases} -s_{i-1}^T s_{j-1} & \text{if } i > j \\ 0 & \text{else} \end{cases}.$$

For the multi-step recursion of H_{n+1} an elementary calculation yields the following recursion for Eq.(II.12):

$$H_{n+1} = H_0 \prod_{j=0}^{n} W_j - \sum_{j=0}^{n} \left(Z_j \prod_{i=j+1}^{n} W_i \right). \qquad \text{(II.17)}$$

where the products ascend from left to right with the empty product defined as 1, and

$$W_j := I - \frac{y_j y_j^T}{\|y_j\|^2} \quad \text{and} \quad Z_j := \beta_j \frac{s_j y_j^T}{\|y_j\|^2}.$$

We prefer this recursion over the recursion proposed by Srivastava[9] because, as we shall see in the following sections, we gain valuable insight into the geometry of the operations for the same storage requirements. Srivastava's formulation was implimented for LAPW code by[11], although extensive numerical experience in the Wien2k code indicates that H_0 needs to be adjusted dynamically, for reasons that will be clearer later.

In the recursions Eq.(II.15) and Eq.(II.17) the initial matrix, B_0 and H_0 respectively, is crucial. Several authors have studied optimal initializations[23,25,26,27,28,29]. We explore scalings in greater detail in Subsection III C. A few papers have also explored strategies for combining the updates Eq.(II.15) and Eq.(II.17) in order to take advantage of the strengths of each. IP and Todd[23] derive an update that is a convex combination of GB and BB with weighting determined explicitly so as to yield an optimally conditioned matrix. Their method is locally superlinearly convergent[23] (Corollary 11) under standard assumptions and the additional conditions that the initial approximate Jacobian, B_0, is close to the true Jacobian at the solution and that the sequence of matrix updates and their inverses are uniformly bounded. Martinez[30] switches between GB and BB based on a simple criterion that estimates which method will give the best performance at a given step. Numerical experiments showed this to be a promising approach, but rates of convergence were not pursued.

D. Multisecant Methods

In the context of path following algorithms, it is natural to look only at local information in order to construct a step direction and length. To generate the $n+1$-th Jacobian approximation the

methods described above satisfy the matrix secant equation Eq.(II.5) or Eq.(II.6) for the current step s_n and residual difference y_n. However, for highly nonlinear problems, where iterates may belong to domains of attraction of different solutions, it is perhaps more appropriate to view the previous data as samples of an unknown process in a high dimensional space. In this context, updating the Jacobian based only on the most recent sample and ignoring the other sample points imposes a bias. Invoking some of the terminology used for stochastic optimization, searching for the nearest matrix that satisfies the matrix secant equation only for the most recent sample point is a greedy strategy without recourse.

Multisecant techniques put the previous data on more equal footing with the most recent steps; that is, rather than satisfying the matrix secant equation for only the most recent step one satisfies *all* matrix secant equations *simultaneously*:

$$Y_n = B_n S_n, \tag{II.18}$$

where $S_n = [s_{1,n}, s_{2,n}, \ldots, s_{m,n}]$ and $Y_n = [y_{1,n}, y_{2,n}, \ldots, y_{m,n}]$ are k-by-m ($m \leq \min\{n, k\}$) matrices whose columns are previous steps and residual differences respectively. Multisecant techniques have been studied by several authors in the mathematical literature over the last 48 years[15,17,18,24,31,32,33,34,35,36]. A few authors in the physical sciences[5,6,7,8,10] independently derived updates that, we will show, are simple relaxations of more conventional multisecant methods.

For our application the dimension of the problem, k, is much larger than the number of iterations, n, so we will simply consider m matrix secant equations with $m < n$. We do not as yet specify the composition of S_n and Y_n since there are many options. The construction given in Eq.(II.7) is conventional for matrix secant methods; an alternative construction centers the steps on the initial point ρ_0 rather than the previous step as in Eq.(II.7):

$$s_{j,n} = \rho_j - \rho_0 \quad \text{and} \quad y_{j,n} = (F(\rho_j) - \rho_j) - (F(\rho_0) - \rho_0), \quad (j = 0, 1, \ldots, n-1). \tag{II.19}$$

The method we propose in the following section is centered on the current point

$$s_{j,n} = \rho_j - \rho_n \quad \text{and} \quad y_{j,n} = (F(\rho_j) - \rho_j) - (F(\rho_n) - \rho_n) \quad (j = 0, 1, \ldots, n-1). \tag{II.20}$$

Many multisecant methods are easily understood by formulating the underlying optimization problem each of the approximate Jacobians (implicitly) solves. We consider first the optimization problem

$$\underset{X \in \mathbb{R}^{k \times k}}{\text{minimize}} \frac{\alpha}{2} \|A - X\|^2 + \iota_C(X) \tag{II.21}$$

where, throughout, the norm of a matrix is the *Frobenius* norm, A is a real $k \times k$ matrix and D, G are real $k \times m$ matrices such that the set C defined below is nonempty:

$$C := \left\{ X \in \mathbb{R}^{k \times k} \text{ such that } XD = G \right\}. \tag{II.22}$$

The corresponding indicator function of C, ι_C, is defined by

$$\iota_C(X) = \begin{cases} 0 & \text{if } X \in C \\ \infty & \text{else.} \end{cases}$$

The solution to the optimization problem Eq.(II.21) expressed as

$$\operatorname{argmin}_X \left\{ \frac{\alpha}{2} \|A - X\|^2 + \iota_C(X) \right\} \tag{II.23}$$

is the *prox mapping*[12,37] of the indicator function to C at A:

$$\operatorname{prox}_{1/\alpha, \iota_C}(A) := \operatorname{argmin}_X \left\{ \frac{\alpha}{2} \|A - X\|^2 + \iota_C(X) \right\}.$$

It is an elementary exercise (see, for instance,[38] (Chapter 1, Section G)) to show that

$$\operatorname{prox}_{1/\alpha, \iota_C}(A) = P_C(A) \qquad (\text{for all } \alpha > 0)$$

where

$$P_C(A) = \operatorname{argmin}_{X \in C} \{\|A - X\|\}$$

is the projection of A onto C. This projection has a simple closed form so long as the columns of D are linearly independent:

$$P_C(A) = A + (G - AD) \left(D^T D \right)^{-1} D^T. \tag{II.24}$$

Specializing to multisecants, if $A = B_0$, the n'th approximation to the Jacobian, $D = S_n \in \mathbb{R}^{k \times m}$ and $G = Y_n \in \mathbb{R}^{k \times m}$ ($1 \leq m \leq n$), the columns of which are denoted y_j and s_j respectively ($j \in [0, n]$), then we arrive at the multisecant extension of the Broyden's first update (MSGB) as studied by[17,18,32,35]:

$$B_{n+1} = P_C(B_0) = B_0 + (Y_n - B_0 S_n) \left(S_n^T S_n \right)^{-1} S_n^T. \tag{II.25}$$

Elementary calculations using the Sherman-Morrison-Woodbury formula yield the multi-step recursion for B_{n+1}^{-1}, analogous to Eq.(II.15),

$$B_{n+1}^{-1} = B_0^{-1} + \left(S_n - B_0^{-1} Y_n \right) \left((S_n^T S_n)^{-1} S_n^T B_0^{-1} Y_n \right)^{-1} (S_n^T S_n)^{-1} S_n^T B_0^{-1} \tag{II.26}$$

For the update Eq.(II.13) v_n is the projection of s_n onto the orthogonal complement of $[s_{n-m}, s_{n-m+1}, \ldots, s_n]$ $(0 \leq m < n)$ where s_j and y_j are given by Eq.(II.7). Here the update B_{n+1} will satisfy $m+1$ *consecutive* secant equations $B_{n+1}s_j = y_j$ $(j = n - m, \ldots, n)$. Methods based on this approach are called projected secant updates. This idea, however, seems not to have benefited from the endorsement of prominent researchers at the time, and hence there is little numerical experience to recommend it. A notable exception is the work of Martinez and collaborators[22,34,39,40]. Sequences based on update Eq.(II.25) are shown in[35] (Theorem 2.5) to be locally q-superlinearly convergent if, in addition to other standard assumptions, the approximate Jacobians, B_n stay close to the behavior of the true Jacobian, and if the columns of S_n are strongly linearly independent as measured by the condition number

$$\kappa(S_n) := \|S_n\| \left\| \left(S_n^T S_n\right)^{-1} S_n^T \right\|.$$

An alternative specialization of Eq.(II.24) leads to a multisecant form of Broyden's second method (MSBB) if we let $A = H_0$, $D = Y_n$ and $G = S_n$ so that

$$H_{n+1} = P_C(H_0) = H_0 + (S_n - H_0 Y_n) \left(Y_n^T Y_n\right)^{-1} Y_n^T. \tag{II.27}$$

To our knowledge, there are no published numerical comparisons of Eq.(II.27) to alternatives, neither is there any published convergence theory, though we believe this is only a minor modification of the theory for Eq.(II.25).

Yet another specialization of Eq.(II.24) is a simplex gradient technique[41] for vector-valued functions proposed by Gragg and Stewart[33] and implemented in[34]. Here the Jacobian update is given by

$$B_{n+1} = P_C(0) = Y_n (S_n^T S_n)^{-1} S_n^T. \tag{II.28}$$

where $A = 0$, $D = Y_n$ and $G = S_n$ where the steps s_n and residual differences y_n are centered on the *initial point* ρ_0 as in Eq.(II.19) .

Independent studies appearing in the physics literature that parallel the mathematical literature have a different variational form. The various approaches can all be shown to be specializations of the the following optimization problem

$$\underset{X \in \mathbb{R}^{k \times k}}{\text{minimize}} \ \frac{1}{2} \sum_{j=1}^{n} \alpha_j \, \text{dist}^2_{C_j}(X) + \frac{\alpha_0}{2} \|A - X\|^2 \tag{II.29}$$

where each C_j is a set of the form Eq.(II.22) and $A \in \mathbb{R}^{k \times k}$, and $\text{dist}_{C_j}(X)$ is the Euclidean

distance of X to the set C_j. A short calculation yields the solution X_* to Eq.(II.29)

$$X_* = \frac{\alpha_0}{\sum_{j=0}^n \alpha_j} A + \sum_{j=1}^n \frac{\alpha_j}{\sum_{j=0}^n \alpha_j} P_{C_j} A. \tag{II.30}$$

From Eq.(II.24) this can be written explicitly as

$$X_* = \sum_{j \in \mathbb{J}} \gamma_j A + \sum_{j=1}^n \gamma_j \left((G_j - A D_j) \left(D_j^T D_j \right)^{-1} D_j^T \right) \tag{II.31}$$

where

$$\gamma_j := \frac{\alpha_j}{\sum_{j=0}^n \alpha_j}. \tag{II.32}$$

Specializing to multisecants, let $A = B_n$, $D_j = s_j$ and $G = y_j$, where s_j and y_j are defined by Eq.(II.7). Then the optimization problem Eq.(II.29) corresponds to the variational formulation of a method proposed by Vanderbilt and Louie[10]. A local convergence analysis, together with numerical tests are studied in[5]. Our derivation and formulation of the update, however, appears to be new and clarifies the connections between their method and Eq.(II.25) above:

$$B_{n+1} = \sum_{j=0}^n \gamma_j B_n + \sum_{j=1}^n \gamma_j \left((y_j - B_n s_j) \left(s_j^T s_j \right)^{-1} s_j^T \right). \tag{II.33}$$

If instead we let let $A = H_n$, $D_j = y_j$ and $G_j = s_j$, we get the update proposed by Johnson[6]:

$$H_{n+1} = \sum_{j=0}^n \gamma_j H_n + \sum_{j=1}^n \gamma_j \left((s_j - H_n y_j) \left(y_j^T y_j \right)^{-1} y_j^T \right). \tag{II.34}$$

Again, our derivation is different, and the new formulation makes the connection with Eq.(II.27) more transparent.

The weighting scheme of[6,10] is similar to a technique proposed by Pulay[8]. A dynamic weighting scheme that optimizes the weights γ_j simultaneously with the determination of the matrix H_n or B_n is possible via the extended least squares techniques outlined in[42]. From the analysis above, the "active ingredients" in the methods of[10] and[6] involve the nonuniform weighting of the conventional multisecant formulations of Broyden's methods, and the centering of the prox mapping on the most recent matrix approximation B_n and H_n rather than on B_0 and H_0 as is the convention in Eq.(II.25) and Eq.(II.27). A variation of Eq.(II.34) due to Kawata et al[7] combines the method of Johnson with a construction of the columns of S_n and Y_n proposed by Pulay[8] and given in Eq.(II.20). We note that the methods summarized by Eq.(II.33)-(II.34) and their relatives solve single matrix secant equations *in parallel* while the methods summarized by Eq.(II.25)-(II.27) solve multiple matrix secant equations *simultaneously*, which is more restrictive.

In the above analysis we are not specific about *how many* previous steps should be included in the matrices S_n and Y_n. Recall that these matrices are made up of m columns of previous step information where $m \in [1, n]$. If $m < n$ then one is implicitly executing a *limited memory* technique. The name limited memory stems from the sequential ordering of steps according to Eq.(II.7) and refers to the practice of excluding points more than m steps previous in the construction of the current matrix update[24]. If one constructs S_n and Y_n via Eq.(II.20), as we do in the following numerical experiments, then one would exclude points that are most distant from the current point ρ_n. This is a reasonable strategy for highly nonlinear problems, where the linear approximation that is at the heart of quasi-Newton methods is only valid on a local neighborhood of the current point. For extremely large problems such a strategy is also expedient since the matrix updates need not be explicitly stored as they can be constructed from a few stored vectors.

Finally, we note some other multisecant approaches that don't fit into the framework above. In[35] a multisecant BFGS update is studied, however there do not appear to be computationally efficient and stable methods for generating updates that preserve the algebraic structure of the original BFGS update, namely symmetry and positivity. The SR1 update, also studied in[35], is shown to be computationally efficient and stable though at the cost of symmetry. In contrast to these, Broyden's updates are natural candidates for multisecant approaches since they do not enforce symmetry of the Jacobian (they are designed for solving systems of nonlinear equations rather than for solving optimization problems as with BFGS and SR1). For problems where the Jacobian is ill-conditioned or singular at the solution, Frank and Schnabel[43,44] consider methods based on the third terms of the Taylor series expansion in Eq.(II.2), which they called "tensor methods", that build upon the multisecant model. They reported on average improvements of 33% for highly controlled test problems in which the null space of the Jacobian has dimension 1 or 2. There appears to be little more numerical experience with tensor methods for matrix secant approaches. While this may be an avenue for further research, we do not pursue this idea in the present work.

III. NEW DEVELOPMENTS: SAFEGUARDED MULTI-SECANTS

Newton-like algorithms, such as Broyden's methods, are not global techniques for solving equations and can behave wildly, even chaotically, far from a solution. As we have seen in the previous section, solving the self-consistent field equations is equivalent to finding the fixed points of the SCF mapping F given by Eq.(I.2). The extent to which many algorithms behave, or misbehave,

depends on the functional properties of the SCF mapping. Consider, for instance, an even simpler algorithm to Broyden's method, namely

$$\rho_{n+1} = F(\rho_n); \tag{III.1}$$

that is, we replace the current density with the density under the SCF mapping. This is a simple form of a fixed point iteration. If F is a *contraction* on some closed subset of the space of densities (i.e. points move *closer* to one another under the mapping F), then the sequence ρ_n converges to the *unique* fixed point ρ_* of F (this is known as the Banach Contraction Theorem see, for instance[45]). If F is not a contraction, then Eq.(III.1) could continue forever without ever approaching a fixed point. Successive iterates might form a characteristic path, or they might behave chaotically. One could reasonably conjecture that many Kohn-Sham SCF mappings are not everywhere contractive since Eq.(III.1) is not a popular numerical method. Less restrictive than contractive mappings are *nonexpansive* mappings (i.e. points do not move further apart under the mapping F). Of course, all contractions are nonexpansive, but the converse does not hold. If F is a nonexpansive mapping on a closed, bounded and *convex* subset of the space of densities, then the modified mapping $\widetilde{F}(\rho) := \rho + \lambda(F(\rho) - \rho)$ for small $\lambda > 0$ is contractive, and $\widetilde{F}(\rho)$, and hence F, has a fixed point[46,47]. The iteration to find the fixed point of $\widetilde{F}$ corresponding to Eq.(III.1) in terms of the original mapping F is

$$\rho_{n+1} = \widetilde{F}(\rho_n) := \rho_n + \lambda(F(\rho_n) - \rho_n). \tag{III.2}$$

Most readers will recognize this as the Pratt step[48]. Convergence is guaranteed for nonexpansive SCF mappings on compact convex regions (though convergence can be extremely slow), but if λ is too large, or F is only locally nonexpansive, or worse, not even nonexpansive, then the theory for fixed point iterations like Eq.(III.2) and even Broyden's methods does not have much to say and numerical behavior cannot be guaranteed. In other words, anything can happen.

In particular, since it is possible for the SCF mapping to have several fixed points there is no guarantee that an algorithm will converge to the *correct* fixed point, as is the case if the fixed point is nonphysical, known as a "ghost band". Indeed, the iteration need not converge at all. Even if one supposes that the SCF mapping is contractive on closed neighborhoods of each of these fixed points, if an algorithm takes too large a step it could leave the domain of attraction of one fixed point and drift towards another fixed point. This process of drifting between attractors of the SCF mapping could continue indefinitely, and with a chaotic trajectory, if the algorithm is not sufficiently well controlled.

Note that one can extend the above concepts to a subset of the density variables. For instance, the sp-electron states might behave well, while d-electron states might be very difficult to converge. This suggests a certain separability of the variables associated with these quantities, and that the SCF mapping is more sensitive to some subsets of the density variables than others. Indeed, a frequent observation exhibited in Section IV is that the density within the muffin-tins often behaves very differently to the density in the interstitial region.

The problematic part of the Kohn-Sham mapping is the effective potential V_ρ. In general, there is no closed form for V_ρ. For certain approximations, denoted $\widetilde{V}$, it is possible to prove the correspondence between the fixed points of the corresponding SCF mapping $F_{\widetilde{V}}$ and solutions to the Kohn-Sham equations[3], and, moreover, that $F_{\widetilde{V}}$ is a contraction[49]. However, for exact V_ρ at finite temperatures existence and uniqueness of fixed points is an open question, further complicated by the occurrence of systems with multiple coexisting phases[3]. For the practitioner who simply wants her software to converge for a particular example, unfortunately this means that the algorithms come only with extremely limited warranties that may not even be verifiable.

With this caveat in mind, and before we present the details of our numerical method, we take a moment to describe in physical terms some of the features of ab-initio calculations that are problematic, together with common symptoms of poorly convergent problems.

i. In many cases, for instance bulk MgO, the algorithms reach an acceptable solution in a surprisingly small number of iterations, e.g. $10 - 20$ for 10^4 unknown density components. This implies that, at least for a substantial subset of the density parameters, the domain of attraction of the fixed point is large and the SCF mapping has "good" functional properties on this domain.

ii. In some cases there can be issues with the scaling of different parts of the density because they are represented in quite different fashions. For instance, with LAPW methods the plane wave components outside the muffin-tins are represented by the Fourier coefficients whereas the density inside the muffin-tins is expanded in terms of spherical harmonics.

iii. The conventional wisdom for LAPW-based methods is that the muffin-tins should be as large as possible without overlapping. This implies that the basis set used for the muffin-tins is somehow better suited for the physics or for the geometry of the atoms. This is manifested in more rapid convergence of the coefficients corresponding to these basis elements and indicates that the domain of attraction of the fixed point for these coefficients is large relative to the domain of attraction for the fixed point of the plane wave elements.

iv. The most physically interesting problems are often harder to solve. A spin unpolarized DFT calculation of NiO, for example, may converge very slowly. The slow convergence of the mixing cycle is in part because spin unpolarized the system is metallic, but is also coincidental with an imperfect functional description of this system, in which case the Hamiltonian in Eq.(I.2) can be ill-posed. It is not uncommon to compromise on the physical model, particularly for large and complicated problems.

v. In some cases, for instance when there are d or f electrons, charge carriers are in a large unit cell and for surfaces, mixing converges poorly and can easily diverge. In the literature this is called "charge sloshing" because one has oscillations of charge density between different spatial regions of the problem or between different local states such as d-electrons. Mathematically this sometimes corresponds to ill-conditioning when a small change in the density ρ can lead to large change in $F(\rho)$, with large eigenvalues of the matrix H (or small eigenvalues of B). Alternatively, it may be that the higher-order terms of Eq.(II.2) are large, so neglecting them is only appropriate for a very small change in the density. A third possibility in the case of charge sloshing is that the SCF mapping is not nonexpansive (and hence not contractive) along this trajectory.

vi. It is possible with a LAPW method for the calculations to enter regions where the new density has become unphysical. Such numerical densities are called "ghost bands" and are unphysical solutions of the Kohn-Sham equations Eq.(I.2).

To illustrate these features a simple model is an O_2 molecule starting from atomic densities where the two atoms are deliberately treated differently, one starting with a spin of $+2$ the other with 0. Shown in Figure 1 is the variation of the spin (a) and total charge within the muffin tins (QMT) (b) varies during a representative SCF iteration. While the spins converge smoothly to the final solution, the total charge oscillates or "sloshes". Figure 1 shows the behavior of the spin (a) and the total charge, QMT (b), within the muffin tins for sequences of densities approaching the solution. For this sequence of points the spin of the density ρ_n, shown in Figure 1(a), converges smoothly to the solution. A view of the charges of the electrons within the muffin tins, Figure 1(b), tells a different story. The total charge within the muffin tins appears to zig-zag toward the solution.

The behavior of different physical quantities for sequences of points approaching a fixed point is independent of algorithm design (as opposed to algorithm performance) and is symptomatic of the functional properties of the SCF mapping, i.e. the underlying physical problem. Our purpose is to

(a)   (b)
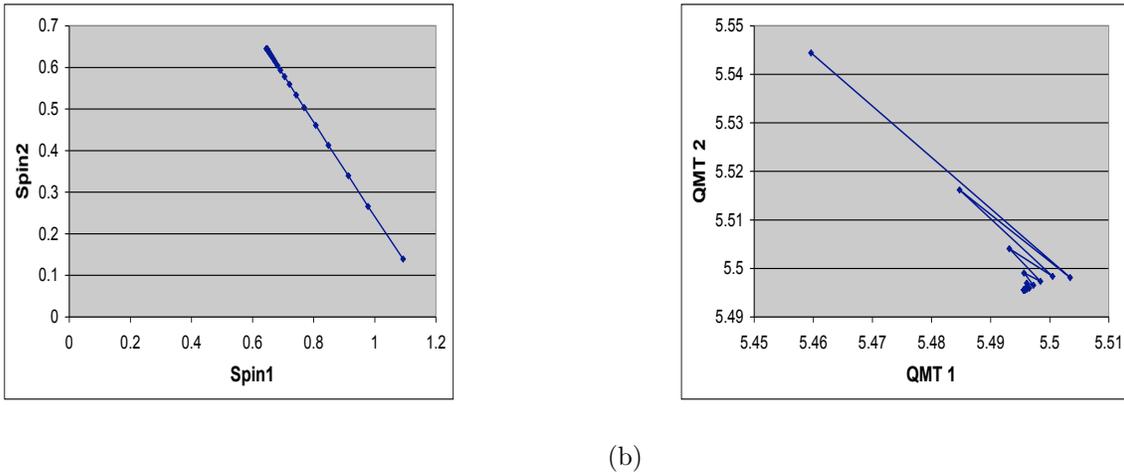
FIG. 1: Iterates for an O_2 molecule with atomic densities having a spin of +2 the other with 0. Figure (a) shows the spin within the muffin tins and (b) shows the total charge within the muffin tins (QMT) during a representative SCF iteration.

design an algorithm that performs well for both regular SCF mappings as well as more pathological cases. As each model presents unique analytical irregularities, we seek to address the most common challenges in our numerical algorithms.

A. Toward a robust algorithm

In addition to numerical performance for a wide range of DFT calculations, a good numerical method, in our opinion, will require little or no user intervention. A key observation is that the dimension of the underlying problem is on the order of 10^4 or higher while the information that one uses to model the fixed point mapping is at most dimension $2n$ where n is the number of iterations (on the order of 10^0). As mentioned above, the conventional view is that the n steps and residual differences generated in matrix secant methods are deterministic points on a path to the solution. While this is true for problems of small and moderate dimension, for high-dimensional problems such as SCF calculations, we can alternatively consider the n steps as random samples of a high-dimensional mapping, albeit with decreasing randomness as the matrix secant model is refined in subsequent iterations. We therefore consider the vectors s_j and y_j given by Eq.(II.7) as merely data samples with *less* significance given to the order in which the samples were collected than with conventional matrix secant approaches. In particular, we center our model on the current iterate

and calculate all steps and residual differences computed relative to the current point according to Eq.(II.20). The algorithm then predicts the behavior of the SCF mapping Eq.(I.2) at ρ_n given an appropriate model of the previous data.

The multi-secant methods detailed in the previous sections can all be rewritten as

$$\rho_{n+1} \;=\; \rho_n - H_0\Big(g_n - Y_{n-1}A_n g_n\Big) - S_{n-1}A_n g_n. \tag{III.3}$$

where $g_n = F(\rho_n) - \rho_n$, A_n is a matrix dependent on the method, and H_0 is an initial matrix secant estimate. Let us write this as

$$s_n = u_n + p_n \tag{III.4}$$

where, according to Eq.(III.3), $p_n = -S_{n-1}A_n g_n$ and $u_n = -H_0(I - Y_{n-1}A_n)g_n$. We show in Subsection III B that p_n is the part of the vector g_n that can be explained by (is in the range of) the data at step n, and u_n is the part that is orthogonal to this information, and hence unpredicted. The numerical challenges described in (iv)-(ii) are embedded in the unpredicted component of the new step. Written this way, we can distinguish two numerical challenges. First and most obvious, one must safeguard against the unpredicted behavior u_n. More subtle is the *a posteriori* accounting of the vector g_n and an underlying *distance to ill-posedness* of the prior information, which reflects the quality of the information. In other words, there is *no free lunch*: not only do we have to be cautious of unpredicted behavior, but we also have to be careful of how we treat our predicted behavior. It is reasonable to expect that with more information the size of the unknown orthogonal component u_n will decrease, however, this is not always the case if the model does not adequately capture the true SCF mapping, or does not adequately adjust for the presence of multiple scales within the model. In addition, a poorly conditioned, or nearly ill-posed model might have an unstable direction in which numerical errors overwhelm meaningful information on the same scale. Our numerical strategy must safeguard against both large unpredicted components and unstable models. We show next how this perspective translates into algorithms.

B. Scaling, regularization, and preconditioning

The discussion in the previous subsection is easily made rigorous when we consider the multisecant formulation of Broyden's second method Eq.(II.27) MSBB with the steps centered according to either Eq.(II.7) or Eq.(II.20). To see this, note that from Eq.(II.9) with H_n replaced by Eq.(II.27), the modification of Eq.(III.3) to MSBB amounts to letting

$$A_n := \left(Y_{n-1}^T Y_{n-1}\right)^{-1} Y_{n-1}^T. \tag{III.5}$$

Note that $\left(Y_{n-1}^T Y_{n-1}\right)^{-1} Y_{n-1}^T g_n$ is the solution to the least squares minimization problem

$$\underset{z \in \mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}\|Y_{n-1}z - g_n\|^2, \tag{III.6}$$

where $m \in [1, n-1]$ is the number of previous data points used in the update. In other words, $\left(Y_{n-1}^T Y_{n-1}\right)^{-1} Y_{n-1}^T g_n$ is the element in the domain of Y_{n-1} that comes closest (in the least squares sense) to "predicting" the vector g_n. It follows, then, that

$$\left(I - Y_{n-1}A_n\right)g_n = \left(I - Y_{n-1}\left(Y_{n-1}^T Y_{n-1}\right)^{-1} Y_{n-1}^T\right)g_n \tag{III.7}$$

is the orthogonal projection of g_n onto the space orthogonal to the residual differences y_j defined by one of Eq.(II.7) or Eq.(II.20), our prior data.

The above discussion assumes that Y_{n-1} is full-rank. If the columns of Y_{n-1} are nearly linearly dependent, then the inverse $\left(Y_{n-1}^T Y_{n-1}\right)^{-1}$ can be numerically unstable. More fundamentally, we are implicitly assuming that the approximation to the Jacobian in Eq.(II.3) is, first of all, valid on the neighborhood of ρ_n defined by the other data points and, second of all, that the Newton step is the *right* step to take. If either one of these assumptions does not hold, as would be the case when we are far from the solution and our sample points are far apart, conventional optimization strategies link local and global techniques by allowing steps to rotate between the steepest descent direction (in the present setting, the direction of the vector g_n) and a Newton-like direction. One well-known strategy of this kind is the Levenberg-Marquardt algorithm[50,51]. We propose a different technique that is an unusual use of a classical regularization technique usually attributed to Tikhonov[52,53,54] and rediscovered in the statistics community under the name of ridge regression[55], though the more general notion of proximal mappings due to Moreau[12] predates both of these. In particular we regularize Eq.(III.6) in the usual way:

$$\underset{z \in \mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}\|Y_{n-1}z - g_n\|^2 + \tfrac{\alpha}{2}\|z\|^2, \qquad (\alpha > 0). \tag{III.8}$$

Recalling the prox mapping introduced in the previous section, we could center the regularization more generally on any point z_0, however, since we have no prior information about z the logical choice for the regularization is to center it on the origin. The solution to Eq.(III.8) is

$$z_n = \left(Y_{n-1}^T Y_{n-1} + \alpha I\right)^{-1} Y_{n-1}^T g_n \tag{III.9}$$

which yields the following regularization of A_n:

$$A_n^\alpha := \left(Y_{n-1}^T Y_{n-1} + \alpha I\right)^{-1} Y_{n-1}^T. \tag{III.10}$$

Note that as $\alpha \to \infty$, $A_n^\alpha \to 0$, and the step generated by Eq.(III.3) rotates to the direction $H_0 g_n$. If H_0 is a scaling of the identity (see Subsection III C) then in the context of optimization, where g_n is actually a gradient, the stronger the regularization, the more the step rotates in the direction of steepest descent and away from a Newton-like direction. We thus interpret the regularization parameter in both the conventional way, stabilizing $\left(Y_{n-1}^T Y_{n-1}\right)^{-1}$, and as an estimation of the uncertainty of the approximate Newton step. Given our understanding of the previous step data as pseudo-random samples from an unknown process, the latter interpretation has a very natural explanation in terms of the Wiener filter for a signal with normally distributed zero-mean white noise. The size of the regularization parameter corresponds to the energy of the noise, or uncertainty in our model.

Johnson[6] proposes a scaling of the columns of the matrices of Y_n and S_n for numerical reasons, though this can easily be shown to have no formal impact on the algorithm. In the context of regularization, however, such a scaling can have a significant effect on the choice of the regularization parameter. This scaling is equivalent to multiplication of the matrices Y_n and S_n on the right by the diagonal matrix Ψ_n. The rescaled least squares problem analogous to Eq.(III.8) is

$$\underset{z \in \mathbb{R}^m}{\text{minimize}} \; \tfrac{1}{2}\|Y_{n-1}\Psi_n z - g_n\|^2 + \tfrac{\alpha}{2}\|z\|^2, \qquad (\alpha > 0) \tag{III.11}$$

with the solution

$$\left(\Psi_n Y_{n-1}^T Y_{n-1}\Psi_n + \alpha I\right)^{-1}\Psi_n Y_{n-1}^T.$$

It follows immediately from this that if we normalize the columns of Y_{n-1},

$$\Psi_n = \begin{pmatrix} 1/\|y_1^{(n-1)}\| & 0 & \dots & 0 \\ 0 & 1/\|y_2^{(n-1)}\| & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1/\|y_m^{(n-1)}\| \end{pmatrix} \tag{III.12}$$

where $y_j^{(n-1)}$ is the j-th column of Y_{n-1}, then our regularization parameter will be *independent* of scaling between the columns of the matrix Y_{n-1}. This allows for more robust regularization strategies. Viewing the regularization as a Wiener filter applied to the approximate Newton step, the scaling reduces the effect of outliers on the regularization parameter in the least squares estimation, these outliers coming from steps that are relatively far away from the solution we seek. We denote the matrix corresponding to this scaling, together with the regularization α by $A_n^{\alpha,\Psi}$ where

$$A_n^{\alpha,\Psi} := \Psi_n \left(\Psi_n Y_{n-1}^T Y_{n-1}\Psi_n + \alpha I\right)^{-1}\Psi_n Y_{n-1}^T. \tag{III.13}$$

The step is then generated by Eq.(III.3) with A_n replaced by $A_n^{\alpha,\Psi}$.

Formalizing our approach for Broyden's first method MSGB is not as obvious. From Eq.(II.8) with B_n^{-1} replaced by Eq.(II.26), the modification of Eq.(III.3) for MSGB amounts to letting $H_0 = B_0^{-1}$ and

$$A_n := \left(\left(S_{n-1}^T S_{n-1}\right)^{-1} S_{n-1}^T B_0^{-1} Y_{n-1} \right)^{-1} \left(S_{n-1}^T S_{n-1}\right)^{-1} S_{n-1}^T B_0^{-1}. \tag{III.14}$$

Again, we note that $\left(S_{n-1}^T S_{n-1}\right)^{-1} S_{n-1}^T B_0^{-1} w$ is the solution to the least squares problem

$$\underset{z \in \mathbb{R}^{n-1}}{\text{minimize}} \; \tfrac{1}{2}\|S_{n-1}z - B_0^{-1}w\|^2.$$

If, in addition, $B_0 = 1/\sigma I$, then an elementary calculation yields the simplification to Eq.(III.14)

$$A_n = \left(S_{n-1}^T Y_{n-1}\right)^{-1} S_{n-1}^T \tag{III.15}$$

If $\left(S_{n-1}^T Y_{n-1}\right)^{-1}$ is well-defined, then the mapping $I - Y_{n-1}A_n$ is a *nonorthogonal projection*[61] onto the nullspace of the columns of S_{n-1}, or in other words, a projection onto the space orthogonal to the range of the columns of S_{n-1}, our prior step data. Unlike Eq.(III.5) the projection is *not* to a nearest element in the range of $S_{n-1}^{\perp}$, hence, by definition, the resulting step will be larger than the orthogonal projection.

The above assumes that $\left(S_{n-1}^T Y_{n-1}\right)^{-1}$ is well-defined, which is likely, but this says nothing of whether or not $S_{n-1}^T Y_{n-1}$ is well-conditioned. Indeed, $S_{n-1}^T Y_{n-1}$ need not have real eigenvalues, or be positive definite. Regularization of $\left(S_{n-1}^T Y_{n-1}\right)^{-1}$ in Eq.(III.14) gives $\left(S_{n-1}^T Y_{n-1} + \alpha I\right)^{-1}$ which shifts the eigenvalues to the right. Since $S_{n-1}^T Y_{n-1}$ could have negative eigenvalues, unless α is chosen large enough, this regularization could result in an even *more* ill-conditioned matrix. Our numerical experience is that $\alpha > 10^{-6}$ is sufficiently large to avoid this possibility for the applications of interest to us.

We turn next to preconditioning and scaling. We propose rescaling the density ρ_n to account for multiple scales between the interstitial electrons and the muffin tin electrons. There are many possible strategies. Such scalings are generically represented by multiplying the density ρ_n at each iteration n *on the left* by an arbitrary invertible diagonal matrix Ω_n. The same scaling must also be applied to the result of the SCF mapping acting upon ρ_n. If these scalings are applied *before* the mixing operation and undone after the mixer yields its proposed step, then one need not change any of the formalism above; specifically, one replaces Y_n, S_n, and A_n in Eq.(III.3) with $\widehat{Y}_n := \Omega_n Y_n$, $\widehat{S}_n := \Omega_n S_n$, and,

$$A_n^{\alpha,\Psi_n,\Omega_n} := \begin{cases} \Psi_n \left(\Psi_n \widehat{S}_{n-1}^T \widehat{Y}_{n-1} \Psi_n + \alpha I \right)^{-1} \Psi_n \widehat{S}_{n-1}^T \Omega_n & \text{(MSGB), or} \\[2mm] \Psi_n \left(\Psi_n \widehat{Y}_{n-1}^T \widehat{Y}_{n-1} \Psi_n + \alpha I \right)^{-1} \Psi_n \widehat{Y}_{n-1}^T \Omega_n & \text{(MSBB)} \end{cases} \tag{III.16}$$

One of the benefits of such scalings is to increase the numerical accuracy of matrix multiplication when the matrices consist of elements of vastly different scales.

The preconditioner used in the numerical experiments in Section IV rescales the change in the interstitial electrons relative to that in the muffin-tin electrons. Recall that the residual of the SCF mapping for the density ρ_n is $g_n = F(\rho_n) - \rho_n$. We represent the interstitial and muffin-tin portions of this residual by $g_n^{(I)}$ and $g_n^{(M)}$ respectively where

$$g_n = \begin{pmatrix} g_n^{(I)} \\ g_n^{(M)} \end{pmatrix}.$$

Denote the averages of the residuals of these components separately by

$$\bar{g}_n^{(I)} = \sum_{j=0}^{n} \|g_j^{(I)}\|/\|g_j\|, \quad \text{and} \quad \bar{g}_n^{(M)} = \sum_{j=0}^{n} \|g_j^{(M)}\|/\|g_j\|, \tag{III.17}$$

Using the formalism above, our preconditioner Ω_n is defined by

$$\Omega_n = \begin{pmatrix} \omega_n I_1 & 0 \\ 0 & I_2 \end{pmatrix} \quad \text{where} \quad \omega_n = \sqrt{\frac{\bar{g}_n^{(M)}}{\bar{g}_n^{(I)}}} \tag{III.18}$$

and I_j is the $l_j \times l_j$ identity matrix where l_j is the dimension of the interstitial/muffin tin electrons respectively. We note that the ω_n term enters the multisecant form squared, hence our use of a square root. Removing this square root is also reasonable, and in some cases is better in numerical tests, but it can be less stable and lead to runaway behavior where the interstitial regions converge too rapidly, upsetting the balance between these and the muffin tin electrons. More sophisticated preconditioning are also plausible, for instance a dielectric term for the plane waves[56,57], though we found this simple form to be very effective.

Before concluding this section we address a physical point. A relevant question is whether the step constructed by these matrix secant methods conserves the total charge – if not, then an additional constraint is needed. By construction all the s_n, y_n values conserve charge, as does g_n, and the preconditioners and scalings simply change the matrix given by Eq.(III.16). The result will then conserve charge automatically within numerical accuracy, so no explicit charge constraint is necessary.

C. Step Control

In Broyden's original numerical experiments he constructed B_0 from a finite difference approximation to the true Jacobian (see[14] (Section 7)). This is not a practical approach for extremely large
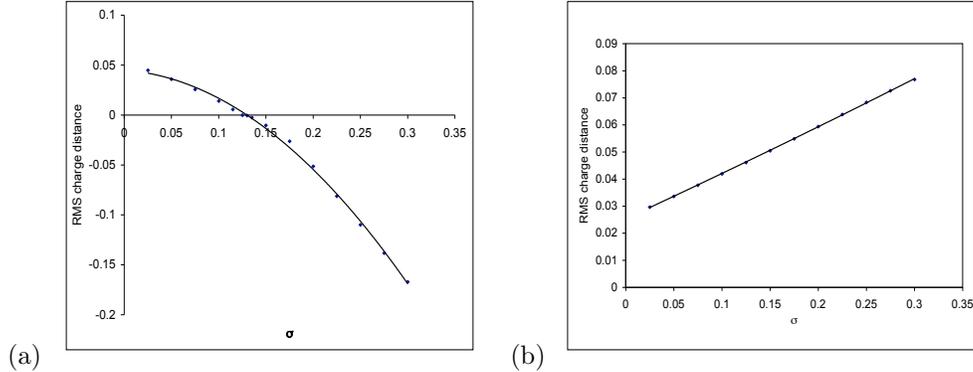
FIG. 2: The root mean squared change of the charge within the muffin tins plotted against σ (a) far from the solution and (b) near the solution.

problems such as DFT calculations. For large problems, the convention for the initial estimate B_0 or H_0 is a scaling of the identity; that is, at each iteration n we choose $H_{0,n} = \sigma_n I$ and likewise $B_{0,n} = 1/\sigma_n$. The choice of the scaling is critical – if it is poorly chosen iterations can stagnate or diverge. Note that the generating matrix can vary at each iteration. With this generating matrix the unpredicted component of the step s_n, given by Eq.(III.3)-(III.4) is $u_n = -\sigma_n(I - Y_{n-1}A_n)g_n$. The scaling σ_n has no impact on the predicted component $p_n = -S_{n-1}A_n g_n$. A technical discussion of strategies for choosing σ_n are intimately connected to a convergence analysis of the algorithms, which is the topic of subsequent work.

For our purposes it suffices to give a number of effective controls, with reasonable heuristics. To motivate our strategy we consider the numerical experiment, Figure 2, showing the root mean squared change of the charge within the muffin tins plotted against various values of σ on different neighborhoods of the solution. Far from the solution, the variation is non-linear (Figure 2(a)) over the range $0.05 \le \sigma \le 0.30$. Since the step size is directly proportional to the size of σ Figure 2(a) indicates that the linear model is correct only on a small neighborhood of the current iterate. If σ is too large, the algorithm chooses a step outside the range of validity of the local model; for an LAPW code this will lead to ghost bands and can lead to divergence. As the iterations proceed, that is, on closer neighborhoods to the solution, the change in the charge as a function of σ appears to be linear (Figure 2(b)) over the same range, indicating that the linear model of the SCF mapping is accurate on a larger neighborhood of the iterate. Thus we interpret σ in the MSBB update as a type of trust-region parameter for a linear model of the SCF mapping.

Our strategy for implementing a dynamic step length σ_n has three parts. First among these is to constrain σ_n so that the step in the direction of the unpredicted component has an upper bound

that is proportional to the size of the predicted component:

$$\sigma_n \leq R|p_n|/|g_n| \tag{III.19}$$

where R is a fixed parameter. In the context of the MSBB update, this ensures that the unpredicted (and unpredictable) component of the step at each iterations does not dominate the total step. One has to take some step along this component, as otherwise no new information is generated; however if too greedy a step is taken the algorithm can diverge. For an "easy" problem R can be larger than for a "difficult" problem since for easy problems the unpredicted component of the step is naturally well-scaled. Our experience indicates that for robust performance across a wide variety of problems, the parameter R is the most important control element for σ_n. For hard problems a value of R from 0.05 to 0.15 works well. As a second level of control, we bound the total variation between successive scalings:

$$\tilde{\sigma}_n = \sigma_{n-1} * \max(0.5, \min(2.0, \|g_{n-1}\|/\|g_n\|)). \tag{III.20}$$

Note that, by this control, σ_n cannot be less than half nor more than twice the previous scaling. Note also that we do not reject steps that yield a larger residual g_n, but rather reduce the size of the step in the unpredicted direction. In almost all cases a large improvement is achieved in the next step by retaining the bad step, though this strategy runs the risk of developing ghost bands. As a third level of control, we include an upper bound on the absolute value of the scaling, $\overline{\sigma}$. For our applications we found $\overline{\sigma} \approx 0.1 - 0.2$ to be effective. Finally, for the very first cycle we take a small step with

$$\sigma_0 = \overline{\sigma} * (0.1 + \exp(-2.0 * \max(dQ, dPW/3.5, dRMT))), \tag{III.21}$$

where dQ is the change in the charge within the muffin tins, dPW is the change in the rescaled plane waves and $dRMT$ is the change of the density within the muffin tins. This form is based upon numerical experience with Wien2k, and is somewhat conservative.

D. Summary

Algorithm III.1 (Regularized, preconditioned, limited-memory multisecant method)

0. Choose an initial ρ_0, σ_0 according to Eq.(III.21), generate $\rho_1 = \rho_0 + \lambda(F(\rho_0) - \rho_0)$ for $\lambda > 0$ some appropriately chosen step length (this is the Pratt step Eq.(III.2)), set $n = 1$ and fix $\alpha > 0$ (10^{-6} to 10^{-4}).

1. If the convergence criterion is met, terminate. Otherwise, given S_{n-1} and Y_{n-1}, whose columns are steps s_j and residual differences y_j respectively ($j = n-m, , n-(m-1), \ldots, n-1$ for some appropriate number of prior steps, e.g. $m = \min\{n, 8\}$) centered on the current point ρ_n as in Eq.(II.20), calculate $A_n^{\alpha, \Psi_n, \Omega_n}$ via Eq.(III.16) for either MSBB or MSGB with the scaling Ψ_n given by Eq.(III.12) and the preconditioner Ω_n given by Eq.(III.17)-(III.18). Determine the value of σ_n according to

$$\sigma_n = \min\{\tilde{\sigma}_n, R|p_n|/|g_n|, \overline{\sigma}\} \tag{III.22}$$

where $\tilde{\sigma}_n$ is given by Eq.(III.20) and $\overline{\sigma}$ is some appropriately chosen upper bound (0.1 to 0.2). Calculate the next step ρ_{n+1} according to Eq.(III.3) with A_n replaced by $A_n^{\alpha, \Psi_n, \Omega_n}$.

2. Evaluate $F(\rho_{n+1})$, set $n = n + 1$ and repeat Step 1.

IV. RESULTS

We test the performance of the algorithm on five examples of increasing physical difficulty, all run using the Wien2k code[4] and the PBE functional[58]; we provide the details below with technical information so they can be reproduced as well as reasons for their choice.

Model 1 Simple bulk MgO, spin-unpolarized with RMT's of 1.8 a.u., an RKMAX of 7 and a $5 \times 5 \times 5$ k-point mesh and a Mermin-functional[2] (i.e. Fermi-Dirac distribution) with a temperature of 0.0068eV. This is a very easy to solve problem.

Model 2 Bulk Pd, spin-unpolarized with RMT's of 2.0 a.u., an RKMAX of 7.5, a $5 \times 5 \times 5$ k-point mesh and a Mermin-functional with a temperature of 0.0068eV. This is slightly harder because of the possibility of sloshing between the d-electron states and the fact that one should use a larger sampling of reciprocal space.

Model 3 A bulk silicon cell with an RMT of 2.16 a.u., an RKMAX of 7.0, a $6 \times 6 \times 6$ k-point mesh and a Mermin-functional with a temperature of 0.0013eV.

Model 4 A $2 \times 2 \times 2$ Pd supercell with a vacancy at the origin, RMT's of 2.5 a.u., an RKMAX of 6.5, a k-point mesh of $3 \times 3 \times 3$ and a Mermin-functional with a temperature of $0.0068eV$. Here, in addition to sloshing between d-electron states one can have longer-range dielectric sloshing. In addition, this is a poorly constructed problem because the RKMAX is too small as is the k-point mesh.

Model 5 A $4.757 \times 4.757 \times 34.957$ a.u., spin-polarized (111) fcc nickel surface with seven atoms in the range $-1/3 \leq z \leq 1/3$. Technical parameters were RMTs of 2.13, an RKMAX of 7 and a $11 \times 11 \times 1$ k-point mesh, also with a Mermin-function temperature of $0.0068eV$. It should be noted that the two surfaces are sufficiently close together, so there is real electron density in the vacuum. In this case one can have spin sloshing, d-electron sloshing as well as long-range Coulomb sloshing of electrons in the vacuum.

In all cases we started from densities calculated as a sum of independent atoms, and the calculations were run with both forms of Broyden multisecants given by Eq.(II.27) and Eq.(II.25), as well as the more conventional Broyden first Eq.(II.10) and second Eq.(II.12) methods. Convergence criteria were an energy change of 10^{-5} Rydbergs and an RMS convergence of the charge within the muffin tins of 10^{-5} electrons. For the multisecant implementations eight prior memory steps were used. To simplify the results, unless noted otherwise we used fixed values of the regularization parameter α of 10^{-4} and $R = 0.1$. In almost all cases, Figure 3 shows that the convergence appears to be linear, although the precision of the calculations does not allow one to observe the final asymptotic behavior, including rates of convergence, of the algorithms.

TABLE I: Iterations to convergence as a function of σ for models $1-5$ with fixed $\alpha = 10^{-4}$ and $R = 0.1$. The mean and standard deviation are for $\overline{\sigma}$ between 0.05 and 0.8 for Models $1-3$, 0.05 to 0.5 for Models 4 and 5.

	MSBB		MSGB		BB		GB	
	mean	stdev	mean	stdev	mean	stdev	mean	stdev
Model 1	16.22	0.44	14.56	1.01	18.67	1.66	22.44	12.71
Model 2	12	0	12.89	0.33	20.11	0.78	39.78	9.58
Model 3	15.44	2.35	16.78	0.67	25.22	2.95	57.56	7.76
Model 4	24.17	2.04	29.17	4.92	–	–	–	–
Model 5	54.60	3.51	–	–	–	–	–	–

For the very simple Model 1 all the methods converge quickly and the parameter $\overline{\sigma}$ has no significant impact on performance. The MSGB method is slightly faster, but as the latter results indicate this is an exception. If $\overline{\sigma}$ is too small (below 0.025) convergence is slower as illustrated in the plot of the number of iterations to convergence versus both $\overline{\sigma}$ and α shown in Figure 4. Interestingly, even for this very simple case the multisecant methods are significantly faster.
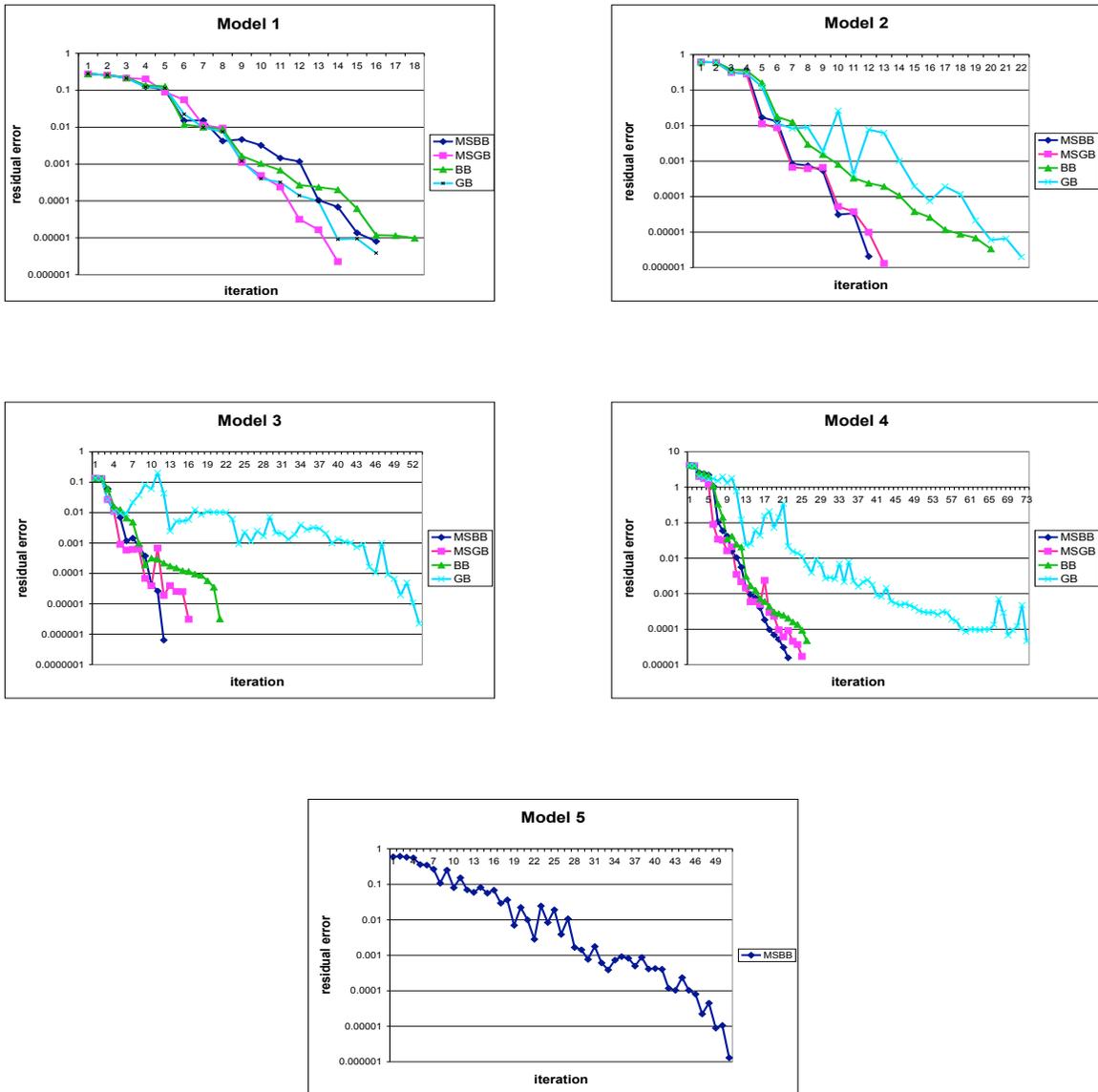
FIG. 3: Plot of the convergence for models 1-5 using the multisecant update based on Broyden's first method (MSGB) and second method (MSBB), compared to Broyden's second method (BB) and Broyden's first method (GB). In model 5 the only algorithm to converge is MSBB.

For the slightly more complicated Model 2, both multisecant methods converge rapidly, whereas the BB method converges more slowly and the GB method is worst by a significant margin. The principal difference between Models 1 and 2 is that in Model 1 there are large changes during the iterations both within the muffin-tins, as well as for the plane waves, whereas in Model 2 almost all the changes are in the plane waves. This supports the rule-of-thumb discussed earlier that one
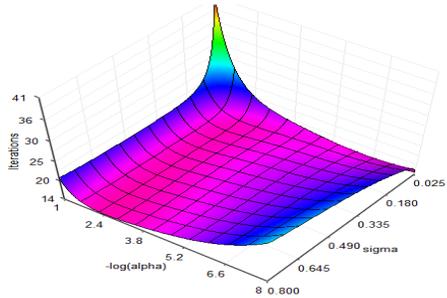
FIG. 4: Number of iterations to convergence for MSGB on Model 1 as a function of the value of the algorithm parameters α and σ. The surface plot has been fitted by a polynomial to obtain smooth contours representing the behavior.

should make the muffin tins as large as possible without overlapping.

With Model 3 the multisecant methods significantly outperform the classical secant methods. For bulk silicon much of the covalent bonding lies in the interstitial region. We conjecture, therefore, that the improvement is due to the improved step direction and size for the multisecant methods that allow these methods to handle the greater variations of the Kohn -Sham mapping for this basis set.

The same trend continues with both Model 4 and Model 5 to the extent that BB and GB only converge for "good" values of $\overline{\sigma}$ (which have to be found by trial and error) and in many cases diverge. For the hardest problem we report here, Model 5, only the MSBB method converged. If one added a line search the other methods would probably converge albeit less rapidly and with a many more SCF evaluations.

The σ parameter in the MSBB update gives one direct control over the size of the steps, which is an important feature for models with strong variations. The control of steps is less immediate for the MSGB update and involves a more sensitive coupling of the regularization parameter α and the step size parameter σ. This is illustrated by the greater variance in performance of the MSGB update versus MSBB for models 1, 2, 4, and 5 shown in Table I.

V. DISCUSSION

To summarize the main points of this work:

- We argue that for DFT problems, where many physically interesting models result in non-contractive SCF mappings, one should consider the information from previous points of the SCF cycle more as samples of a higher-dimensional space than as part of a deterministic path. As a consequence multisecant methods are better than sequential secant updates, as born out in the results.

- There is a fundamental difference between methods based upon Broyden's first (GB) and second (BB) methods in terms of the space they operate in. The second method is more robust and handles poorly constructed, (nearly) ill-posed problems better – in general these are the more interesting physical problems.

- Scaling, regularization and preconditioning have a significant impact on algorithm performance. Moreover, regularization acts simultaneously to reduce instabilities due both to linear dependencies as well as to deficiencies in the model.

- Controlling the step size σ_n along the direction about which no information is available is critical. For difficult problems, this step should in general be *smaller* than for easy problems.

- The multisecant method based upon Broyden's second formulation (MSBB) with appropriate safeguards simply and quickly solves problems which may defeat a novice, sometimes even an expert.

The method we have detailed (MSBB) is robust and has been part of the main Wien2k distribution (7.3) for some months without any apparent problems. Even in the hands of an experienced user for complicated problems such as LDA+U we have been told of cases where the MSBB version is three times faster than the earlier BB code. The default values of $\alpha = 10^{-4}$ and $R = 0.1$ will be approximately correct for a pseudopotential code where preconditioning the variables is not necessary though there are strong variations Kohn-Sham mapping. We have not attempted to impliment the MSBB algorithm for a pseudopotential code but see no reason why it should not work at least as well. One can of course adjust these parameters to improve a single problem, but we recommend values that perhaps are slightly slower in a few cases, but more robust for a wide variety of problems. There may also be ways to stabilize MSGB so that it could possibly work better for pseudopotential codes where preconditioning is easier.

Some additional comments are appropriate about the role of the term in the regularization. As mentioned earlier, we are using this *simultaneously* in three ways, firstly as a standard regularization technique to avoid ill-conditioning associated with near linear dependence of the columns of Y_n, secondly as a Levenberg-Marquardt-type strategy to rotate the step and thirdly in a standard Wiener filter sense to account for model uncertainty. The regularization parameter can be considered to scale proportionally to the noise or uncertainty in the secant equations. Far from the solution the quasi-Newton step may not be appropriate, suggesting that one should use a larger regularization. Similarly, near the solution if the quasi-Newton step is accurate, it will yield faster rates of convergence, in which case one would choose a smaller regularization. While one could dynamically adjust the regularization parameter, for our numerical experiments we choose a relatively large fixed value of α (10^{-4}). This, in our experience, yields adequate overall convergence and better convergence in the "dangerous" early stages of the iterations.

We emphasize once again the link between convergence of the mixing process and the functional properties of the underlying Kohn-Sham mapping. A poorly constructed problem will in most cases converge much more slowly than a well constructed one. This may be a consequence of short-cuts in the DFT calculation, e.g. too few k-points or numerical errors in an iterative diagonalization, or it can be due to a poorly constructed Hamiltonian or perhaps density functional. For the general user poor convergence should be taken as a suggestion that the model of the physics may not have been properly constructed.

The fact that we that we obtain improvement under the assumption that most models of physical interest do not lead to contractive, or more generally monotone SCF mappings raises some questions. It is well established that current density functionals are inexact descriptions of the physics, but the exact analytic properties of many physical systems are unknown. In particular, for many systems it is not known whether the SCF operator is monotone, let alone that it has fixed points, although it is hard to conceive of an experimentally observable equilibrium structure that does not have fixed pionts. An interesting question to raise is whether the SCF operator is monotonic with the "true" density functional that correctly describes the physics. Since in many cases the effective potential V_ρ has no closed form, it is not known whether many of these theoretical properties are verifiable. It is tempting to infer analytic properties from numerical experiments – and we have made numerical progress by doing just this – but one cannot on numerical evidence alone determine the extent to which numerical behavior is indicative of the true nature of the physical system. As a final speculation, we raise the question of whether the character of the SCF mapping can be experimentally measured, or whether this type of behavior is a mathematical

anomaly resulting from being much further away from equilibrium than any feasible experimental system will ever be.

There are several directions of research with regard to algorithms. Firstly, the heuristics for adjusting the step size σ_n need to be put on firm mathematical footing. This would accompany a study of the asymptotic behavior of the algorithm and is the subject of future research. While the analysis of Eq.(III.5) has attractive interpretations in terms of nearest points in the range and space orthogonal to the prior data, the notion of "nearest" is with respect to the usual Euclidean (L^2) norm, which is biased towards outliers. One could consider the development of algorithms based on weighted norms, or even non-Euclidean prox mappings as opposed to those detailed in Subsection II D. The Ω_n considered by[5,6,10] is in the spirit of weighted norms. Other areas for improvement could be found in the initialization of the iterations. We used the Pratt step, however one could use information from a previous SCF iteration.

Finally, while we have used some physics in helping to design the algorithm, there may be more that could be exploited. We find particularly appealing the observation discussed at the beginning of Section III that the density appears to be separable into distinct subsets. One might envision tailoring algorithms to exploit this property. For instance, one could iterate on the components of the density associated with the muffin-tins, while holding the interstitial electron density fixed. Alternatively, one could iterate on the *sp*-electron density holding the *d*-electron density fixed, or one could iterate on other observables such as the spin associated with a particular atom. Such an approach might allow one to isolate irregular variables within the SCF mapping and design algorithms accordingly. This general approach is known as *operator splitting* about which there is a vast literature. (see, for instance[59] and references therein). This would allow one to isolate the analytical properties of the SCF operator and work more directly with specific physical quantities.

---

[*] Electronic address: rluke@math.udel.edu

[1] W. Kohn and S. L. J., Phys. Rev. **140**, A 1133 (1965).

[2] N. D. Mermin, Phys. Rev. **137**, A 1441 (1965).

[3] E. Prodan, J. Phys. A. **38**, 5647 (2005).

[4] P. Blaha, K. Schwarz, G. Madsen, D. Kvasnicka, and J. Luitz, *WIEN2k, An Augmented Plane Wave + Local Orbitals Program for Calculating Crystal Properties* (Institute for Materials Chemistry, TU Vienna, `http://www.wien2k.at/`, 2006).

[5] F. Crittin and M. Bierlaire, in *Proceedings of the 3rd Swiss Transport Research Conference* (STRC, 2003).

[6] D. D. Johnson, Phys. Rev. B **38**, 12807 (1988).

[7] M. Kawata, C. M. Cortis, and R. A. Friesner, J. Chem. Phys. **108**, 4426 (1998).

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

[9] G. P. Srivastava, J. Phys. A: Math. Gen. **17**, L317 (1984).

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

[11] D. Singh, H. Krakauer, and C. S. Wang, Phys. Rev. B **34**, 8391 (1986).

[12] J. J. Moreau, Comptes Rendus de l'Académie des Sciences de Paris **255**, 2897 (1962).

[13] S. Boyd and L. Vandenberghe, *Convex Optimization* (Oxford University Press, New York, 2003).

[14] C. G. Broyden, Mathematics of Computation **19**, 577 (1965).

[15] J. E. Dennis and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice Hall, Englewood Cliffs, NJ, 1996).

[16] J. E. Dennis and J. J. Moré, SIAM Rev. **19**, 46 (1977).

[17] J. G. P. Barnes, Comput. J. **8**, 66 (1965).

[18] D. M. Gay and R. B. Schnabel, in *Nonlinear Programming*, edited by O. L. Mangasarian, R. R. Meyer, and S. M. Robinson (Academic Press, New York, 1978), vol. 3, pp. 245–281.

[19] J. M. Martínez and M. C. Zambaldi, Optim. Method Software **1**, 129 (1992).

[20] E. Spedicato and Z. Huang, Comput. **58**, 69 (1997).

[21] L. Luksan and J. Vlcek, Optim. Methods Software **8**, 185 (1998).

[22] J. M. Martínez, J. Comp. Appl. Math. **124**, 97 (2000).

[23] C. M. IP and M. J. Todd, SIAM J. Numer. Anal. **25**, 206 (1988).

[24] R. H. Byrd, J. Nocedal, and R. B. Schnabel, Math. Prog. **63**, 129 (1994).

[25] W. C. Davidon, Tech. Rep., Atomic Energy Commission Research and Development Report AWL–5990, Argonne National Laboratory, Argonne, IL (1959).

[26] W. C. Davidon, Math. Programming **9**, 1 (1975).

[27] S. S. Oren and E. Spedicato, Math.Prog. **10**, 70 (1976).

[28] D. F. Shanno and K. Phua, Math. Prog. **14**, 149 (1978).

[29] R. B. Schnabel, Math. Programming **15**, 247 (1978).

[30] J. M. Martínez and L. Ochi, Matematica Aplicada e Computational **1**, 135 (1982).

[31] P. Wolfe, Comm. ACM **12**, 12 (1959).

[32] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* (Academic Press, New York, 1970).

[33] W. Gragg and G. Stewart, SIAM J. Numer. Anal. **13**, 889 (1976).

[34] J. M. Martínez, BIT **19**, 236 (1979).

[35] R. B. Schnabel, Tech. Rep. CU-CS-247-83, University of Colorado, Boulder (1983).

[36] J. Ford and I. Moghrabi, J. Comp. Appl. Math. **82**, 105 (1997).

[37] J. J. Moreau, Bull. de la Soc. math. de France **93**, 273 (1965).

[38] R. T. Rockafellar and R. J. Wets, *Variational Analysis* (Springer, Berlin, 1998).

[39] M. A. Gomes-Ruggiero and J. M. Martínez, Math. Modelling Numer. Anal. **26**, 309 (1992).

[40] M. A. Gomes-Ruggiero, J. M. Martínez, and A. C. Moretti, SIAM J. Sci. Statist. Comput. **13**, 459 (1992).

[41] R. Mifflin, Math. Program. **9**, 100 (1975).

[42] D. R. Luke, J. V. Burke, and R. G. Lyon, SIAM Rev. **44**, 169 (2002).

[43] R. B. Schnabel and P. D. Frank, SIAM J. Numer. Anal. **21**, 815 (21).

[44] R. B. Schnabel and P. D. Frank, Tech. Rep. CU-CS-334-86, University of Colorado, Boulder (1986).

[45] J. M. Borwein and A. S. Lewis, *Convex analysis and nonlinear optimization : theory and examples* (Springer Verlag, New York, 2006), 2nd ed.

[46] F. E. Browder, Proc. Nat. Acad. Sci. U.S.A. **54**, 1041 (1965).

[47] W. A. Kirk, Amer. Math. Monthly **72**, 1004 (1965).

[48] G. Pratt, Phys. Rev. **88**, 1217 (1952).

[49] E. Prodan and N. P., J. Stat. Phys. **111**, 967 (2003).

[50] K. Levenberg, The Quarterly of Applied Mathematics **2**, 164 (1944).

[51] D. Marquardt, SIAM J. Appl. Math. **11**, 431 (1963).

[52] A. N. Tihonov, Dokl. Akad. Nauk SSSR **151**, 501 (1963).

[53] A. N. Tihonov, Dokl. Akad. Nauk SSSR **153**, 49 (1963).

[54] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems* (SIAM, 1998).

[55] A. E. Hoerl and R. W. Kennard, Technometrics **12**, 55 (1970).

[56] D. Raczkowski, A. Canning, and L. W. Wang, Phys. Rev. B **64**, 121101(R) (2001).

[57] K. M. Ho, J. Ihm, and J. D. Joannopoulos, Phys. Rev. B **25**, 4260 (1982).

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

[59] D. R. Luke, SIAM J. Optim. ((to appear)).

[60] ":=" distinguishes a *definition* from an equation.

[61] A projection is defined as any mapping P such that $P^2 = P$. An orthogonal projection onto a set C is the point in C that is nearest, with respect to the norm, to the point being projected.