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, Illinois 60201, USA*
[2]*Department of Mathematical Sciences, University of Delaware, Newark, Delaware 19716-2553, USA*

(Received 16 January 2008; revised manuscript received 26 June 2008; published 18 August 2008)

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.

DOI: 10.1103/PhysRevB.78.075114 PACS number(s): 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]

$$(H_0 + V_\rho)\phi_i = \epsilon_i\phi_i \tag{1.1a}$$

$$\rho(x) = \sum_i (1 + e^{\beta(\epsilon_i-\mu)})^{-1}|\phi_i(x)|^2. \tag{1.1b}$$

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 $H_\rho := H_0+\lambda V_\rho$ (Ref. 4) denote the Kohn-Sham Hamiltonian and reformulate the above system of equations as a nonlinear fixed-point problem: find ρ such that

$$F(\rho)(x) := (1 + e^{\beta(H_\rho-\mu)})^{-1}(x,x) = \rho(x), \tag{1.2}$$

where μ is the unique solution to $N=\text{trace}\{[1+e^{\beta(H_\rho-\mu)}]^{-1}\}$. 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. (1.2).

Numerical algorithms for solving Eq. (1.2) abound the representative examples we focus on here are in Refs. 5–12. 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 where systematic approaches to the choice of algorithm parameters are well established. 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 Secs. II and III we discuss the leading methods in an analytical framework that clarifies similarities as well as fundamental differences. Our analysis sheds light on why the algorithms can fail which suggests strategies for design of an improved method. The classes of algorithms we study are predicated upon mappings with a great deal of regularity—properties that the SCF mapping is not guaranteed to satisfy in all instances. Therefore, rather than viewing successive iterates as deterministic steps in a path to a solution, we treat the prior steps as random samples in a high-dimensional space. This viewpoint leads to a natural interpretation of the algorithm in terms of predicted and unpredicted components, as well as the need for regularization and controls on the relative magnitude of the unpredicted step. In Sec. VI we present numerical results for both very easy problems as well as semipathological 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

For fixed atom locations, we wish to determine the electron density ρ_*, a real-valued vector with k elements. With an estimated density ρ_n at the nth step of an iterative procedure for determining ρ_*, we check whether our estimate satisfies the *ab initio* self-consistent field equations given by Eq. (1.2). 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 nonlinear equations

$$F(\rho_*) - \rho_* = 0. \tag{2.1}$$

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

A. Newton-like methods

Given a point ρ_n, Newton's method generates the next approximate solution to Eq. (2.1), ρ_{n+1}, by

$$\rho_{n+1} = \rho_n - [J(\rho_n) - I]^{-1}[F(\rho_n) - \rho_n]. \qquad (2.2)$$

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 is prohibitive for high-dimensional problems such as density-functional calculations. Instead one can approximate the Jacobian via solutions to the matrix secant equation $B_n \approx [J(\rho_n) - I]$ where

$$B_n(\rho_n - \rho_{n-1}) = \{[F(\rho_n) - \rho_n] - [F(\rho_{n-1}) - \rho_{n-1}]\}. \qquad (2.3)$$

Introducing new variables, this is represented as

$$B_n s_{n-1} = y_{n-1} \qquad (2.4)$$

or

$$H_n y_{n-1} = s_{n-1}, \qquad (2.5)$$

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

$$s_{n-1} = \rho_n - \rho_{n-1} \quad \text{and}$$

$$y_{n-1} = [F(\rho_n) - \rho_n] - [F(\rho_{n-1}) - \rho_{n-1}]. \qquad (2.6)$$

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

$$\rho_{n+1} = \rho_n - B_n^{-1}[F(\rho_n) - \rho_n], \qquad (2.7)$$

where B_n satisfies Eq. (2.4), or by

$$\rho_{n+1} = \rho_n - H_n[F(\rho_n) - \rho_n], \qquad (2.8)$$

where H_n satisfies Eq. (2.5). The variables in Eqs. (2.4) and (2.5) are the matrices B_n and H_n, respectively, and there are infinitely many possible solutions, each leading to a different numerical technique. Our focus in this study is on improvements of the Broyden family discussed next.

B. 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 $(s_0, y_0), (s_1, y_1), \ldots, (s_{n-1}, y_{n-1})$ subject to the constraint that B_{n+1} satisfy Eq. (2.4). Broyden[14] looked at two approaches, given here in a multistep recursion.

The first (B1) is based on updates to the approximate Jacobian in Eq. (2.4) and is shown in Ref. 15 to be

$$B_{n+1}^{-1} = B_0^{-1} - (B_0^{-1}Y_n - \beta_n S_n)(L_n + S_n^T B_0^{-1} Y_n)^{-1} S_n^T B_0^{-1}, \qquad (2.9)$$

where β_n is a scaling,

$$S_n := [s_0, s_1, s_2, \ldots, s_n], \quad Y_n := [y_0, y_1, y_2, \ldots, y_n]$$

$$(k\text{-by-}(n+1) \text{ matrices}) \qquad (2.10)$$

and $(L_n)_{i,j} := \{-s_{i-1}^T s_{j-1} \text{ if } i > j; 0 \text{ otherwise}\}$.

The second of Broyden's methods (B2) is based on updates to the approximate *inverse* Jacobian in Eq. (2.5) and is given by

$$H_{n+1} = H_0 \prod_{j=0}^{n} W_j + \sum_{j=0}^{n} \left(Z_j \prod_{i=j+1}^{n} W_i \right), \qquad (2.11)$$

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

$$W_n := I - \frac{y_n y_n^T}{\|y_n\|^2} \quad \text{and} \quad Z_n := \beta_n \frac{s_n y_n^T}{\|y_n\|^2} \quad (n = 1, 2, \ldots).$$

Here and throughout this work the norm $\|y\| = \sqrt{y^T y}$ is the Euclidean norm and a vector (understood to be a *column* vector) or matrix raised to the power^T indicates the transpose. Note that our sign convention is different to the sign in Broyden's paper where he takes $H_{n+1} = -B_{n+1}^{-1}$. Our recursion appears to be new, and for $\beta_n = 1$ can be shown to be equivalent to a recursion proposed by Srivastava[10] with the same storage requirements.

Both Eqs. (2.9) and (2.11) can be performed without storing or forming the matrix explicitly. In the recursions Eqs. (2.9) and (2.11) the initial matrices B_0 and H_0, respectively, are crucial; we explore scalings in greater detail in Sec. III C. Srivastava's formulation was initially implemented for LAPW code by Ref. 12 with H_0 fixed, but a dynamic H_0 yields substantially better performance.

Both of Broyden's methods are shown in Ref. 16 to converge locally superlinearly under the standard assumptions that the Jacobian exists, is nonsingular, and Lipschitz continuous at the solution. Update Eq. (2.11), however, was not recommended by Broyden and subsequently became known as Broyden's "bad" method.

Broyden's updates are the nearest matrices to the previous matrix with respect to the Frobenious norm[17] that satisfy the current matrix secant equation Eq. (2.4) or Eq. (2.5). The main difference between B1 and B2 is the space in which the "nearest" criterion is applied.[18] For B1 the criterion is applied in the domain of the mapping, while B2 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. (2.7) that an ill-conditioned matrix update B_n will lead to a large and possibly unstable estimation of the step s_n. On the other hand, from Eq. (2.8) 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; we return to this issue below.

C. Multisecant methods

To generate the $n+1$th Jacobian approximation the methods described above satisfy the matrix secant equation Eq. (2.4) or Eq. (2.5) for the current step s_n and residual difference y_n. Updating the Jacobian based only on the most recent sample and ignoring the other sample points imposes a bias toward the most recent step. 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*:

$$Y_n = B_n S_n \quad \text{or} \quad S_n = H_n Y_n, \tag{2.12}$$

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 ($m \le \min\{n,k\}$) matrices whose columns are previous steps and residual differences, respectively. Multisecant techniques have been thoroughly studied in the mathematical literature.[15,19–27] Methods appearing independently in the physical sciences literature[6–9,11] are relaxations of more conventional multisecant methods. A very recent study of multisecant methods brought to our attention by an anonymous referee is in Ref. 28.

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

$$\underset{X \in C}{\text{minimize}} \frac{1}{2}\|A - X\|^2, \tag{2.13}$$

where, throughout, the norm of a matrix is the *Frobenius* norm, A is a real $k \times k$ matrix, and the set $C := \{X \in \mathbb{R}^{k\times k}$ such that $XD = G\}$. For D, G real $k \times m$ matrices such that C is nonempty. If the columns of D are linearly independent, the solution X_* to the optimization problem Eq. (2.13) is the *orthogonal projection* of A onto C, written explicitly as

$$X_* = A + (G - AD)(D^T D)^{-1} D^T. \tag{2.14}$$

Specializing to multisecants, if A is an approximation to the Jacobian, $D = S_n \in \mathbb{R}^{k\times m}$, and $G = Y_n \in \mathbb{R}^{k\times m}$ ($1 \le m \le n$), 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 (MSB1) as studied by[20,21,23,25]

$$B_{n+1} = A + (Y_n - AS_n)(S_n^T S_n)^{-1} S_n^T. \tag{2.15}$$

Elementary calculations using the Sherman-Morrison-Woodbury formula yield the multistep recursion for B_{n+1}^{-1}, analogous to Eq. (2.9):

$$B_{n+1}^{-1} = A^{-1} + (S_n - A^{-1}Y_n)[(S_n^T S_n)^{-1} S_n^T A^{-1} Y_n]^{-1}(S_n^T S_n)^{-1} S_n^T A^{-1}. \tag{2.16}$$

Sequences based on update Eq. (2.15) are shown in Ref. 25 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. Moreover, storage requirements for this formulation are no greater than those of Srivastava's implementation of Broyden's second method.

An alternative specialization of Eq. (2.14) leads to a multisecant form of Broyden's second method (MSB2) if we let A be an approximation to the inverse of the Jacobian, $D = Y_n$, and $G = S_n$ so that

$$H_{n+1} = A + (S_n - AY_n)(Y_n^T Y_n)^{-1} Y_n^T. \tag{2.17}$$

To our knowledge, there are no published numerical comparisons of Eq. (2.17) to alternatives, neither is there any published convergence theory, though we believe this is only a minor modification of the theory for Eq. (2.15). Again, the storage requirements for this recursion are equivalent to MSB1 and B2.

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 optimization problem

$$\underset{X \in \mathbb{R}^{k\times k}}{\text{minimize}} \frac{1}{2}\sum_{j=1}^{n} \alpha_j \text{dist}_{C_j}^2(X) + \frac{\alpha_0}{2}\|A - X\|^2, \tag{2.18}$$

where each $C_j := \{X \in \mathbb{R}^{k\times k}$ such that $XD_j = G_j\}$, $A \in \mathbb{R}^{k\times k}$, 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. (2.18)

$$X_* = \sum_{j\in\mathbb{J}} \gamma_j A + \sum_{j=1}^{n} \gamma_j [(G_j - AD_j)(D_j^T D_j)^{-1} D_j^T],$$

$$\text{where} \quad \gamma_j := \frac{\alpha_j}{\sum_{j=0}^{n} \alpha_j}. \tag{2.19}$$

Specializing to multisecants, let $A = B_n$, $D_j = s_j$, and $G_j = y_j$, where s_j and y_j are defined by Eq. (2.6). Then the optimization problem Eq. (2.18) corresponds to the variational formulation of a method proposed by Vanderbilt and Louie.[11] A local convergence analysis, together with numerical tests are studied in Ref. 6. Our derivation and formulation of the update, however, appears to be new and clarifies the connections between their method and Eq. (2.15) above:

$$B_{n+1} = \sum_{j=0}^{n} \gamma_j B_n + \sum_{j=1}^{n} \gamma_j [(y_j - B_n s_j)(s_j^T s_j)^{-1} s_j^T]. \tag{2.20}$$

If instead we let $A = H_n$, $D_j = y_j$, and $G_j = s_j$, we get the update proposed by Johnson:[7]

$$H_{n+1} = \sum_{j=0}^{n} \gamma_j H_n + \sum_{j=1}^{n} \gamma_j [(s_j - H_n y_j)(y_j^T y_j)^{-1} y_j^T]. \tag{2.21}$$

Again, our derivation is different and the new formulation makes the connection with Eq. (2.17) more transparent.

The weighting scheme of Refs. 7 and 11 is similar to a technique proposed by Pulay.[9] 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 Ref. 29. A variation of Eq. (2.21) due to Kawata *et al.*[8] combines the method of Johnson with a construction of the columns of S_n and Y_n proposed by Pulay[9] and given in Eq. (3.3). We note that the methods summarized by Eqs. (2.20) and (2.21) and their relatives solve single matrix secant equations *in parallel* and then average these solutions, while the methods summarized by Eqs. (2.15)–(2.17) seek the single matrix that solves all the 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.[15] If one constructs S_n and Y_n via Eq. (3.3), 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.

III. SAFEGUARDED MULTISECANTS

Newton-like algorithms are not global techniques for solving equations and can behave wildly, even chaotically, far from a solution. 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. The extent to which algorithms behave, or misbehave, depends on the functional properties of the SCF mapping. Consider the following simple algorithm:

$$\rho_{n+1} = F(\rho_n). \tag{3.1}$$

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 (Banach contraction theorem see, for instance, Ref. 30). If F is not a contraction, then Eq. (3.1) could continue forever without ever approaching a fixed point. Successive iterates might form a characteristic path, or they might behave chaotically. Less restrictive than contractive mappings are *nonexpansive* mappings (i.e., points do not move further apart under the mapping F). If F is nonexpansive on a closed convex symmetric subset of Euclidean space $\mathbb{X}$ and has fixed points (as it would if $\mathbb{X}$ were *bounded*[31,32]), then for any $\rho_0 \in \mathbb{X}$ the sequence of steps defined by the iteration

$$(n = 0,1,2,\ldots) \qquad \rho_{n+1} = \widetilde{F}_\lambda(\rho_n) \coloneqq \rho_n + \lambda[F(\rho_n) - \rho_n] \tag{3.2}$$

converges to a fixed point of F (see Theorem 2.1 and Corollary 2.3 of Ref. 33).

Most readers will recognize iteration Eq. (3.2) as the Pratt[34] step. When $F(\rho_n) - \rho_n$ is an approximation to the gradient of the Kohn-Sham energy functional, then the Pratt step is simply an approximation to steepest descent with (fixed) step length λ. Though convergence is guaranteed for nonexpansive SCF mappings on compact convex regions it can be extremely slow. If F is not nonexpansive, then the numerical behavior of fixed-point iterations such as Eq. (3.2) and even Broyden's methods cannot be guaranteed.

Note that one can extend the above concepts to a subset of the density variables. For instance, the *sp*-electron states might converge quickly, while *d*-electron states might be very difficult to converge. Indeed, a frequent observation 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 as $\widetilde{V}$, 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.[35] 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]

With this in mind, and before we present the details of our algorithm, we 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 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 density-functional theory (DFT) calculation of NiO, for example, may converge very slowly. The slow convergence of the mixing cycle is in part because the spin-unpolarized system is metallic, but is also coincidental with an imperfect functional description of this system, in which case the Hamiltonian in Eq. (1.2) 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

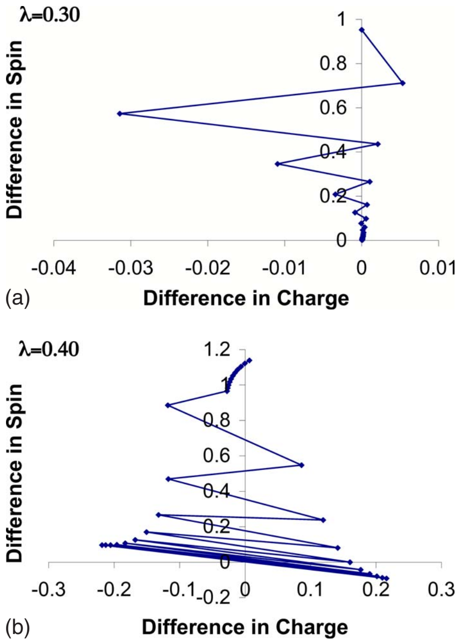

FIG. 1. (Color online) Iterates for an O_2 molecule with atomic densities having a spin of +2 the other with 0. The figures show the difference between the spins (vertical axis) for the two atoms and the difference between the total charges (horizontal axis) within the muffin tins for iterates generated by the Pratt step [Eq. (3.2)] with different fixed step-length parameters λ. In frame (a) the Pratt step parameter is $\lambda=0.30$, in frame (b) the Pratt step parameter is $\lambda=0.40$.

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 in the Taylor-series expansion of the Jacobian 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. None of these possibilities is mutually exclusive.

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 Fig. 1 is the difference between the spins (vertical axis) for the two atoms and the difference between the total charges (horizontal axis) within the muffin tins for iterates generated by the Pratt step [Eq. (3.2)] with different fixed step-length parameters λ. While the spins converge relatively smoothly to the final solution, the total charge oscillates or "sloshes." The charge oscillations become unstable for a relatively small change in the Pratt step parameter.

A. Mathematical framework

There are two elements that distinguish our approach from previous work on matrix secant methods: first is the view of the matrix secant update as a *Jacobian simplex* of a vector-valued mapping and second is the separation of the matrix update into *predicted* and *unpredicted* components. Both of these viewpoints are rooted in the observation that the dimension of the underlying problem is on the order of 10^4 or higher while the information used to model the fixed-point mapping is at most dimension $2n$ where n is the number of iterations (on the order of 10^0). 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. Alternatively, we consider the n steps as random samples of a high-dimensional mapping.

The origins of many matrix secant methods are closely related to the conjugate gradient algorithm. According to this interpretation the construction of the matrices S_n and Y_n given in Eq. (2.6) is consistent with the columns of S_n being *conjugate directions*. Viewing the steps instead as *samples* on a small neighborhood of the current iterate leads us to the alternative centering

$$s_{j,n}=\rho_j-\rho_n \quad \text{and} \quad y_{j,n}=[F(\rho_j)-\rho_j]-[F(\rho_n)-\rho_n]$$

$$(j=0,1,\ldots,n-1). \tag{3.3}$$

The matrix secant update built from these step and residual differences is essentially a finite difference approximation to the Jacobian centered at ρ_n and, in the context of scalar-valued functions, yields what is known as the *gradient simplex*. The generalization in the present context is then appropriately called a Jacobian simplex. We therefore consider the vectors s_j and y_j given by Eq. (3.3) merely as data samples with no significance given to the order in which the samples were collected. This is a fundamentally different approach than the conventional matrix secant updates based on Eq. (2.6).

Independent of how one centers the step history is how one treats the components of the new step generated by the matrix secant update. Given the data samples, the algorithm *predicts* the behavior of the SCF mapping Eq. (1.2) at ρ_n. The multisecant methods detailed in the previous sections can all be rewritten as

$$\rho_{n+1}=\rho_n-A_0(I-Y_{n-1}A_n)g_n-S_{n-1}A_ng_n, \tag{3.4}$$

where $g_n=F(\rho_n)-\rho_n$, A_n is a matrix dependent on the method, and A_0 is an *inverse* Jacobian estimate. Let us write this as

$$\rho_{n+1}-\rho_n=u_n+p_n, \tag{3.5}$$

where, according to Eq. (3.4), $p_n=-S_{n-1}A_ng_n$ and $u_n=-A_0(I-Y_{n-1}A_n)g_n$. We interpret p_n as 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 component that is orthogonal to this information, and hence unpredicted. Intuitively, taking too large a step along the unpredicted component may be a bad idea and one of the sources of instabilities in these methods. We propose a strategy for controlling this step component in Sec. III C. One of the main differences between (MS)B1 and (MS)B2 is the size of the unpredicted component. We show below that this component is inherently larger for (MS)B1 than for (MS)B2. Therefore if instabilities are due to large steps in the unpredicted direction it follows that B2 and MSB2 may perform better, which we will see later is the case for DFT problems.

This is made rigorous when we consider the multisecant formulation of Broyden's second method. Rewriting Eq. (2.8) with H_n replaced by Eq. (2.17) and rearranging components according to Eq. (3.4) yields

$$A_n := (Y_{n-1}^T Y_{n-1})^{-1} Y_{n-1}^T. \tag{3.6}$$

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

$$\underset{z\in\mathbb{R}^m}{\text{minimize}}\frac{1}{2}\|Y_{n-1}z - g_n\|^2, \tag{3.7}$$

where $m \in [1, n-1]$ is the number of previous data points used in the update. Here, $(Y_{n-1}^T Y_{n-1})^{-1} Y_{n-1}^T g_n$ 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

$$(I - Y_{n-1}A_n)g_n = [I - Y_{n-1}(Y_{n-1}^T Y_{n-1})^{-1} Y_{n-1}^T]g_n \tag{3.8}$$

is the orthogonal projection of g_n onto the space orthogonal to the residual differences y_j defined by one of Eq. (2.6) or Eq. (3.3), our prior data.

The formalization for Broyden's first method is not as immediate. From Eq. (2.7) with B_n^{-1} replaced by Eq. (2.16), the modification of Eq. (3.4) for MSB1 amounts to

$$A_n := [(S_{n-1}^T S_{n-1})^{-1} S_{n-1}^T A_0 Y_{n-1}]^{-1} (S_{n-1}^T S_{n-1})^{-1} S_{n-1}^T A_0, \tag{3.9}$$

where, again, A_0 is an estimate of the *inverse* Jacobian. Again, we note that $(S_{n-1}^T S_{n-1})^{-1} S_{n-1}^T A_0 w$ is the solution to the least-squares problem $\underset{z\in\mathbb{R}^{n-1}}{\text{minimize}}\frac{1}{2}\|S_{n-1}z - A_0 w\|^2$. If, in addition, $A_0 = \sigma I$, then an elementary calculation yields the simplification to Eq. (3.9)

$$A_n = (S_{n-1}^T Y_{n-1})^{-1} S_{n-1}^T. \tag{3.10}$$

If $(S_{n-1}^T Y_{n-1})^{-1}$ is well defined, then the mapping $I - Y_{n-1}A_n$ is a *nonorthogonal projection*[36] onto the null space 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. (3.6) 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.

B. Regularization and preconditioning: the matrix A_n

The discussion in Sec. III A of MSB2 assumes that Y_{n-1} is full rank. If the columns of Y_{n-1} are nearly linearly dependent, then the inverse $(Y_{n-1}^T Y_{n-1})^{-1}$ can be numerically unstable. More fundamentally, we are implicitly assuming that the approximation to the Jacobian in Eq. (2.2) 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.[37,38] We propose a different technique that is an unusual use of a classical *regularization* technique usually attributed to Tikhonov[39,40] (see Ref. 41 for more details) and rediscovered in the statistics community under the name of ridge regression,[42] though the more general notion of proximal mappings due to Moreau[43] predates both of these. In particular we regularize Eq. (3.7) in the usual way:

$$\underset{z\in\mathbb{R}^m}{\text{minimize}}\frac{1}{2}\|Y_{n-1}z - g_n\|^2 + \frac{\alpha}{2}\|z\|^2, \quad (\alpha > 0). \tag{3.11}$$

The solution to Eq. (3.11) is

$$z_n = (Y_{n-1}^T Y_{n-1} + \alpha I)^{-1} Y_{n-1}^T g_n, \tag{3.12}$$

which yields the following regularization of A_n given by Eq. (3.6):

$$A_n^{\alpha} := (Y_{n-1}^T Y_{n-1} + \alpha I)^{-1} Y_{n-1}^T. \tag{3.13}$$

Note that as $\alpha \to \infty$, $A_n^{\alpha} \to 0$ and the step generated by Eq. (3.4) rotates to the direction $A_0 g_n$. We thus interpret the regularization parameter in both the conventional way, stabilizing $(Y_{n-1}^T Y_{n-1})^{-1}$ and as an estimation of the uncertainty of the approximate Newton step. Given our understanding of the previous step data as pseudorandom 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.

Similarly, the discussion of MSB1 in Sec. III A also assumes that $(S_{n-1}^T Y_{n-1})^{-1}$ is well defined, but this says nothing of whether or not $S_{n-1}^T Y_{n-1}$ is well conditioned. Regularization of $(S_{n-1}^T Y_{n-1})^{-1}$ in Eq. (3.10) gives

$$A_n^{\alpha} := (S_{n-1}^T Y_{n-1} + \alpha I)^{-1} S_{n-1}^T, \tag{3.14}$$

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. We have found $\alpha > 10^{-6}$ large enough.

Johnson[7] proposes a normalization 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. Such a normalization can, however, have a significant effect on the choice of the regularization parameter. This is equivalent to multiplication of the matrices Y_n and S_n on the right by the diagonal matrix Ψ_n. We show the formalism for MSB2—MSB1 is handled similarly. The least-squares problem analogous to Eq. (3.11) under such a renormalization is

$$\underset{z\in\mathbb{R}^m}{\text{minimize}}\frac{1}{2}\|Y_{n-1}\Psi_n z - g_n\|^2 + \frac{\alpha}{2}\|z\|^2, \quad (\alpha > 0), \tag{3.15}$$

with the solution $(\Psi_n Y_{n-1}^T Y_{n-1} \Psi_n + \alpha I)^{-1} \Psi_n Y_{n-1}^T$. It follows immediately from this that if we normalize the columns of Y_{n-1},

$$\Psi_n = \begin{pmatrix} 1/\|y_1^{(n-1)}\| & 0 & \dots & 0 \\ 0 & 1/\|y_2^{(n-1)}\| & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1/\|y_m^{(n-1)}\| \end{pmatrix}, \tag{3.16}$$

where $y_j^{(n-1)}$ is the jth column of Y_{n-1}, then our regularization parameter will be *independent* of multiple scales between the columns of the matrix Y_{n-1}. Viewing the regularization as a Wiener filter applied to the approximate Newton step, the normalization 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 current point. We denote the matrix corresponding to this renormalization, together with the regularization α by $A_n^{\alpha,\Psi}$ where

$$A_n^{\alpha,\Psi} := \begin{cases} \Psi_n(\Psi_n S_{n-1}^T Y_{n-1}\Psi_n + \alpha I)^{-1}\Psi_n S_{n-1}^T & \text{(MSB1), or} \\ \Psi_n(\Psi_n Y_{n-1}^T Y_{n-1}\Psi_n + \alpha I)^{-1}\Psi_n Y_{n-1}^T & \text{(MSB2).} \end{cases} \tag{3.17}$$

We turn next to preconditioning. We propose rescaling elements of the density ρ_n to account for multiple scales between the interstitial electrons and the muffin-tin electrons. Such a preconditioning is generically represented by multiplying the density ρ_n at each iteration n *on the left* by an arbitrary invertible diagonal matrix Ω_n. One need not change any of the formalism above; specifically, one replaces Y_n, S_n, and A_n in Eq. (3.4) with $\hat{Y}_n := \Omega_n Y_n$, $\hat{S}_n := \Omega_n S_n$, and

$$A_n^{\alpha,\Psi_n,\Omega_n} := \begin{cases} \Psi_n(\Psi_n \hat{S}_{n-1}^T \hat{Y}_{n-1}\Psi_n + \alpha I)^{-1}\Psi_n \hat{S}_{n-1}^T\Omega_n & \text{(MSB1), or} \\ \Psi_n(\Psi_n \hat{Y}_{n-1}^T \hat{Y}_{n-1}\Psi_n + \alpha I)^{-1}\Psi_n \hat{Y}_{n-1}^T\Omega_n & \text{(MSB2).} \end{cases} \tag{3.18}$$

The preconditioner used in the numerical experiments in Sec. IV rescales the change in the interstitial electrons relative to that in the muffin-tin electrons. We represent the interstitial and muffin-tin portions of the residual $g_n = F(\rho_n) - \rho_n$ by $g_n^{(I)}$ and $g_n^{(M)}$, respectively, where $g_n = (g_n^{(I)^T}, g_n^{(M)^T})^T$. The averages of the residuals of these components separately are

$$\bar{g}_n^{(I)} = \sum_{j=0}^{n} \|g_j^{(I)}\|/\|g_j\| \quad \text{and} \quad \bar{g}_n^{(M)} = \sum_{j=0}^{n} \|g_j^{(M)}\|/\|g_j\|. \tag{3.19}$$

Our preconditioner Ω_n is defined by

$$\Omega_n = \begin{pmatrix} \omega_n I_1 & 0 \\ 0 & I_2 \end{pmatrix}, \quad \text{where} \quad \omega_n = \sqrt{\frac{\bar{g}_n^{(M)}}{\bar{g}_n^{(I)}}}, \tag{3.20}$$

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. More sophisticated preconditioning are also plausible, for instance a dielectric term for the plane waves,[44,45] though we found this simple form to be very effective.

Before concluding this section we note that, by construction, the vectors s_n and y_n conserve charge, as does the residual g_n, and the preconditioners and normalizations do not have any effect on the charge. The result will then conserve charge automatically within numerical accuracy, so no explicit charge constraint is necessary.

C. Step control and the generating matrix A_0

In Broyden's original numerical experiments he constructed the initial matrix A_0 from a finite difference approximation to the true Jacobian [see Ref. 14 (Sec. VII)]. This is not a practical approach for DFT calculations. The convention for the initial estimate is a scaling of the identity; that is, at each iteration n we choose $A_{0,n} = \sigma_n I$. From the previous analysis, the magnitude of the unpredicted step depends upon σ_n, increasing linearly for both B2 and MSB2 and in general increasing for both B1 and MSB1 as well although in a more complicated fashion. Therefore, by controlling σ_n we control the size of the step in the unpredicted direction. The choice of the scaling is critical—if it is poorly chosen iterations can stagnate or diverge. A more technical discussion of strategies for choosing σ_n is 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. Our strategy for implementing a dynamic step length σ_n has three parts. First we 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:

$$\sigma_n \le R|p_n|/|g_n|, \tag{3.21}$$

where R is a fixed parameter. One has to take some step along this component, as otherwise no new information is generated; however if too large a step is taken the algorithm can diverge. As a second level of control, we bound the total variation between successive scalings:

$$\tilde{\sigma}_n = \sigma_{n-1} * \max[0.5, \min(2.0, \|g_{n-1}\|/\|g_n\|)]. \tag{3.22}$$

Note 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. As a third level of control, we include an upper bound on the absolute value of the scaling $\bar{\sigma}$.

Of these controls on σ_n our numerical experience is that the parameter R is the most important. For hard problems we have found that a value of R from 0.05 to 0.15 works well. The upper bound on σ_n that we have found to be effective is $\bar{\sigma} \approx 0.1-0.2$. These values are problem specific, however, and may fail for examples we have not considered. An automatic dynamic choice for σ_n in conjunction with standard trust region strategies is the subject of future research.

Before concluding we note that for the very first cycle we take a small step with

$$\sigma_0 = \bar{\sigma} * \{0.1 + \exp[-2.0 * \max(dQ, d\mathrm{PW}/3.5, dRMT)]\}, \tag{3.23}$$

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. (3.23), 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. (3.2)], set $n=1$ and fix $\alpha > 0$ (10^{-6}–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. (3.3), calculate $A_n^{\alpha,\Psi_n,\Omega_n}$ via Eq. (3.18) for either MSB2 or MSB1 with the scaling Ψ_n given by Eq. (3.16) and the preconditioner Ω_n given by Eqs. (3.19) and (3.20). Determine the value of σ_n according to

$$\sigma_n = \min\{\tilde{\sigma}_n, R|p_n|/|g_n|, \bar{\sigma}\}, \tag{3.24}$$

where $\tilde{\sigma}_n$ is given by Eq. (3.22) and $\bar{\sigma}$ is some appropriately chosen upper bound (0.1–0.2). Calculate the next step ρ_{n+1} according to Eq. (3.4) 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[5] and the PBE functional;[46] 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 RMTs of 1.8 a.u., an RKMAX of 7, a 5×5×5 k-point mesh, and a Mermin functional[2] (i.e., Fermi-Dirac distribution) with a temperature of 0.0068 eV. This is a very easy to solve problem.

Model 2. Bulk Pd, spin-unpolarized with RMTs of 2.0 a.u., an RKMAX of 7.5, a 5×5×5 k-point mesh, and a Mermin functional with a temperature of 0.0068 eV. 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 6×6×6 k-point mesh, and a Mermin functional with a temperature of 0.0013 eV.

Model 4. A 2×2×2 Pd supercell with a vacancy at the origin, RMTs of 2.5 a.u., an RKMAX of 6.5, a k-point mesh of 3×3×3, and a Mermin functional with a temperature of 0.0068 eV. 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 4.757×4.757×34.957 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 11×11×1 k-point mesh, also with a Mermin-function temperature of 0.0068 eV. 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 Eqs. (2.15) and (2.17), as well as the more conventional Broyden first Eq. (2.9) and second Eq. (2.11) 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, Fig. 2 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.

For the very simple model 1 all the methods converge quickly and the parameter $\bar{\sigma}$ has no significant impact on performance. The MSB1 method is slightly faster, but as the latter results indicate this is an exception. If $\bar{\sigma}$ is too small (below 0.025) convergence is slower. Interestingly, even for this very simple case the multisecant methods are significantly faster.

For the slightly more complicated model 2, both multisecant methods converge rapidly, whereas the B2 method converges more slowly and the B1 method is worse 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 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

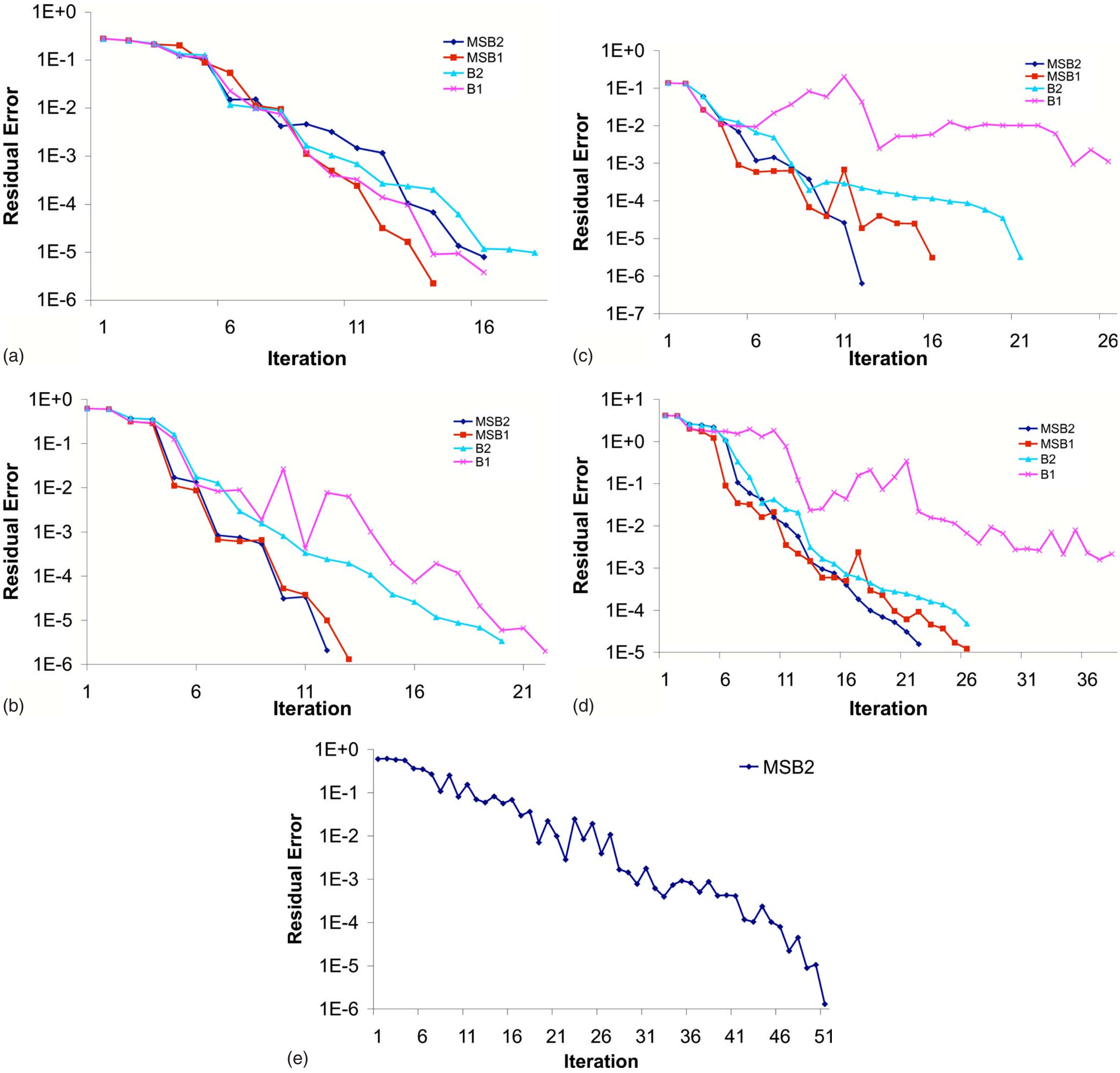

FIG. 2. (Color online) Plot of the convergence for models 1–5 [frames (a)–(e), respectively] using the multisecant update based on Broyden's first method and second method, compared to Broyden's second method and Broyden's first method. In model 5 the only algorithm to converge is MSB2.

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 models 4 and 5 to the extent that B2 and B1 only converge for "good" values of $\bar{\sigma}$ (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 MSB2 method converged. If one added a line search the other methods would probably converge albeit less rapidly and with many more SCF evaluations.

The σ parameter in the MSB2 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 MSB1 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 MSB1 update versus MSB2 for models 1, 2, 4, and 5 shown in Table I.

V. DISCUSSION

To summarize the main points of this work:

(i) We argue that for DFT problems, where many physically interesting models result in noncontractive SCF mappings, one should consider the information from previous

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 $\bar{\sigma}$ between 0.05 and 0.8 for models 1–3 and 0.05–0.5 for models 4 and 5.

	MSB2		MSB1		B2		B1	
	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	–	–	–	–	–	–

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.

(ii) There is a fundamental difference between methods based upon Broyden's first (B1) and second (B2) 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.

(iii) 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.

(iv) 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.

(v) The multisecant method based upon Broyden's second formulation (MSB2) with appropriate safeguards simply and quickly solves problems which may defeat a novice, sometimes even an expert.

The method we have detailed MSB2 is robust and has been part of the main WIEN2k distribution since August 2007 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 MSB2 version is three times faster than the earlier B2 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 implement the MSB2 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 MSB1 so that it could possibly work better for pseudopotential codes where preconditioning is easier.

We acknowledge that we have only considered relatively small problems here, but experience indicates that the convergence depends only very weakly (if at all) upon the size of the problem either in terms of the size of the basis set or the number of atoms. For instance, for a h-BN/Rh(111) nanomesh slab of 1108 atoms[47] with the earlier Broyden mixing algorithm it did not converge even after 200 iterations, but did in 30–40 with the new algorithm. For other large structures, for instance a Si (111) 7×7 surface with 498 atoms, starting from neutral atoms the convergence is only slightly slower than it is for model 3. 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; a single atom may converge slower than 10^4 atoms. 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.

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: first as a standard regularization technique to avoid ill conditioning associated with near linear dependence of the columns of Y_n; second as a Levenberg-Marquardt-type strategy to rotate the step; and third 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.

The fact 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 points. 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. First, 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. (3.6) 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 toward outliers. One could consider the development of algorithms based on weighted norms, or even non-Euclidean prox mappings as opposed to those detailed in Sec. III C. The Ω_n considered by Refs. 6, 7, and 11 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 Sec. 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 Ref. 48 and references therein). This would allow one to isolate the analytical properties of the SCF operator and work more directly with specific physical quantities.

ACKNOWLEDGMENTS

This work was funded by NSF under Grants No. DMR-0455371/001 (L.D.M.) and No. DMS-0712796 (D.R.L.).

*rluke@math.udel.edu

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

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

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

[4] distinguishes a *definition* from an equation.

[5] P. Blaha, K. Schwarz, G. Madsen, D. Kvasnicka, and J. Luitz, *WIEN2k, An Augmented Plane Wave Plus Local Orbitals Program for Calculating Crystal Properties* (Institute for Materials Chemistry, Technical University of Vienna, Vienna, 2006).

[6] F. Crittin and M. Bierlaire, Proceedings of the Third Swiss Transport Research Conference (STRC), 2003 (unpublished).

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

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

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

[10] G. P. Srivastava, J. Phys. A **17**, L317 (1984).

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

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

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

[14] C. G. Broyden, Math. Comput. **19**, 577 (1965).

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

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

[17] The Frobenious norm of a matrix is the square root of the sum of squares of the matrix entries.

[18] C. M. IP and M. J. Todd, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **25**, 206 (1988).

[19] P. Wolfe, Commun. ACM **2** (12), 12 (1959).

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

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

[22] W. Gragg and G. Stewart, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **13**, 889 (1976).

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

[24] J. M. Martínez, BIT, Nord. Tidskr. Inf.behandl. **19**, 236 (1979).

[25] R. B. Schnabel, University of Colorado Technical Report No. CU-CS-247-83, 1983 (unpublished).

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

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

[28] H. Fang and Y. Saad, University of Minnesota Technical Report, 2007 (unpublished).

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

[30] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization :Theory and Examples*, 2nd ed. (Springer-Verlag, New York, 2006).

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

[32] W. A. Kirk, Am. Math. Monthly **72**, 1004 (1965).

[33] J. B. Baillon, R. E. Bruck, and S. Reich, Houst. J. Math **4**, 1 (1978).

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

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

[36] 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.

[37] K. Levenberg, Q. Appl. Math. **2**, 164 (1944).

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

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

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

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

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

[43] J. J. Moreau, Acad. Sci., Paris, C. R. **255**, 2897 (1962).

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

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

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

[47] R. Laskowski and P. Blaha, J. Phys.: Condens. Matter **20**, 064207 (2008).

[48] D. R. Luke, SIAM J. Optim. **19**, 714 (2008).