

# Predictive Mixing for Density Functional Theory (and Other Fixed-Point Problems)

L. D. Marks\*



Cite This: <https://doi.org/10.1021/acs.jctc.1c00630>



Read Online

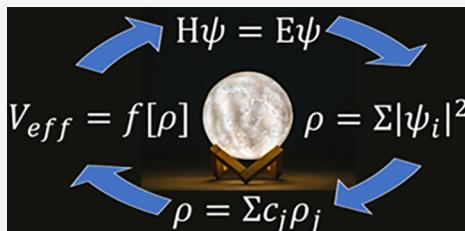
ACCESS |

Metrics & More

Article Recommendations

Supporting Information

**ABSTRACT:** Density functional theory calculations use a significant fraction of current supercomputing time. The resources required scale with the problem size, the internal workings of the code, and the number of iterations to convergence, with the latter being controlled by what is called “mixing”. This paper describes a new approach to handling trust regions within these and other fixed-point problems. Rather than adjusting the trust region based upon improvement, the prior steps are used to estimate what the parameters and trust regions should be, effectively estimating the optimal Polyak step from the prior history. Detailed results are shown for eight structures using both the “good” and “bad” multisecant versions as well as the Anderson method and a hybrid approach, all with the same predictive method. Additional comparisons are made for 36 cases with a fixed algorithm greed. The predictive method works well independent of which method is used for the candidate step, and it is capable of adapting to different problem types particularly when coupled with the hybrid approach.



## 1. INTRODUCTION

A classic problem in many different areas of science is solving a set of nonlinear equations, what is called solving a fixed-point problem. One area where this is very important is in density functional theory (DFT), where the solution to the Kohn–Sham equations<sup>1,2</sup> is a fixed-point density under the action of a nonlinear quantum mechanical operation. The total time (and computational effort) needed to converge scales as the multiple of the time for each individual step, for instance solving the eigenvalue problem, and the total number of steps. The latter is determined by how efficiently the fixed-point problem is solved.

While solving fixed-point problems (root finding and/or nonlinear equations) is described in many undergraduate courses on numerical methods, they are not as simple as they might seem. This is particularly the case when gradients are either not available or are prohibitively expensive to calculate, as in DFT problems. The most common approach for these is to use quasi-Newton methods or a least-squares approach to approximate the Jacobian—the two are mathematically equivalent to second order. This is similar to methods used for optimization where the mathematical background and methods are much better developed. Unfortunately, fixed-point problems do not necessarily possess the local convexity of optimization problems, and convergence is not so readily guaranteed. Indeed, it is really somewhat amazing that DFT problems with thousands of variables often converge in only a few tens of self-consistent iterations. If they did not converge that fast, modern ab initio methods would be considerably less useful.

It is established that one can obtain convergence of the Kohn–Sham equations with appropriately chosen parameters for the self-consistent iterations and exact functionals, e.g., refs

3–9, although whether this will occur in finite computational time is not guaranteed. Unfortunately, convergence with general parameters, imperfect functionals, and starting points far from the solution is not guaranteed. While convergence can often be obtained by adjusting internal parameters until they work, this is highly undesirable. A target has to be a method that will converge fast without needing user intervention.

Why do the calculations sometimes not converge? In the DFT literature how they behave has often been analyzed in terms of physical variations of the electron density. A classic example of this is for large calculations with partially occupied states at the Fermi level, for instance for a surface. In such cases the density can oscillate across the material, for instance between the outer surface and the inside. This phenomenon is called “sloshing” in the DFT literature, and the oscillations can increase in magnitude leading to divergent behavior. This would be described differently mathematically, namely as a consequence of the algorithm taking steps that are larger than appropriate leading to instabilities.

It is established in optimization and general methods for nonlinear equations (e.g., refs 10–14) that convergence can only be guaranteed if either a line search or a trust region is used to control the step size; for more details on trust region methods, see Yuan.<sup>15</sup> In most cases a trust region approach is faster than a

Received: June 24, 2021

line search for DFT as it requires a smaller number of computationally (very) expensive calculations. In optimization, how large the trust radius should be is well-defined because in general gradients are available; if the function reduces significantly, it is good so they can be expanded, and the opposite if the function does not reduce as expected. In energy-based DFT methods where true gradients are available, trust regions are common; see, e.g., refs 16–20. However, energy minimization is fundamentally different from a fixed-point method without gradients.

While how to handle trust regions is well-established in optimization, it is not well-established for fixed-point problems where there can be problems with scaling and other uncertainties. What is a reasonable reduction to use to expand the trust radius, particularly when the problem can change discontinuously for something as simple as an isolated oxygen molecule (e.g., refs 21, 22) or when during convergence the system transitions from metallic to insulating? How should one scale components such as the kinetic energy density in meta-GGA methods, or other terms? In addition, gradients are not available, so most standard trust region methods such as dogleg<sup>23</sup> are not appropriate.

The intention of this article is to suggest a different approach to handle trust regions and control of parameters in fixed-point methods, focusing on DFT although the method should be general. Instead of contracting/enlarging these based upon a potentially biased metric, the approach calculates what should have been used for the most recent step and then uses this as a prediction of what to use in the current step—effectively calculating the optimum Polyak step size.<sup>14</sup> The advantage of this approach is that it removes the need for the user (or programmer) to decide what is good and what is not; instead this is determined automatically. This predictive approach is not specific to one particular method of choosing the candidate step for the next iteration, and it is completely general. There is no need to have one set of parameters for a metal, a different one for a surface, and a third set for an insulator.

The structure of this paper is as follows. In section 2 I will recap the basic structure of linear mixing methods, paying attention to points that have often been either glossed over or ignored in the literature, but are important for production code. Some aspects of this are specialized to all-electron codes such as Wien2k,<sup>24,25</sup> where they have been implemented<sup>21,22,26</sup> in the production version. However, they are general and much of this section is presented so that other codes can use appropriate scaling and trust region methods in their mixing algorithms, avoiding pitfalls. Some of these were described in earlier work;<sup>21,22,26</sup> some have been mentioned in the “Readme” notes with different Wien2k releases, but never described in detail; some exist in the literature outside DFT; significant parts of the analysis and algorithm are new.

I will then explain the predictive algorithm. This is followed first by examples comparing different algorithms commonly used in DFT calculations showing how the step sizes and trust regions change for different underlying algorithms. This is followed by a more detailed analysis comparing the predictive approach to one without most of the controls for 36 structures across a wide range of different systems.

While it would be extreme to claim that the predictive approach described herein will solve all problems, it appears to come close. Tests to date both by the author and by others indicate that it does not fail provided that the underlying physical problem is decently posed.

## 2. LINEAR MIXING METHODS

The purpose of this section is to recap the basics of mixing and to provide updates to aspects of the algorithms developed previously which have not been formally published although they have been present in the production code for Wien2k<sup>24,25</sup> for some years. At the same time the similarity and differences between the original Broyden<sup>27</sup> approach, the commonly used Anderson acceleration<sup>28–34</sup> which is sometimes called the Pulay method<sup>35–39</sup> and sometimes direct inversion in the iterative subspace (DIIS), as well as scaled multiscant,<sup>21,22</sup> and a hybrid method<sup>26</sup> will be described. The Anderson, Pulay, and DIIS methods are nominally the same, although they could be different in terms of how they are implemented numerically where details are often not published.

It is important to recognize that there are three components to a mixing algorithm, all of which need to be defined for reproducibility:

- (1) what the variables are, including scaling and preconditioning, as well as metrics
- (2) determination of the candidate step, including both the predicted and unpredicted components<sup>21,22</sup> as well as regularization
- (3) control of the step used, for instance, a line search or trust region method

It is common to find mixing methods defined just by stating the method of determining the candidate step, whether it is DIIS, Pulay, or other, and incomplete details. This is equivalent to stating that DFT calculations were performed without defining the functional used.

**2.1. Basic Formulation.** The formalism for mixing is well-established and will be described here for consistency based upon prior work of the author<sup>21,22,26</sup> in order to form the basis for the other analysis in this and other sections. A recent analysis with different opinions can be found in refs 40 and 41. Consider some density  $\rho(r, R)$  as a function of position  $r$  and a vector of  $N_a$  atom positions  $R = (R_1, R_2, \dots, R_m)$ . The solutions of the equations of Kohn–Sham density functional theory<sup>1,2</sup> can be written as

$$(H_0 + V)\phi_i = \epsilon_i\phi_i \quad (1)$$

$$F(\rho(r, R)) = \sum_i (1 + \exp((\epsilon_i - \mu)/kT))^{-1} |\phi_i|^2 \quad (2)$$

with eigenvectors (orbitals)  $\phi_i$  and eigenvalues (energies)  $\epsilon_i$ , where  $H_0$  is the noninteracting single-particle Hamiltonian,  $V$  is the effective local potential,  $\mu$  is the chemical potential,  $k$  is Boltzmann’s constant, and  $T$  is the temperature. In its simplest form we seek the fixed-point solution for a given density and atomic positions, i.e.

$$F(\rho(r, R)) - \rho(r, R) = D(\rho(r, R)) = 0 \quad (3)$$

where  $D(\rho(r, R))$  is the density residual which is analogous to the negative of the gradient in optimization. This form is specific to pure DFT calculations where the only active variables are the density; the forms for orbital-based density methods are slightly different and would involve a density matrix, an orbital potential, or include the wave functions as part of the active variables. These, as well as other terms such as constraints on atomic positions or linearization energies, can all be included without problem. In addition to the density condition, for a complete solution we seek the minimum of the total energy of the system,

i.e., for a total energy  $E(\rho(r,R))$  (including electronic entropy<sup>2</sup>) the derivative as a function of the atomic positions:

$$\partial E(\rho(r, R))/\partial R = g(\rho(r, R)) = 0 \quad (4)$$

Many algorithms solve eq 3 for fixed atomic positions and then change them using a minimization algorithm to move toward the solution of eq 4, reconverge the density, and iterate. Rather than solving these serially, they can be merged; i.e., we seek the fixed-point solution of

$$(D(\rho(r, R)), -g(\rho(r, R))) = \text{Res}(\rho(r, R)) = 0 \quad (5)$$

where  $\text{Res}(\rho(r, R))$  is a generalized residual vector. This is an approach initially suggested by Bendt and Zunger,<sup>42</sup> which has been implemented<sup>26</sup> within the Wien2k code. For a DFT code where the basis set is not atom position dependent, the gradients are the negative of the Hellmann–Feynman forces; otherwise additional corrections for the basis set are needed, commonly called Pulay corrections.<sup>43–46</sup> These are calculated with the Kohn–Sham density of  $F(\rho(r, R))$  whereas the Hellmann–Feynman calculations use  $\rho(r, R)$ . Hence the gradients  $g(\rho(r, R))$  in eqs 4 and 5 are not the Born–Oppenheimer surface gradients; they are instead vectors that share a common fixed point. (The Born–Oppenheimer surface is defined as the density as a function of  $R$  for which  $D(\rho(r, R)) = 0$ .)

Both the approach and notation used to solve eq 3 or 5 vary. Some algorithms deal with them as a least-squares problem while others use a Taylor series—herein I will use the latter. As has been demonstrated previously by several authors, the approaches are nominally identical (e.g., refs 10, 21, 22, 47–50) but differ in important implementation details. Dropping the  $(r, R)$  notation for brevity, and using  $\rho_n$  and  $R_n$  to describe the density and positions respectively for iteration  $n$  and  $\Delta(\rho_n, R_n)$  for a change in the density and positions, this suggests a standard Newton algorithm using a vector/matrix representation:

$$(\rho_{n+1}, R_{n+1}) = (\rho_n, R_n) - H_n \text{Res}_n + O(\text{Res}_n^2) \quad (6)$$

$$\text{Res}_{n+1} = \text{Res}_n - B_n \Delta(\rho_n, R_n) + O(\Delta(\rho_n, R_n)^2) \quad (7)$$

where  $H_n$  is the inverse of the Jacobian  $B_n$  for the change in density/pseudogradients with density/atomic positions. (The use of  $H$  here is conventional, unfortunately confusable with a Hamiltonian.) When the higher-order terms in the Taylor series are neglected, this is a linear model; these terms matter as will be described later. The computational cost of calculating the complete  $H_n$  or  $B_n$  is prohibitive, so instead they are approximated using a quasi-Newton method.

Introducing the new variables

$$y_{j,n} = \text{Res}_n - \text{Res}_j \quad (8)$$

$$s_{j,n} = (\rho_j, R_j) - (\rho_n, R_n) \quad (9)$$

and the matrices

$$S_n = [s_{n-k,n}, s_{n-k+1,n}, \dots, s_{n-1,n}]$$

and

$$Y_n = [y_{n-k,n}, y_{n-k+1,n}, \dots, y_{n-1,n}]$$

for  $k$  prior steps, we require that  $H_n$  and  $B_n$  satisfy the multisecant equations:

$$H_n Y_n = S_n \quad \text{or} \quad B_n S_n = Y_n \quad (10)$$

for which there are general rank-one solutions

$$H_n = \sigma_n [I - Y_n \text{Inv}(Y_n^T W) W^T] + \beta_n S_n \text{Inv}(Y_n^T W) W^T \quad (11)$$

$$B_n = \eta_n [I - S_n \text{Inv}(S_n^T V) V^T] + \gamma_n Y_n \text{Inv}(S_n^T V) V^T \quad (12)$$

with  $W$  and  $V$  any vector of size  $N_b + 3N_a$  (the size of the basis set for the density and related terms and the number of atomic position variables, respectively),  $\sigma_n$  and  $\eta_n$  are the *algorithm Greed*,  $\beta_n$  and  $\gamma_n$  will be referred to as *Damping terms* included to account for the neglect of the higher order terms in eqs 6 and 7, and “Inv” stands for inverse, which will be discussed in section 2.3. As discussed in earlier work,<sup>21,22,26</sup> the Greed determines how large a step the algorithm will take in the unpredicted direction and plays a critical role. A definition of “greedy algorithms” is appropriate:<sup>51</sup>

“A Greedy algorithm always makes the choice that looks best at the moment. That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution.”

Too much Greed can lead to instability depending upon the magnitude of the higher-order terms in eqs 6 and 7; too little and the algorithm may not converge or may converge only very slowly. (The latter case where steps are too small has analogies to the standard Wolfe and Armijo–Goldstein conditions.<sup>12</sup>)

In addition to the Greed and nonlinearity control via the Damping, one other term is useful to define. If  $\|Y^T Y\|_F \ll \|S^T S\|_F$ , where  $\|A\|_F$  is the Frobenius norm of the matrix  $A$ , the problem can be defined as “soft” in the spirit of soft phonon modes or similar in a phase transition, that is, those where large changes in variables lead to small changes in the residue, the magnitude of the residual vector. In contrast, if the inequality is in the other direction, the problem is “hard”. Hard problems may also be “stiff” in the sense used for differential equations, although this does not have to be the case.

The next step is then given by

$$(\rho_{n+1}, R_{n+1}) = (\rho_n, R_n) + H_n \text{Res}_n = (\rho_n, R_n) + \sigma_n U_n + \beta_n P_n \quad (13)$$

$$U_n = [I - Y_n \text{Inv}(Y_n^T W) W^T] \text{Res}_n; \\ P_n = S_n \text{Inv}(Y_n^T W) W^T \text{Res}_n \quad (14)$$

Here  $U_n$  is the component of the current residue about which no information is available, the unpredicted part;  $P_n$  satisfies the secant conditions of eq 10 and is the predicted part based upon previous steps. While in principle both  $S_n$  and  $Y_n$  can include all previous densities, in practice only a small number, between 6 and 16, are needed. Some DFT codes report a need to use much larger numbers which may be due to bad scaling or regularization; the number should not be significantly larger than the number of important eigenvectors of the Jacobian.

These equations deliberately do not specify the form of  $U$  or  $W$ ; I will return to these after some other key components are described.

**2.2. Units.** The formulation described in section 2.1 is deceptively simple. While it can be implemented as written, this is not optimal. Significant improvements can be achieved by ensuring that the variables are consistent with both the underlying mathematics and physics. This point was made in 1988 by Blügel,<sup>52</sup> but was missed at least for earlier versions of the Wien2k code; it is certainly omitted in some public domain codes.

One important issue is units. Equations 6–14 use standard vector/matrix representation, which implicitly leads to  $L^2$  metrics for the matrix elements of (for instance)  $Y_n^T W$ . (These products are nominally  $l^2$ , i.e., sums of the products of the terms, but strictly represent integrals for which the notation  $L^2$  is used.) These metrics have to be physically consistent. In particular, the contributions from different parts of the basis set have to be invariant to changes such as doubling of the unit cell, as well as reducing the symmetry of Fourier or local contributions, and also nondensity parameters such as density matrix or orbital potentials. They also have to be invariant to, for instance, increasing the volume of space without electrons in a surface calculation or for an isolated atom or molecule.

To illustrate this, consider an isolated atom with electrons inside the muffin tin and also plane waves in a cubic unit cell of volume  $V$ , where the volume is large, and ignore the plane wave pseudocharge inside the muffin tins (see the Appendix). The change during mixing in terms of electrons per unit volume is not independent of the volume, so these units are inappropriate. The unit of total electron charge to some power for the vector products in  $Y_n^T W$  is independent of the volume, so products which are proportional to this are viable. Noting that the number of plane waves to some maximum wavevector scales with the volume, whereas the densities scale inversely with the volume, the volume integral of the density squared (when the number of planes wave is included) scales as the total number of electrons squared. If one now changes the size of the muffin tins to transfer some density from those to the plane waves, it follows that the same units are required for the products within the muffin tins.

Hence plausible units of the vector products in  $Y_n^T W$  are the total charge squared or the volume integral of density squared summed over all variables. Tests indicate that the latter is better, consistent with the fact that we are dealing with “density” functional theory. Being specific, for a vector of variables  $A(r, R)$  the  $L^2$  squared should be the appropriate integral over volume, i.e.

$$(L^2)^2 = A^2 \equiv \int w(r) A(r, R) \cdot A(r, R) dV \quad (15)$$

with  $w(r)$  the scaling of input variables which includes the multiplicity of atoms and/or plane waves and other terms. The fixed-point problem is solved for the variables used which includes the pseudocharge (Appendix) and other terms. Scaling is done by preconditioning the vectors and matrices by multiplying them by  $\sqrt{w(r)}$  to later exploit BLAS/LAPACK calls, removing this scaling at the end.

Less clear is how to scale orbital terms such as density matrices or atomic positions (except for multiplicities which enter by a square root similar to eq 15). In the original implementations<sup>21,22,26</sup> an adjustable scale was used, but it now seems best to use fixed scales. Atomic positions are used in atomic units (au), the energy unit is the rydberg (Ry), and pseudoforces are in Ry/au, all divided by  $\sqrt{4\pi}$ , which is an adequate scaling.

**2.3. Regularization.** The equations in section 2.1 implicitly involve least-squares fits and implicitly or explicitly include a matrix inverse, which can lead to ill-conditioning; see, for instance, refs 13, 53, and 54. Consequently, all numerical mixing approaches involve some form of regularization, which may be machine precision truncation or another default in library codes. The majority of literature analyses do not state the regularization used.

Experience with the Wien2k code indicates that a conventional Tikhonov regularization<sup>55</sup> (aka regularized Moore–Penrose pseudoinverse) of the singular values of  $W^T Y$  by  $\lambda = \xi_{\max} \cdot 2 \times 10^{-4}$  is approximately correct, where  $\xi_{\max}$  is the largest singular value. Significantly smaller values such as  $\xi_{\max} \cdot 1 \times 10^{-8}$  can lead to instabilities; larger ones such as  $\xi_{\max} \cdot 1 \times 10^{-2}$  decrease the speed. Although methods to estimate the regularization exist, e.g., refs 56 and 57, they do not seem to work in tests. Being specific, with the standard single-value decomposition of a matrix  $A$  as

$$A = U \Sigma V^T \quad \text{with} \quad \Sigma_{ii} = \xi_i \quad (16)$$

the Inv operator in eq 11 is interpreted as

$$\text{Inv}(A) = V \Theta U^T \quad \text{with} \quad \Theta_{ii} = \xi_i / (\xi_i^2 + \lambda^2) \quad (17)$$

In the limit of no regularization, with  $W = V$ , eqs 11 and 12 are true inverses; with regularization, they are not and the term “dual” is more appropriate. An extended analysis of other types of regularization can be found in ref 31.

One useful metric is the effective rank, defined for  $M$  memories as

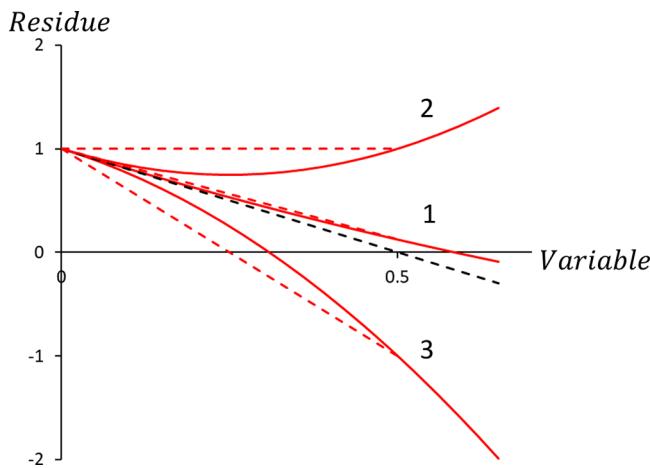
$$\text{Rank} = (1/M) \sum_i \xi_i^2 / (\xi_i^2 + \lambda^2) \quad (18)$$

Empirically, values of 0.7–0.9 indicate that the problem is fairly well-posed; both larger and smaller values indicate probable problems.

**2.4. Nonlinearities and Trust Region.** A standard phenomenon with DFT calculations, as mentioned earlier, is “charge sloshing”, where density appears to oscillate between different regions. It is important to understand the underlying mathematics.

All methods (except the tensor approach<sup>58–62</sup>) ignore the higher-order terms in eqs 6 and 7. Provided that these are small, solutions to eq 6 can be expected to converge superlinearly (see refs 10, 12, 13, 21, 22, and 26 and references therein). However, it is not appropriate to ignore the higher-order terms, particularly far from the solution. Because of these nonlinear terms, there are (at least) two mathematical sources of sloshing. The first is the magnitude of the unpredicted step in eq 11. Depending upon the problem (the inverse Jacobian along the unpredicted direction), this should be either small or large. The second source is the neglect of higher-order terms in the predicted step. The damping terms  $\beta_n$  and  $\gamma_n$  account for this, and they are both problem dependent and change as the variables converge to the fixed point.

The standard approach for quasi-Newton methods to handle such issues is to limit the step size, along both the predicted and unpredicted directions, using what is called a “trust region” (e.g., refs 10, 12, 14, 15). The trust region radius is controlled inside the code, and candidate steps are limited so that they are not larger than this radius. If the calculation is making good progress, the trust radius is expanded; if it is not, the trust radius is reduced. As illustrated in Figure 1, depending upon the sign and magnitude of the ignored terms, controlling nonlinearities will require a small trust radius or a large one—it is highly problem dependent. For curve 1 in Figure 1, the nonlinearities are not critical, but for curves 2 and 3, which have positive and negative second-order terms respectively, not only is their overshooting of the “best” value, but the simplex gradients<sup>63</sup> that would be used in future steps are poor. (The set of prior directions in the simplex gradient<sup>63</sup> has many of the properties of regular



**Figure 1.** Illustration of effect of nonlinearities, with the  $x$ -axis some variable and the  $y$ -axis the residue. The solid lines 1–3 schematically represent different types of problem. At the origin the gradient is shown in a black dashed line. Moving to 0.5 along the  $x$ -axis is good for small nonlinearities as in curve 1 but not so good for 2 and 3. In addition, the simplex gradients (red dashed lines) change drastically.

gradients,<sup>64</sup> but also differences as they are a limited set of directional secant slopes with finite step sizes.)

Note also from Figure 1 that the optimal step along the unknown direction ( $\sigma_n$  of eq 13) has to be large enough to come close to the minimum residual along this direction, but also not too large. The optimal step is often called the Polyak step,<sup>14</sup> a convention that will be used herein. With too small a value of the Greed the algorithm can “starve to death”. This shows up in calculations as will be seen in section 5.6; too small a value also often leads to premature convergence if the atomic positions are within the fixed-point method. The need for a step which is not too small has similarities to the standard Armijo–Goldstein inequality for sufficient decrease and similar step controls used in optimization.<sup>11,12</sup>

For the predicted component the trust region approach used in the Wien2k code is a standard method, solving the subproblem that minimizes the Lagrangian  $L$  for a step which is a linear combination ( $V$ ) of the prior steps, of size  $C$ :

$$L = \|B_n VS_n - \beta_n B_n P_n\|^2 - \psi(\|VS_n\|^2 - C^2) \quad (19)$$

Then, using for the next step

**Table 1. Different Methods of Forming the Predicted and Unpredicted Steps, in Terms of the Projection Matrix  $W$ , Whether the Iterations Include Differences between Adjacent Iterations (Sequential) or Are about the Current Point, Whether the Least-Squares Problem Is Solved by Overwriting Adjacent Points or Simultaneously for All, and Whether There Is Any Scaling<sup>a</sup>**

name	$W$ matrix	centering	form	rescaling	notes
good Broyden	$S$	sequential	overwriting	none	rare
bad Broyden	$Y$	sequential	overwriting	none	rare
DIIS	$Y$	current	matrix	none	common
MSEC	$Y$	current	matrix	diagonal	obsolete
MSGB	$S$	current	matrix	diagonal	noisy
MSR1	$Y + \alpha S$	current	matrix	diagonal	optimal
HYB1	$Y + \alpha S$	current	matrix	none	good
HYB2	$Y + \alpha S$	sequential	matrix	diagonal	good

<sup>a</sup>Results are reported here for the DIIS, MSEC, MSGB, and MSR1 algorithms. Two slightly different versions, HYB1 and HYB2, have also been tested, but since they are slightly worse than MSR1 they are not discussed further herein. For reference, larger values of regularization are needed for the sequential versions.

$$(\rho_{n+1}, R_{n+1}) = (\rho_n, R_n) + \sigma_n U_n + \|VS_n\|/\beta\|P_n\| + VS_n \quad (20)$$

reducing the size by changing  $\psi$  until the trust region conditions are obeyed. This is equivalent to a Levenberg–Marquardt<sup>65,66</sup> algorithm for the predicted step; the unpredicted component is scaled down by the reduction of the predicted component. Note that eq 20 solves for the best residue decrease, as against the step closest to a full step.

For the unpredicted component, earlier versions in Wien2k used an implicit trust approach where the size of the Greed  $\sigma_n$  was increased/decreased depending upon whether the algorithm was making progress. The total step and other metrics were also controlled by a trust radius which increased when the algorithm was making good progress by reducing the residue and decreased when the residue increased significantly. This worked, although there is a better approach as will be described later. For completeness, a slightly different form of the implicit trust region for the Greed has been implemented<sup>67</sup> in the Castep code.<sup>68</sup>

**2.5. Backtracking.** No matter how well protected a mixing (or optimization) trust region algorithm is, bad steps will always occur. While the next step may recover, a single bad step contains misleading information and will contaminate the simplex gradients. The standard approach is backtracking, that is, to return to the prior point and then reevaluate what should be done. The current algorithm uses one of two approaches depending upon the residue increase:

1. If the increase is larger than a number  $N_1$  (default 2.0), perform a quadratic fit along the prior step and then use this, discarding the most recent step (so it does not contaminate the simplex gradients of future iterations).

2. If the increase is smaller than  $N_1$  but larger than  $N_2$  (default 1.5), keep the current information but go back to the previous position and residue and recalculate after repeating the prediction for the trust radius and other terms (see section 3). Note that this temporarily increases the number of memories used by one.

Extensive tests confirmed that using a quadratic fit and ignoring the most recent value is better than keeping it. The two default values are approximately correct and are not dependent upon the type of system being analyzed.

**2.6. Choice of Algorithm and History Scaling.** The units, regularization, trust region control, and backtracking in sections 2.2–2.5 do not depend upon the specifics of how the matrices  $W$

and  $V$  in [eqs 11](#) and [12](#) are defined; the main algorithms are summarized in [Table 1](#).

The original algorithms suggested by Broyden<sup>27</sup> used a sequence of rank-one updates, rather than a matrix about the current point, satisfying the secant equations for a series of steps. Since the most recent information overwrites earlier information, this is a greedy algorithm. While this approach was used in earlier DFT codes,<sup>52,69,70</sup> it is no longer common with multisecant matrix methods dominant, i.e., the matrix secant equation, [eq 10](#).

More common in the DFT literature is to center the matrices on the current point. Different forms for  $W$  and  $V$  and slightly different methods of constructing the mathematics lead to variations in the method; all the common methods used for DFT can be described in this fashion as discussed previously.<sup>22</sup> In the mathematics literature the two most common are to take  $W = S_n$ , a multisecant form of Broyden's first method, often called "good Broyden", whereas taking  $W = Y_n$  is a multisecant form of his second method, sometimes called "bad Broyden". In the original paper by Broyden<sup>27</sup> where nonmultisecant methods were introduced, his first method worked but his second did not. A number of papers soon afterward reached the same conclusion, so "bad Broyden" was largely dismissed. However, more recent works in the mathematics literature have questioned whether the "bad" methods fails in all cases.

One common variant is what is called Anderson acceleration,<sup>28–34</sup> the Pulay method,<sup>35–39</sup> and sometimes direct inversion in the iterative subspace (DIIS)—they are all nominally the same, are all centered on the current point (not consecutive points), and use  $W = Y_n$  without any scaling. The conversion from [eq 10](#) to the multisecants of [eqs 11](#) and [12](#) are implicitly least-squares fits<sup>71</sup> over the prior history. One problem that is rarely mentioned is that least squares are biased toward the largest residue (or step), which is different from the sequential approach where relative scaling is not an issue.

To avoid this bias, the approach introduced previously<sup>22</sup> is to rescale by the diagonal values of  $\text{sqrt}(Y_n^T Y_n)$ . This removes the bias and also improves the conditioning of the inverse in the equation. This is the MSEC algorithm that has been used since 2008 in the Wien2k code. The same scaling can be applied to the multisecant version of "good Broyden", which I will refer to as MSGB. As discussed previously,<sup>22</sup> MSGB is a much more greedy algorithm.

A different approach introduced in 2013 in the Wien2k code<sup>24,25</sup> is the MSR1 algorithm,<sup>26</sup> which is inspired by the symmetric rank one (SR1) optimization algorithm.<sup>12,72–75</sup> It uses a linear combination of MSEC and MSGB, i.e.

$$W_n = V_n = Y_n + \alpha_n S_n \quad (21)$$

This algorithm is a member of the fixed-point Broyden family, and it will share the convergence properties of similar algorithms (e.g., refs [53, 54, 76–87](#)). In the limit  $\alpha_n = 0$  it is equivalent to Broyden's second method which most DFT codes use; in the limit  $\alpha_n \rightarrow \infty$  it becomes Broyden's first method.

To bound the value of  $\alpha_n$ , two limits are applied (not the same as those used originally):

(a) The first is an approach similar to that of Shanno and Phua<sup>88</sup> for initial scaling in quasi-Newton methods, namely an upper bound

$$\alpha_n \leq \max(1.0, \|Y^T Y\|_F / \|Y^T S\|_F) \quad (22)$$

For a problem where  $\|Y^T Y\|_F \gg \|Y^T S\|_F$  this includes some of the good Broyden behavior, but not too much. For a soft problem, it tends toward good Broyden.

(b) The second limit applied is the largest value of  $\alpha_n$  which is less than or equal to that in [eq 22](#), for which  $Y_n^T W_n$  has no negative eigenvalues, with the imaginary component less than the real part based upon extensive tests.<sup>89</sup> To explain this, consider some residual  $\text{Res}$  multiplying the inverse Jacobian of [eq 11](#). We can decompose this vector into three parts:

$$\text{Res} = Ya + Sb + Z \quad \text{with} \quad Z^T Y = Z^T S = 0 \quad (23)$$

and  $a$  and  $b$  vectors of the size of the number of memories used, whose precise values do not matter here. Looking at the parts gives the following:

1. The contribution of  $Z$  is independent of  $\alpha_n$ , so it can be ignored.

2. The component  $Ya$  satisfies the secant (least-squares) condition, so there is no restriction on it.

3. For the component  $Sb$  the action of  $H_n$  would be equivalent to a nonpositive definite matrix if there are negative eigenvalues. It is established (e.g., discussion in ref [26](#)) that the Jacobian and its inverse are related to the dielectric band structure<sup>90–96</sup> which has positive definite eigenvalues at its solution. Hence condition b enforces a minimization structure for the components beyond the secant condition, similar to what is used for optimization problems (e.g., ref [12](#)).

For scaling, each memory is scaled such that  $Y_n^T(Y_n + \alpha_n S_n)$  has unitary diagonal. This requires a simultaneous rescaling and iterative solution for  $\alpha_n$ , and it will improve the condition number of the inverse.

One point should be made about the difference between MSGB, MSEC (or DIIS), and MSR1. Consider the vector space spanned by nonzero predicted components from [eq 14](#), i.e., the set of vectors  $\Psi$  which satisfy  $W^T \Psi \neq 0$ . For MSGB this vector space is that of the prior  $S_n$ ; for MSEC and DIIS it is the vector space of the prior  $Y_n$ ; for MSR1 it is the sum of the two spaces. This implies that the predicted fraction of any residue component in MSR1 will in general be larger. Note that, from (b) above, the extra predictive component in MSR1 (relative to MSGB/MSEC/DIIS) is connected to a positive definite Jacobian which is appropriate near the solution.

As expected, both MSEC and DIIS can have difficulty with soft problems. They are equivalent to least-squares-minima solutions<sup>71</sup> with respect to a Frobenius norm in the space of the residual, that is, the matrix  $Y_n$ . Regularization will decrease the role of small eigenvalues, which are those which contain the information about soft modes. In contrast, MSGB solves a least-squares problem in the other space, so it implicitly includes more of the small eigenvalues in the space of the residual—sometimes too much, which is why it can be unstable. MSR1 is between the two extremes and hence does significantly better for problems with soft modes where MSEC and DIIS can fail.

### 3. PREDICTIVE MIXING

Mixing and fixed-point methods without some of the details discussed in [section 2](#) often work, but they can fail or require very small trust radii or Greed and may therefore be very slow. The target should be a method that always works, does not require user intervention, and is fast. Earlier versions of the code<sup>21,22,26</sup> used the  $L^2$  of the residue as a metric of whether to increase or decrease the trust radii and also change the Greed and predictive step scalings  $\sigma_n$  and  $\beta_n$ . This is the standard

approach (e.g., refs 10, 12). However, there are also fundamental problems with using the  $L^2$  of the residue:

1. It is unclear how valid an  $L^2$  metric of the residue is, since some of the variables may be poorly scaled (and/or ill-conditioned). This will vary with the problem and it is hard to handle except by undesirable adjustment on a case by case basis, e.g., one set of parameters for metals and another for insulators.

2. There are ambiguities as to the units to use—relative to the current residue or absolute?

3. During the iterations the problem often changes (electronic phase transitions<sup>26</sup>), and the trust radii may then change significantly. While it is possible (common) for the user to delete the prior history and restart (e.g., refs 10–12, 35, 87, 97, 98), I view this as undesirable for production code.

4. The parameters have to be fundamentally different for hard problems, where both the Greed and other parameters and the trust region have to be small, and soft problems, where they need to be large. It sometimes occurs that the problem is stiff far from the solution but soft near it; the program has to be able to adapt.

5. Commonly used components of trust region methods such as dogleg<sup>23</sup> and Cauchy steps<sup>12</sup> are not reliable as no gradient is available, and the residual may not be a good descent direction. In many cases the angle between the residue and step (e.g., eq 20) is large, sometimes close to 90°; in such cases the residue is not close to the steepest descent direction.

Is there a better method? Consider the general form for the next step as

$$H_n \text{Res}_n = \sigma_n U_n + \beta_n P_n \quad (24)$$

We can estimate both  $\sigma_n$  and  $\beta_n$  by calculating what they should have been for the *prior* step. That is, we minimize for  $\sigma_n$  and  $\beta_n$  using the *current* Jacobian, i.e., minimize

$$\|U_{n-1} - \sigma_n B_n U_{n-1}\|^2 \quad \text{and} \quad \|Y_{n-1}^P - \beta_n B_n P_{n-1}\|^2 \quad (25)$$

where  $Y_{n-1}^P$  is the projected component of the prior predicted residue. In some case the solutions for both  $\sigma_n$  and  $\beta_n$  are negative; only the absolute value is used.

Once values for both  $\sigma_n$  and  $\beta_n$  have been calculated, the maximum step size that should have been used for the last iteration,  $\|\sigma_n U_{n-1} + \beta_n P_{n-1}\|$ , is calculated and used to estimate the new trust radius, again with a running average to avoid severe fluctuations. For cases where atomic positions are simultaneously converged, a trust region control on the atomic step is used based upon this maximum step; that is, the maximum allowed atomic step corresponds to the maximum atomic motion in  $\|\sigma_n U_{n-1} + \beta_n P_{n-1}\|$ . These are used as the trust region radii described in section 2.4. The same approach is used for MSR1, DIIS, MSGB, and MSEC; the only difference is the matrices that are used.

To briefly expand, instead of increasing or decreasing the parameters  $\sigma_n$  and  $\beta_n$  depending upon whether the residue improves or not, they are changed based upon what they should have been for optimal performance in the last iteration. This is equivalent to estimating the Polyak step<sup>14</sup> involving the different parameters. These values are also used to determine the trust region radii of the total step and also the maximum atomic motion. This exploits the implicit assumption in all quasi-Newton methods that the prior history of steps is an adequate predictor. It should be noted that this has similarities to the approach used in tensor methods<sup>58,59</sup> where the previous information is used to estimate a higher-order term, the initial scaling approach of Shanno and Phua<sup>88</sup> in optimization, and

what is called the BB two-point step size gradient method.<sup>99–103</sup> However, it is not the same and appears to be a useful and new alternate to conventional trust region approaches.

One additional point deserves to be mentioned: should the Greed and total size be in absolute units or relative? To sidestep this, an average over the relative and absolute values appears reasonable in practice. The values tend to behave as relative, although this is determined by the solutions of eq 25 and is not specified in the algorithm.

#### 4. ALGORITHM SUMMARY

The same predictive algorithm is used for the different values of the matrix  $W$  (Table 1) for all of MSR1, DIIS, MSGB, and MSEC. (It has also been tested for the sequential methods, although in tests these are inferior so will not be discussed further.) The algorithm can be summarized as follows:

1. Collect prior histories and create the vectors  $Y_n$  and  $S_n$  for a user-specified maximum number of prior values as described in section 2.1, converting units as described in section 2.2. The default is eight prior values for just density values and 10 when atomic positions are simultaneously being optimized.

2. If the step was not good, either backtrack or recalculate as described in section 2.5.

3. Rescale as described in section 2.6, depending upon which algorithm is being used, and also regularize following section 2.3.

4. Calculate the Greed and Damping parameters as described in section 3 for the specific algorithm, and also calculate the maximum step and atomic movement which are used in step 6 as the trust region radii.

5. Calculate the candidate step using the Greed and Damping values for the specific algorithm.

6. If the step or atomic movement is too large, reduce it via a trust region approach as described in section 2.4.

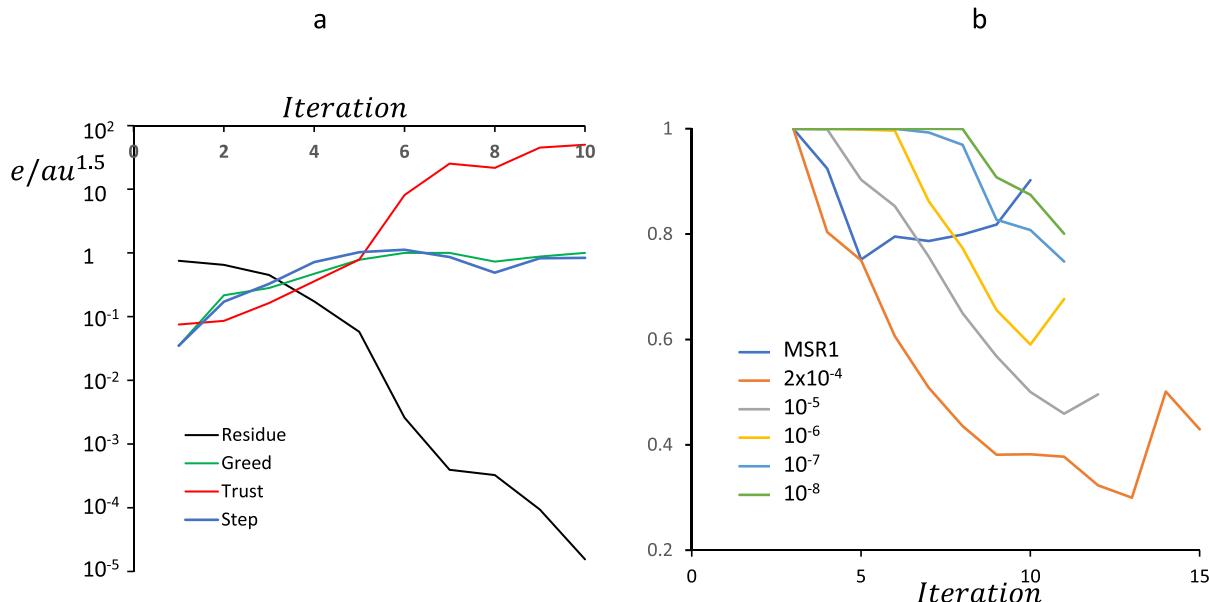
7. Unpack the variables and calculate using the new density, atomic positions, orbital potentials, and other parameters.

For the very first iteration a small Pratt-like step with a Greed of typically around 0.035 is used—the initial value is chosen based upon the initial residue as described previously.<sup>26</sup> (The exact value is not important so long as it is not too small or too large.) The predictions are used after the first Pratt-like step, with no rescaling performed until there are two or more memories.

The only adjustable parameters in the algorithm are

- (1) the number of memory steps, typically 8–12
- (2) the Greed in the first iteration, which just needs to be small enough
- (3) the maximum step in the first iteration, which rarely matters
- (4) the regularization, although the default value appears to be adequate
- (5) when to turn on backtracking

The results are weakly sensitive to the parameters in 1–5; none of them are adjusted in any of the results herein, with the exception of the regularization for DIIS in Figure 2b. The most important controls are determined by the code during the iterations, specifically the Greed, Damping, trust radius size, and maximum atomic movement. There is no need to adjust anything for different materials whether they are molecules, are insulators, are metals, contain transition metals, or are lanthanides or whether they involve spin-orbit, van der Waals terms, moving atoms, or Hubbard  $U$  as well as on-site or full hybrids (and combinations of these).



**Figure 2.** (a) Plot of various parameters and trust radii for bulk MgO using MSR1. (b) Comparison of effective rank using MSR1 and also DIIS with differing regularization as indicated from  $2 \times 10^{-4}$  (the default in all cases) down to  $10^{-8}$ . With MSR1 the convergence is rapid and the trust radius plays no role with a large predicted Greed. With DIIS the default regularization is too strong, and better performance is achieved with smaller values, although this can lead to instabilities.

## 5. RESULTS

In the next sections results will be presented for MSR1, MSEC, DIIS, and MSGB (see Table 1); Crystallographic Information File (CIF) data for all are included in the *Supporting Information*. In all cases the same regularization, predictive algorithms for the total step size, and Greed, Damping, and initial parameters were used, with only the form of  $W_n$  and preconditioning changing.

Results herein can in general be broken into five classes:

1. The first class (section 5.1) is simple problems such bulk unit cells, which typically converge in 10–12 iterations starting from a superimposition of isolated atom densities.

2. The second class (section 5.2) includes simple problems where the atomic positions and densities are simultaneously optimized, such as bulk Mg(OH)<sub>2</sub>, where all methods work. I will note that large values of the Greed can be needed for MSEC and DIIS to work for these, which the predictive approach automatically generates.

3. To compare with some recent all-electron calculations, slightly more complicated problems such as the Pd surface considered by Kim et al.<sup>104</sup> or the Fe and Cr semisurfaces considered by Winkelmann et al.<sup>105</sup> are included (section 5.3). These converge a little more slowly with MSR1, may not be very stable with MSGB, and do not always behave as well with DIIS. In general, these are quite sensitive to inappropriate scaling and to complications due to the pseudocharge (see the Appendix).

4. The fourth class (sections 5.4 and 5.5) includes harder problems such as those which are ill-conditioned or where the atomic positions and densities are simultaneously optimized for large changes in the positions; normally MSR1 significantly outperforms alternatives as will be shown for a WFe multilayer and a chemisorbed water case as two examples.

5. The fifth class (section 5.6) includes a set of 36 different structures comparing the MSR1 and DIIS algorithms with full predictive control to calculations with a fixed algorithm Greed, comparable to approaches in other codes. These range from structures reported to be hard to converge, to seven with more

than 100 atoms. They cover a wide range of different types of problems, including the following: two molecules; surfaces of metals including two with adsorbates; surfaces of small-gap semiconductors or larger-gap oxides; a number of problems with many different types of chemical environments; a number of more routine structures.

In a few cases with MSGB the predictive algorithm crashed due to ghost bands;<sup>106–108</sup> the default action in Wien2k is to stop when these are detected instead of backtracking, although this can be turned off. Beyond that the algorithm never diverged. Some of the fixed Greed calculations in section 5.6 diverged or crashed with ghost bands; they are not protected.

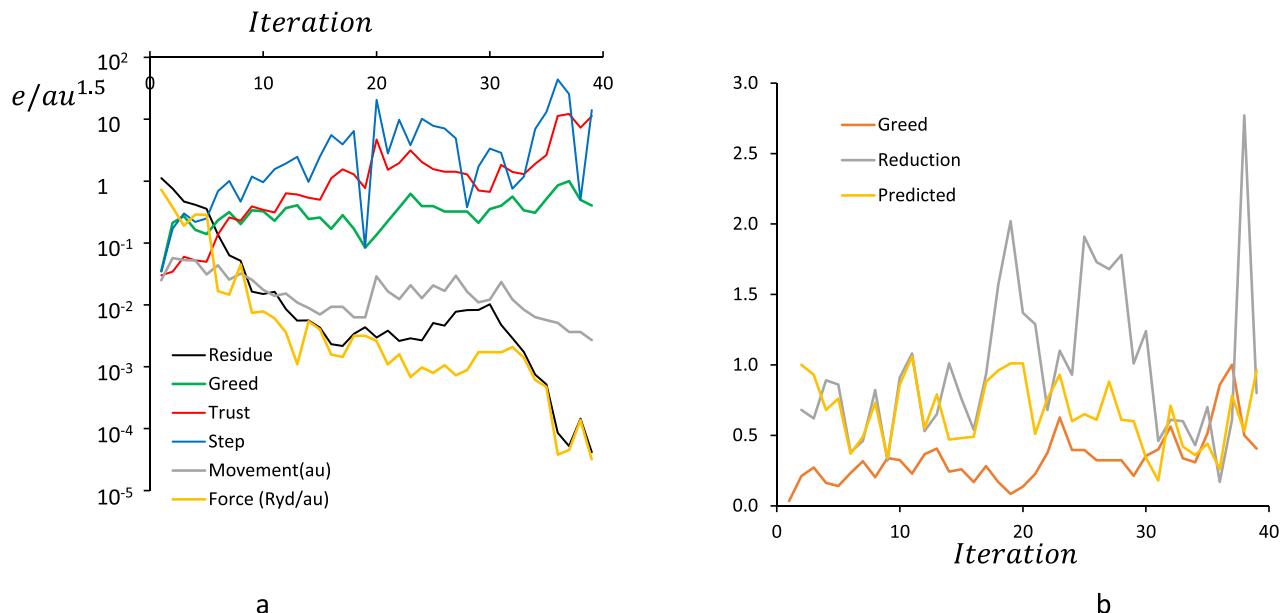
Metrics that will be shown in the following graphs are as follows:

- the  $L^2$  residue per atom (see section 2.2 for an analysis of the units)
- the Greed,  $\sigma_n$  as described earlier in eq 11, a dimensionless number
- Damping, the parameter  $\beta_n$  in eq 11, which is dimensionless
- the step trust region as described in section 3, in the same units as the residue
- how much reduction there was in the residue, as well as the predicted reduction, which is given by

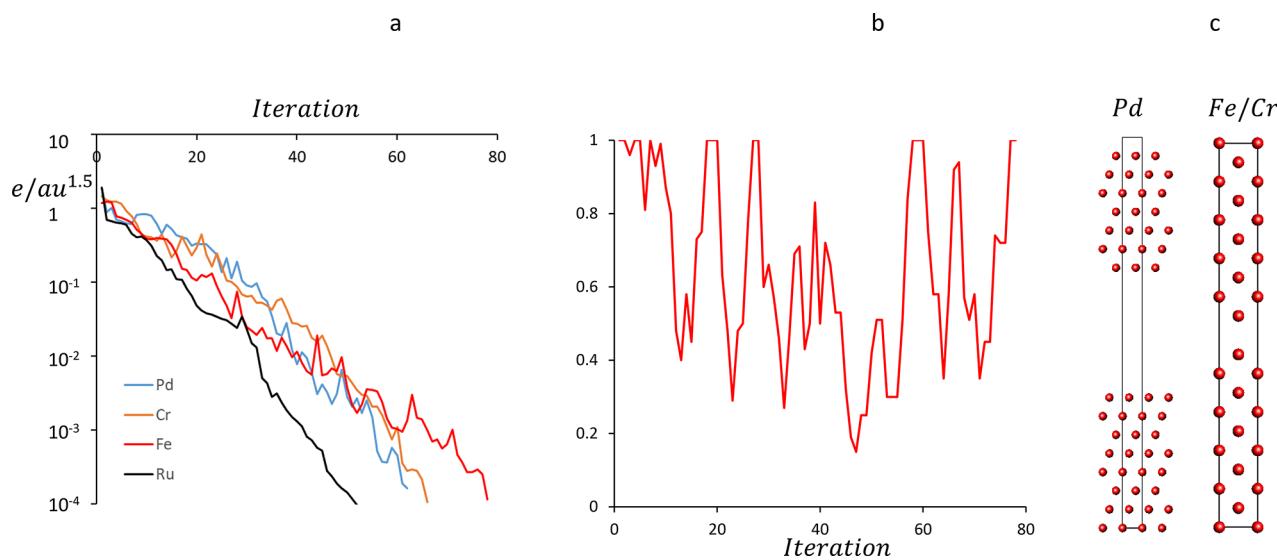
$$\text{Predicted} = \| [I - Y_n \text{ Inv}(Y_n^T W) W^T] \text{Res}_n \| / \| \text{Res}_n \| \quad (26)$$

- how large the step taken was compared to the residue, which is dimensionless
- the effective rank, which was defined in eq 19, which is dimensionless
- the root-mean-squared pseudoforce when relevant, in Ry/au
- for the last case, the total energy relative to the initial energy in Ry

Except for the case of bulk MgO in section 5.1, in all cases default parameters are used, where the Greed can range from  $10^{-3}$  to 1.0,  $1 < \beta_n < 0.15$ , and there is no upper bound on the maximum step trust radius or atomic movements, although



**Figure 3.** Plots of various parameters and trust radii for bulk  $\text{Mg}(\text{OH})_2$  with simultaneous density and atomic position convergence, without spin polarization as further described in the text. The metrics in (a) show how the  $L^2$  residue and the RMS force both decrease, while the various trust region parameters increase. As shown in (b), the predicted reduction is in general smaller than the actual, which is not unexpected and herein is not a reason to decrease the trust regions, both implicit and explicit.



**Figure 4.** (a) Plot of  $L^2$  as a function of iteration using MSR1 for a Pd surface, Cr and Fe ordered vacancies, and a distorted icosahedral Ru cluster with a nearby N atom as described in the text. (b) Variation of Damping for the Fe surface. (c) Structures for Pd and Fe/Cr. Damping plays an important role in stabilizing the fixed-point problem in many cases.

there were lower bounds on these of  $10^{-3}$  and  $10^{-4}$  au. For the examples in sections 5.2 and 5.5, the number of prior memories was the default 10; in all other cases where just the density was being converged, the number was the default 8.

**5.1. Bulk MgO.** A simple example is bulk MgO without spin polarization, for which MSR1 results are shown in Figure 2a. Technical parameters are the PBE functional,<sup>109</sup> muffin tin radius of 1.8 au, a  $6 \times 6 \times 6$   $k$ -mesh with Fermi–Dirac occupancies at room temperature, and a plane wave expansion via RKMAX of 7.0. The convergence of the RMS residue is linear, and after the first couple of iterations the trust radius increases to more than 10 and plays no part. This indicates that the algorithm believes that this is a very linear and well-conditioned problem. The Greed rapidly increases to 0.7–1.0,

and the step magnitude in each iteration also rapidly rises to 0.8–1.1 of the total residue.

Similar results are obtained with MSEC and MSGB whereas DIIS is slower with the same regularization. This is due to the bias toward the initial large residual, which led to an effective rank of  $\sim 0.33$  compared to  $\sim 0.9$  for MSR1. Changing the regularization altered this, increasing the effective rank and improving the convergence as shown in Figure 2b; the smaller the regularization, the larger is the effective rank. This illustrates the effect of the rescaling in MSR1, MSEC, and MSGB which helps in this case, although the relationship between the condition number of  $\text{Inv}(Y_n^T W)$  and the convergence rate does not appear to be simple.

**5.2. Bulk MgOH<sub>2</sub>, Both Density and Atomic Positions.** A second example is simultaneous convergence of densities and atomic positions for bulk Mg(OH)<sub>2</sub> without spin polarization, shown in Figure 3. Technical parameters are the PBE functional,<sup>109</sup> muffin tin radii of 1.6, 1.2, and 0.5 au for Mg, O, and H, respectively; a  $5 \times 5 \times 3$  *k*-mesh with Fermi–Dirac occupancies at room temperature; and a plane wave expansion via RKMAX of 2.5.

As shown in Figure 3a, the Greed rapidly rises to values of 0.2–0.4 and the bound on the total step size in electrons/atom rises from 0.03 to  $\sim$ 11, playing a relatively small role. Through most of the iterations the atoms are moving by about 0.01 au per iteration and the total residue is slowly converging; note that since this is an energy minimization the  $L^2$  residue is not a precise metric of the progress. In many cases, particularly toward the end, the change in the combined density/position residue is significantly larger than the  $L^2$  residue due to soft modes coupling the plane waves and the atomic positions.

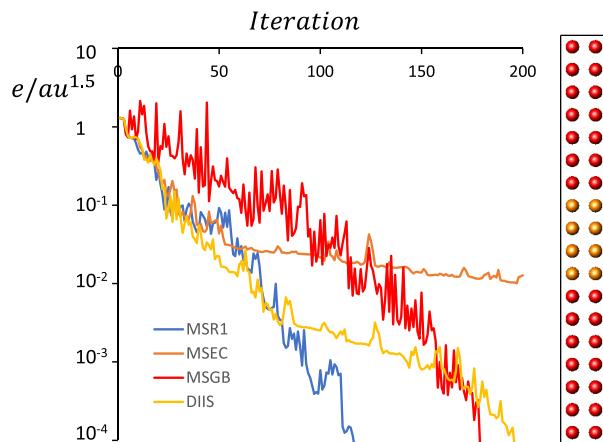
A different view of the convergence is shown in Figure 3b, which shows how much the  $L^2$  reduces in any given iteration as well as what is predicted from the linear model. Sometimes the step is not very good; on four occasions the algorithm backtracks and in one case the step is recalculated.

**5.3. Transition Metal Surfaces.** In order to compare with recent work, cases shown in Figure 4 are a nonmagnetic Pd surface with 15 atoms similar to what was used by Kim et al.,<sup>104</sup> a nonmagnetic Cr and a magnetic Fe surface, both with 19 atoms as recently used by Winkelmann et al.,<sup>105</sup> and a distorted, icosahedral 45 atom Ru cluster with a nearby N atom used by Woods et al.<sup>40,41</sup> In all cases the PBE functional<sup>109</sup> was used; Fermi–Dirac occupancy at room temperature and other technical parameters are as follows:

- Pd surface: RMT 2.5, RKMAX 7.0 with a  $10 \times 10 \times 1$  *k*-mesh
- Cr surface: RMT 2.25, RKMAX 8.0 with a  $10 \times 10 \times 1$  *k*-mesh
- Fe surface: RMT 2.10, RKMAX 7.0 with a  $10 \times 10 \times 1$  *k*-mesh
- Ru cluster: RMT 2.26 and 2.42 for Ru and N respectively with just the gamma-point

Comparing results with different codes is not simple. The results for the Cr and Fe surfaces are better than those reported by Winkelmann et al.<sup>105</sup> for their calculations without a Kerker preconditioner, an important caveat being that while the units used<sup>52</sup> appear to be very similar, they used a different final metric, dividing by the square root of the volume to give  $\text{au}^{-3}$ , and a convergence tolerance of  $10^{-6}$ . In terms of the results in Figure 4a, this is approximately equal to  $10^{-4}$ , with caveats when comparing codes. No attempt has been made to optimize the mixing parameters in Figure 5, whereas Winkelmann et al.<sup>105</sup> report only their best results. (The results reported by Winkelmann et al. with a Kerker preconditioner are better than those herein.)

The results for the Pd surface are comparable to those reported by Kim et al.<sup>104</sup> when they used a Kerker preconditioner, with the same caveats about differences of units and codes. Kim et al. were unable to get their version of MSEC to converge. However, as they admit, their “MSEC” does not contain key components of the original algorithm from 2008.<sup>22</sup> In this author’s opinion it is not surprising that an unscaled and unprotected algorithm does not converge, as Kim et al.<sup>104</sup> found.



**Figure 5.** Plot of  $L^2$  residue for a spin-polarized tungsten–iron multilayer using the different algorithms. The MSR1 algorithm converges well, MSGB is noisy but converges, DIIS is having trouble, and MSEC does not do well here although it was still improving when the calculation was stopped. The structure is shown on the right, with tungsten in gold and iron in red, and the unit cell is marked.

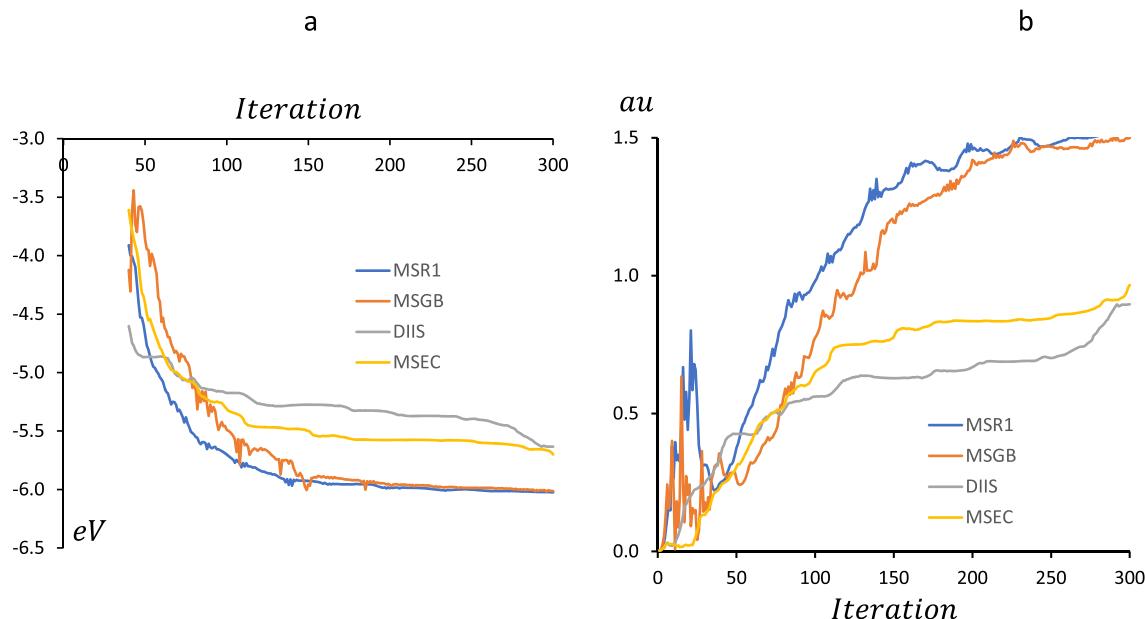
The Ru icosahedral cluster converges herein without any problem: it is a well-conditioned problem. In the work of Woods et al.<sup>40,41</sup> it stagnated near the solution after 500 iterations.<sup>67</sup> This was with a plane wave pseudopotential code (Castep<sup>68</sup>), and the reason why it did not converge merits investigation. Part of this may be because of how the authors controlled the Greed, which they do not describe in their paper beyond stating that they use something similar to the early MSEC method;<sup>21,22</sup> without this disclosure it is impossible to analyze their results further.

These are cases where the problem is somewhat nonlinear, so the Damping term plays an important role. Values of this for the iron surface are shown in Figure 4b; they are similar in the other cases. For these cases the Damping values averaged over the iterations were 0.66, 0.75, 0.65, and 0.91 for Pd, Cr, Fe surfaces, and the Ru cluster, consistent with how “noisy” the convergence is. In tests without the Damping the convergence is significantly worse in most cases, as expected.

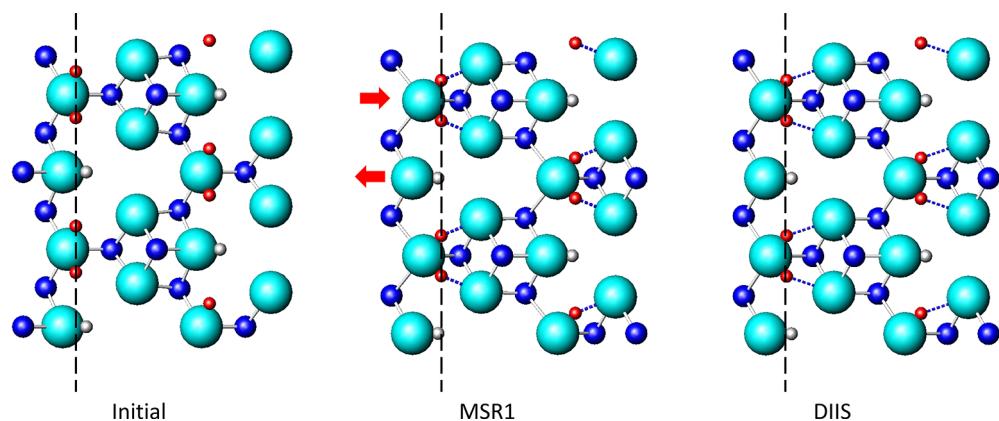
**5.4. Tungsten–Iron Multilayer.** A more complex case shown in Figure 5 is a multilayer containing 28 iron atoms and 8 tungsten atoms in a  $2.87 \times 4.06 \times 35.00 \text{ \AA}$  unit cell with *Cmma* symmetry, spin-polarized. Technical parameters are the PBE functional,<sup>109</sup> muffin tin radii of 2.0 and 2.31 au for Fe and W respectively, a  $7 \times 7 \times 1$  *k*-mesh with Fermi–Dirac occupancies at room temperature, and a plane wave expansion via RKMAX of 7.0. The iron atoms started with a magnetic moment of 3 (ferromagnetic), and the tungsten atoms started with 2. (The converged values have the tungsten nonmagnetic, with a total spin of 31 within the centered unit cell, which compares to 50 for the initial total spin.)

This is a less stable problem, which was hard to converge and often crashed with earlier versions of the mixer in Wien2k. The convergence of the  $L^2$ , shown in Figure 5 follows what one expects:

- MSR1 converges quite cleanly.
- MSGB does converge, but it is clearly noisy.
- DIIS is adequate as it focuses on the largest eigenvalues and will converge those.
- MSEC does not emphasize the largest eigenvalues and would eventually converge but is very slow in this case.



**Figure 6.** (a) Plot of energy in electronvolts versus iteration number of a larger MgO surface with chemisorbed water relative to the initial energy. As expected from the math, both MSGB and MSR1 handle the soft modes well while DIIS and MSEC are less effective. (b) Total distance in atomic units moved by hydrogen atom 3; see also Figure 7.



**Figure 7.** Plot of the top two layers for the MgO + H<sub>2</sub>O case, with hydrogen atom 3 (red) that is shown in Figure 6. The initial structure is on the left, MSR1 final structure is in the middle, and DIIS final structure is on the right. Dark blue is Mg, light blue is O, and gray or red is hydrogen. During the iterations hydrogen bonds are formed (blue, dashed lines), and there are shears indicated by the red arrows. The vertical lines are through atom 3, to help visualize the differences.

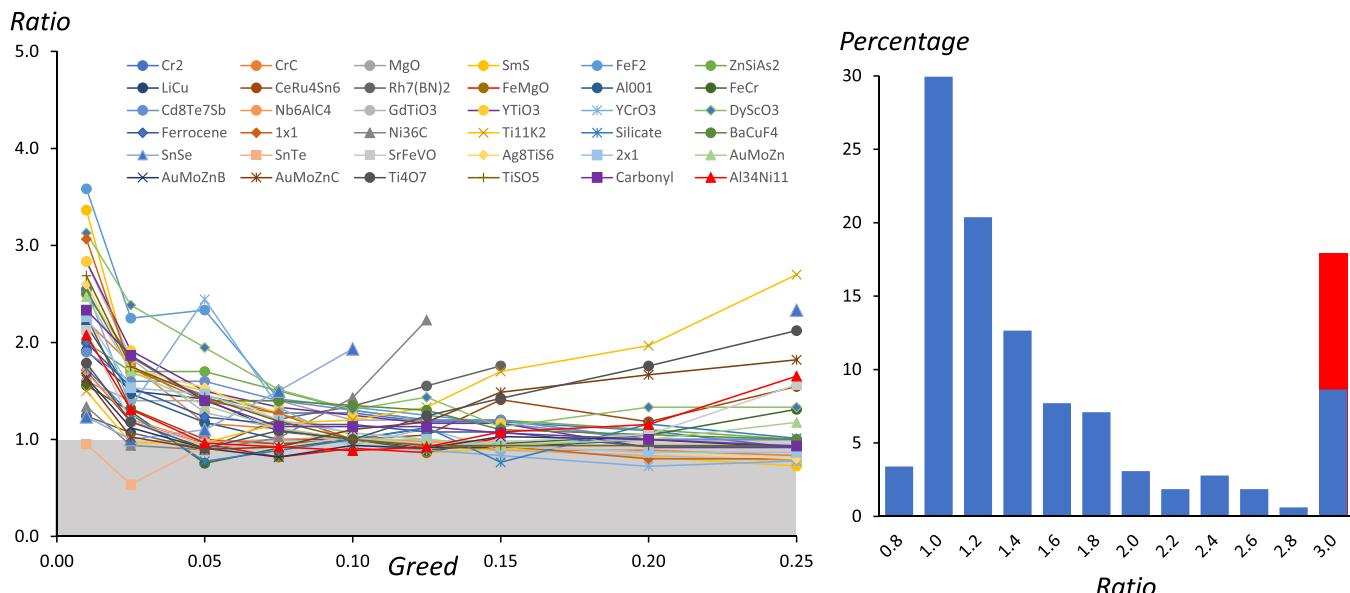
**5.5. MgO Surface with Hydroxide.** Another example is a non-spin-polarized calculation for a MgO surface with chemisorbed water and simultaneous optimization of the density and atomic positions, shown in Figure 6. Technical parameters are the PBE functional,<sup>109</sup> muffin tin radii of 1.63, 1.20, and 0.60 au for Mg, O, and H, respectively; a 3 × 1 × 1 k-mesh; and a plane wave expansion via RKMAX of 2.5. Due to rotation of the hydrogen atoms, this has significant soft modes.

The energy convergence, shown in Figure 6a is as expected: both MSR1 and MSGB do well with MSGB being somewhat noisy; MSEC and DIIS come close but are not as good with the total energy—they are less effective with the soft modes. This is clearer in Figure 6b, which shows the total movement in atomic units of the hydrogen atom (number 3) that moves most for the four different algorithms; both DIIS and MSEC underestimate the hydrogen movement. The difference is clearer in Figure 7, which shows the top of the initial structure on the left, the MSR1 result, and the DIIS result. A vertical dashed line is drawn

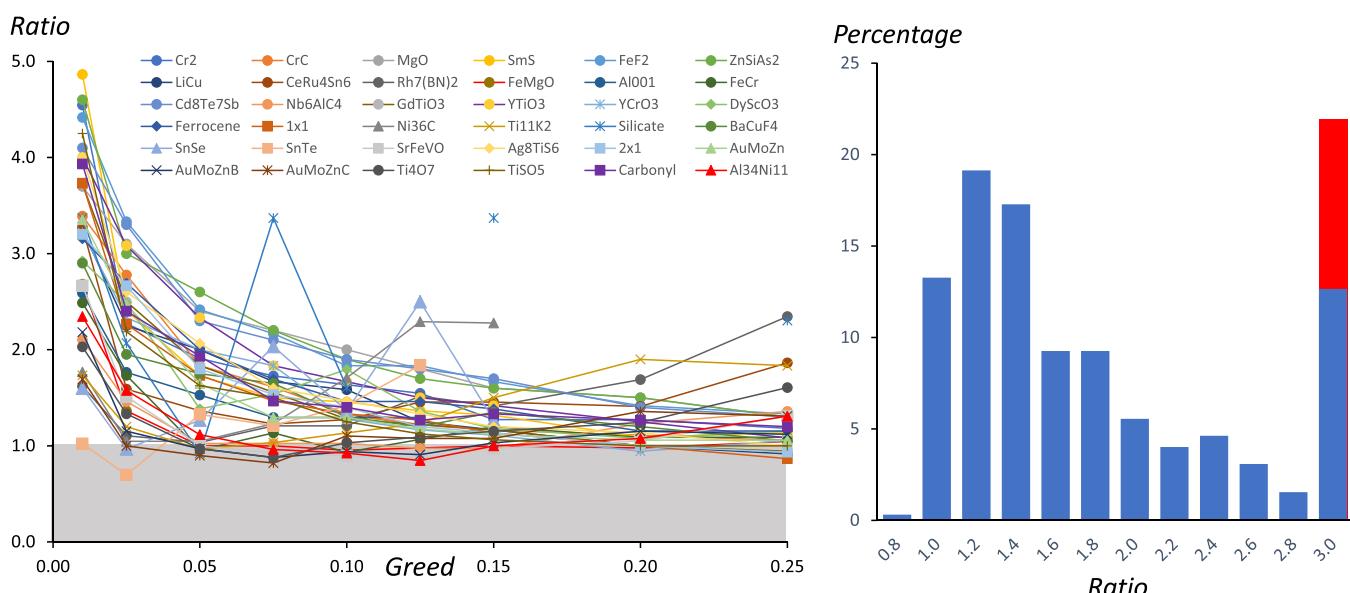
through atom H3 to guide the eye. Both MSR1 and MSGB are able to handle the soft mode associated with the shears indicated, allowing H3 to be more strongly hydrogen bonded.

**5.6. Comparison to Fixed Greed Cases.** To analyze the effect of the predictive controls, I will compare the results with the full algorithm to those where a fixed algorithm Greed is used; the Damping and trust region of section 2.4, the backtracking of section 2.5, and the predictions of section 3 have all been removed. The units of section 2.2 as well as the regularization of section 2.3 and the scaling of section 2.6 have been retained as removing these is inappropriate. These fixed Greed calculations are similar to what is in most other codes, where the user might have to change the Greed if they do not converge, or perform other adjustments.

A total of 36 different structures are included herein, ranging from simple to complex. These include insulators, metals, surfaces, a couple of on-site hybrids, a couple of molecules, and a number of large more complex structures. Very hard problems



**Figure 8.** Results for the ratio of the number of iterations to convergence using MSR1 without prediction and other controls as described in the text, to that with the MSR1 algorithm and prediction as well as controls. On the left the ratio versus Greed is shown, with the region where the ratio is less than 1 shaded for clarity. The numbers are in Table S1, and values that diverged have not been included. On the right a probability histogram is shown, where cases that did not converge are added to the 3.0 number. In red is how the results would change if nonconvergent cases for the Pd, Fe, Cr, and WFe cases were included. Atom positions are in the Supporting Information, with technical parameters embedded in the files.



**Figure 9.** Results for the ratio of the number of iterations to convergence using DIIS without prediction and other controls as described in the text, to that with the MSR1 algorithm and prediction as well as controls. On the left the ratio versus Greed is shown, with the region where the ratio is less than 1 shaded for clarity. The numbers are in Table S1, and values that diverged have not been included. On the right a probability histogram is shown, where cases that did not converge are added to the 3.0 number. In red is how the results would change if nonconvergent cases for the Pd, Fe, Cr, and WFe cases were included. Atom positions are in the Supporting Information, with technical parameters embedded in the files.

such as the WFe described earlier are not included, as they diverged for the range of fixed values used herein. (Many other cases also diverge with fixed Greed unless it is very small.) The intent here is to test more complex cases including ones with a significant number of different types of atoms and chemical environments. The structures can be put into four groups:

1. The first group is structures such as Cr<sub>2</sub> and CrC which were discussed by Daniels and Scuseria<sup>110</sup> as being hard and also a number of structures that Woods et al.<sup>41</sup> consider to be hard, which can be found in their repository.

2. The second group includes a number of relatively large and complex structures, including an iron vanadate, SrFe<sub>3</sub>V<sub>18</sub>O<sub>38</sub>,<sup>112</sup> a large unit cell titanyl sulfate;<sup>113</sup> a complex silicate,<sup>114</sup> Ba<sub>2</sub>FeCe<sub>2</sub>Ti<sub>2</sub>Si<sub>8</sub>O<sub>26</sub>; a large unit cell intermetallic,<sup>115</sup> Au<sub>10</sub>Mo<sub>4</sub>Zn<sub>89</sub>, plus two artificial substitutional derivatives of it — Au<sub>10</sub>Ga<sub>4</sub>Mn<sub>4</sub>O<sub>12</sub>Ru<sub>4</sub>V<sub>10</sub>Zn<sub>49</sub> and Ag<sub>6</sub>AlAu<sub>4</sub>Cr<sub>6</sub>Cu<sub>12</sub>Fe<sub>12</sub>Ga<sub>4</sub>Mo<sub>4</sub>Nb<sub>12</sub>Ni<sub>4</sub>Os<sub>12</sub>Ru<sub>4</sub>V<sub>10</sub>Zn<sub>12</sub>; a carbonyl, Pb(Mn(CO)<sub>5</sub>)<sub>3</sub>(AlCl<sub>4</sub>);<sup>116</sup> a small band gap semiconducting silver ion conductor, Ag<sub>8</sub>TiS<sub>6</sub>,<sup>117</sup> a Magnelli phase superstructure;<sup>118</sup> and a quasicrystal approximant, Al<sub>34</sub>Ni<sub>11</sub>.<sup>119</sup>

3. The third group is a number of different surface structures, including two experimentally relevant structures for  $\text{SrTiO}_3(110)$  surfaces<sup>120</sup> and one for BN on Rh;<sup>121</sup> two semiconductor small band gap (001) surfaces for SnSe and SnTe; an Al(001) surface; and a  $\text{BaCuF}_4(001)$  as well as carbon on a Ni(001) surface from Woods et al.<sup>41</sup>

4. The fourth group is more conventional cases, including structures from the examples that come with the Wien2k code and a few others collected from various other sources over the past decade.

In many cases fixed Greed and DIIS failed to converge simultaneous atom and density problems, so all of the calculations are for fixed positions.

The average numbers of iterations to converge for these test cases with the predictive algorithm were 26 for MSR1, 29 for MSEC, 31 for MSGB, and 32 for DIIS; the maximum numbers in the same order were 89, 118, 119, and 123 (see Table S1). None of the systems was problematic for the predictive approach, although it was not uncommon for the algorithm to backtrack near the start until enough good history steps were present for stable operation; this was clearest for the small band gap surfaces of SnSe and SnTe which are very susceptible to sloshing-type instabilities. For SnSe and SnTe the default stopping behavior of Wien2k when ghost bands are detected was turned off for these, so the backtracking could operate at the start. A number of the fixed-value calculations either diverged or did not converge within a reasonable number of iterations (typically 200).

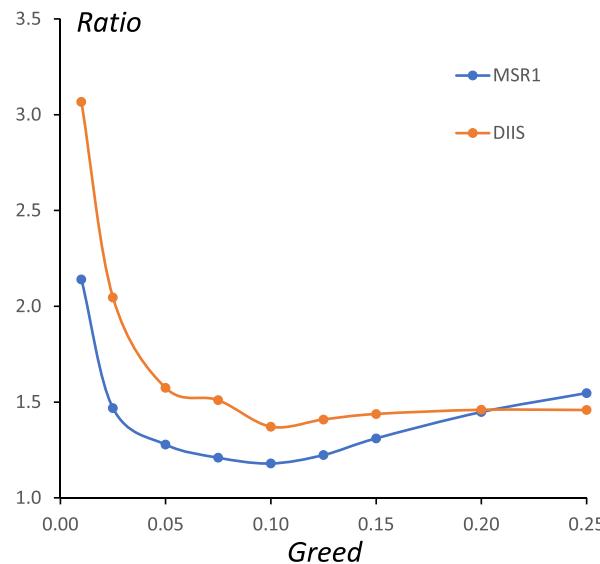
The results relative to the full algorithm MSR1 are shown in Figure 8 for fixed-value MSR1 and Figure 9 for fixed-value DIIS, with additional information in Table S1. All structures are included as CIF files in the Supporting Information, and technical parameters for the calculations are embedded in the CIF files. In both cases the ratios for different Greeds are plotted on the left and a histogram of values appears on the right. Cases where the ratio was larger than 3 have been added to the histograms, as well as those that did not converge with the fixed Greed. The change if the fixed Greed results for the Pd, Fe, Cr, and WFe cases were included is shown in red, as these only converge for very small fixed Greed.

Figure 10 shows the average ratio as a function of the Greed for MSR1 and DIIS, where the cases that did not converge have been included with 3 times the number of iterations. The predictive algorithm on average does better than the fixed ones, particularly as it did not crash or diverge.

The calculations without the controls fall into several classes:

1. Cases where larger Greed converges faster is one class. As mentioned earlier, this depends upon the magnitude of the nonlinear component.
2. Another class is cases where an intermediate Greed is better, and they may diverge for larger values
3. A third class is cases where only a very small Greed works, for instance the Pd, Fe, Cr, and WFe cases shown earlier (not included in Table S1 or the graphs in Figures 8 and 9). This is more often the case for metallic or small-gap surfaces.

The number of iterations to convergence did not scale in a simple fashion with the number of unique or total atoms; as described previously,<sup>26</sup> it is known<sup>11,12,122</sup> that the convergence of quasi-Newton methods depends upon the number and width of eigenvalue clusters. For instance, simple SmS with two atoms in the unit cell and active 4f electrons converged worse than a 2



**Figure 10.** Average ratio relative to the predictive approach as a function of the Greed, where nonconvergence has been included as a value of 3.0.

$\times 1$  (110)  $\text{SrTiO}_3$  surface which had 74 atoms, or the intermetallic  $\text{Au}_{10}\text{Mo}_4\text{Zn}_{89}$  which has 103 atoms. In certain cases the convergence did not smoothly vary with the fixed Greed, which is an indicator of ill conditioning, for instance, the silicate. There was no major indication that metal systems (i.e., those with partial occupancy at the Fermi level) converged worse than insulators, and there was no need to use large smearing terms.

For a few of the systems analyzed specific fixed Greed values were a little faster than the predictive algorithm. The prediction is not perfect as it is extrapolating past performance to the future. I will argue that the fact that the algorithm achieves close to optimal speed across a significant range of material system and problems of differing stability and nonlinearity without any user intervention makes it useful.

## 6. DISCUSSION

The aim herein is a fast algorithm that will always converge, which works in all cases without adjustment. The method has to be applicable to not just density but also combined density and atoms, orbital potentials, meta-GGAs, and anything else. (The code in Wien2k can also handle linearization energies, and an experimental option will “mix” constraints on atomic positions.) While nothing can save a very badly posed problem, the mixing algorithm has to be able to handle both simple and complicated problems, ones which are well-conditioned as well as those which are ill-conditioned. The algorithm should be based upon a combination of solid mathematics and physics/chemistry, as otherwise it is unlikely that it will be generally applicable.

The algorithm herein comes very close to achieving this. If a bad initial density is chosen, it is possible that the iterations can crash at the start with ghost bands,<sup>106–108</sup> sometimes in the first iteration. Without prior history it is impossible to safeguard the algorithm at the start; hence the first Pratt-like step is normally small, being cautious.

The different variants behave as one would expect from the mathematics. In general MSR1 is the best and then MSEC, MSGB, and DIIS (all with the predictive method). MSGB is greedier, which also means that it is noisier, so it is more likely to

fail although it did not for any of the test cases herein. When atomic positions are included, MSR1 is significantly better, with MSGB second; both MSEC and DIIS can fail to properly converge the atomic positions since they do not handle soft modes well.

It should be noted that only the fixed Greed calculations of section 5.6 are somewhat “standard” calculations, although since they include proper units (section 2.2) and regularization (section 2.3) they are not the same as those elsewhere in the DFT literature. My experience is that incorrect units and regularization do not matter for well-conditioned problems, but they do matter for ill-conditioned ones.

It is worth reiterating some points that have been mentioned earlier:

1. No method is guaranteed to converge unless it is safeguarded by either a line search or a trust region control, either implicit or explicit.

2. The different methods of determining the candidate step, i.e., Anderson, Pulay, Broyden, or other, are fundamentally the same except for hidden scalings and regularization that are often incompletely documented. In many cases they all converge similarly (within stochastic variations) so long as the controlling code is appropriate. I suspect that large differences in the convergence as a function of which algorithm is used as reported in the literature are in many cases due to buried parameters.

3. Frequently poor convergence is due to a poorly posed physical problem. Nothing says that every DFT problem should have a simple, unique fixed-point solution, and many certainly have multiple local fixed points. Empirically the radius of convergence of DFT problems depends upon how well-posed they are in terms of the underlying physics and chemistry. While this is hard to prove rigorously, it is common in practice; it is also consistent with having negative or complex dielectric eigenvalues far from the fixed point.

4. Phenomena such as charge sloshing are physical manifestations of nonlinearities of the mathematics, ill-conditioning, or perhaps inappropriate algorithms or problems.

5. Large Greed (mixing factors) is not necessarily good or desirable.

6. Small Greed can be as much of a problem as large Greed.

7. With limited memory methods for optimization, only a few memories are typically used; more is slower—the same herein. The effect of numerical and algorithmic noise will scale inversely with the square root of the number of memories used. Hence, if a large number is needed, this suggests possible problems.

8. While simple implementations of mixing methods will probably work, one can do much better with attention to the scaling, units, and the underlying mathematics.

The algorithm described herein is more complicated than others in the literature. However, it is not more complicated than sophisticated optimization or nonlinear-least-squares algorithms (e.g., refs 10, 12). Generating code to solve the trust region subproblem and the predicted parameters is straightforward, and once done extension to other trust controls is trivial. Additional trust controls are available in Wien2k, for instance on the maximum change of any plane wave component; however, tests indicate that they are redundant and degrade performance. To go between the different algorithms only requires changes of the  $W$  matrix. To what extent MSR1 and also simultaneous density and atom movements will work with pseudopotential codes is not yet clear since it has never, to my knowledge, been tested with modern fixed-point methods—a topic for future work. Analyzing both MSR1 and the predictive

approach for other fixed-point problems is also a topic for future work.

Values for parameters such as the regularization may be code dependent; they should be tested for other codes. (As mentioned earlier, the dependence of convergence in Wien2k on these values is weak.) Various parts of DFT codes involve numerical differentiations, integrations, and quadratures. These, and other parts of the underlying DFT code, can lead to ill conditioning and instabilities; details are rarely if ever reported. This importance of algorithm “noise” and instability was previously mentioned by the author,<sup>26</sup> is well-known in optimization,<sup>14</sup> and has been analyzed in a different context by Toth et al.<sup>123</sup> Over the past few years attention has been paid to improving these in the Wien2k code, which has contributed to improved speed in terms of the number of iterations to convergence.

As a specific example, numerical differentiation is often performed using a polynomial fit over values. As the degree of the polynomial increases, the fit becomes more accurate. However, the conditioning becomes worse. To avoid this, all one-dimensional numerical differentiations in Wien2k have been converted to use a cubic spline fit, which is much better conditioned. A minor loss of accuracy (at the  $10^{-8}$  level or less) is more than compensated for by fewer iterations to convergence and more general stability, particularly since there are always other hidden approximations in DFT codes. Particularly when densities and atomic positions are simultaneously converged, noise due to ill-conditioned internal algorithms or numerical truncations can play a significant role in degrading performance.

It is not hard to combine the methods described herein with the various forms of Kerker estimation of the Jacobian or its inverse (e.g., refs 50, 92, 104, 105, 124–130). There are clear indications that this improves convergence, although there is more to this than appears to have been discussed to date. Two different approaches can be used. The first is to estimate the initial matrices, rather than using a unitary matrix, i.e., change eqs 11 and 13 to

$$H_n = \sigma_n H_n^{\text{Est}} [I - Y_n \text{Inv}(Y_n^T W) W^T] + \beta_n S_n \text{Inv}(Y_n^T W) W^T \quad (27)$$

and

$$(\rho_{n+1}, R_{n+1}) = (\rho_n, R_n) + H_n \text{Res}_n = \sigma_n H_n^{\text{Est}} U_n + \beta_n P_n \quad (28)$$

where  $H_n^{\text{Est}}$  is the estimate. Note that this changes the unpredicted component but has no effect on the predicted component. Better estimating of the unpredicted step should always improve convergence and will not change the trust region algorithms significantly, although it will change the contribution of the unpredicted component to the  $L^2$  metric. In principle the initial estimate for the density could be combined with estimation of the Hessian from a simple spring model (e.g., ref 131) when refining atom positions at the same time.

The other approach redefines the secant equation (10) as

$$H_n Y_n = \Phi_n^{-1} H'_n \Psi_n Y_n = S_n \quad (29)$$

where  $\Phi_n^{-1}$  and  $\Psi_n$  are matrices (or operators) chose such that  $H'_n$  is better conditioned, ideally diagonal. This leads to the change of variables to new  $Y'_n$  and  $S'_n$  given by

$$Y'_n = \Psi_n Y_n \quad \text{and} \quad S'_n = \Phi_n S_n \quad (30)$$

with  $Y'_n$  and  $S'_n$  then used to replace  $Y_n$  and  $S_n$  in the equations/algorithms. This changes not just the unpredicted component but also the predicted component, the relative fraction of previous steps in the MSR1 algorithm, and the effect of regularization. Depending upon how  $\Phi_n^{-1}$  and  $\Psi_n$  are chosen, the algorithm could change from a version of “bad Broyden” to Greedy “good Broyden” or a hybrid similar to MSR1.

Note that, by using a physics-based definition in eq 30 of the secant condition, one automatically is converting from possibly somewhat arbitrary units to ones which are correctly scaled in the sense of section 2.2. To what extent Kerker conditioning in all-electron and other codes remedies inappropriate scaling is an open issue that merits attention.

Methods for Kerker estimation have recently been reported for all-electron codes<sup>104,105</sup> and are compared in Figure 4 to the predictive algorithm without such conditioning. At the time of writing no Kerker estimation code has been constructed for the Wien2k code, so it is not plausible to test the different approaches in detail, particularly for hard problems where trust region control is essential. I am not aware of testing with different forms of the preconditioning of eqs 26 and 27, and sometimes details are missing in publications. More detailed analysis is an issue for future work.

There may be other improvements that can be made, in both the algorithm and how the prediction is done. One example is the tensor approach<sup>58–62</sup> that, similar to the approach here, uses prior steps to estimate the higher-order diagonal terms, which is comparable to using an initial estimation.

In summary, the predictive approach combines speed and robustness and is quite general for the MSR1, MSGB, MSEC, and DIIS methods of calculating the candidate predicted and unpredicted components, although in general MSR1 is better probably because the vector space is larger and it also has in-built positive components which are appropriate for DFT problems. It would be premature to claim that MSR1 and the predictive approach is the best for all possible problems, but it appears to come very close to this. Independent of MSR1, the predictive approach appears to be a powerful addition to available methods.

## APPENDIX: PSEUDOCHARGE

There is a specific issue with all-electron muffin tin methods that merits description, as it does not appear to have been analyzed in the literature. With these methods the basis sets are spherical harmonics within the muffin tins, and with plane waves outside. While the plane waves are formally only involved outside the muffin tins, this is only possible with an infinite basis set. Instead, with finite wavevectors there is density from the plane waves inside the spheres. I call this plane wave density the “pseudocharge”, and in practice the pseudocharge integrated over the spheres is normally comparable to or larger than the plane wave charge integrated outside the spheres.

While in principle one could set this interstitial density to zero inside the spheres, this would lead to other problems including Gibbs phenomena at the muffin tin boundaries due to finite Fourier series and inaccuracies with numerical derivatives when calculating exchange–correlation potentials. The approach in Wien2k is to retain this pseudocharge in the charge density and avoid setting it to zero inside the spheres. This leads to a continuous, well-behaved plane wave density in both the interstitial region and within the muffin tins. (The contributions of the pseudocharge to the potential are not included when the Coulomb and exchange–correlation potentials are calculated;

the Coulomb contribution is removed using the multipole contributions<sup>132</sup> of the pseudocharge and the exchange–correlation contribution is removed by setting the components inside the spheres to zero.)

When mixing is being performed, the convergence of the pseudocharge inside the muffin tins is included. With an infinite basis set and exact numerics, the components of the Jacobian associated with the pseudocharge during mixing are zero. However, this is not the case with a finite basis set. Since the pseudocharge plays a role in numerical differentiations when evaluating gradient and higher derivatives in functionals near the muffin tin boundary, as well as other contributions, it has a nonzero coupling to the Jacobian.

The default starting density is a summation of single atom densities with an extrapolation of the plane waves within the muffin tins. This extrapolation is in general different from what is present when the iterations are converged. Unless care is taken, the pseudocharge can drive the mixing; densities where the pseudocharge is converged but the “real” density is not can easily occur and lead to problems. The limitations on the total step control this. An additional numerical approach is taken in the first iteration, projecting the new density within the muffin tins and using the component of this which is close to the origin in the next iteration. This avoids numerical issues with derivative discontinuities or similar near the muffin tin boundary while compensating for most of the pseudocharge which, with the initial atomic densities, is near the nucleus. There may be better approaches for generating the initial density to further reduce these problems, a topic for future work.

## ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jctc.1c00630>.

Number of iterations to convergence for the 36 systems described in section 5.6, including the numbers for fixed greed using MSR1 and DIIS and those with the predictive algorithm for MSR1, MSEC, MSGB, and DIIS (PDF)  
CIF files for all of the systems used for the calculations, with technical parameters for the 36 systems in section 5.6 embedded in the CIF files (ZIP)

## AUTHOR INFORMATION

### Corresponding Author

L. D. Marks – Department of Materials Science and Engineering, Northwestern University, Evanston, Illinois 60201, United States;  [0000-0002-6659-2016](https://orcid.org/0000-0002-6659-2016); Email: [lmarks@northwestern.edu](mailto:lmarks@northwestern.edu)

Complete contact information is available at: <https://pubs.acs.org/10.1021/acs.jctc.1c00630>

### Notes

The author declares no competing financial interest.

## ACKNOWLEDGMENTS

I would like to thank Lyudmila Dobysheva, Bouafia Hamza, Luis Ogando, and Yundi Quan for testing the algorithm herein. I would also like to thank Robert Schnabel for some background information, and both Xavier Rocquefelte and Fabien Tran for important comments on the mixer and the manuscript. I am indebted to Peter Blaha for many comments over the years including extensive testing of this and other versions of the mixer

in Wien2k. The crystal ball image in the abstract graphic is an open source image under the Unsplash license (<https://unsplash.com/license>) from Virgil Cayasa, taken from <https://unsplash.com/photos/UbvcYKScirk>. This work was supported by the National Science Foundation (NSF) under Grant DMR-1507101.

## ■ REFERENCES

- (1) Kohn, W.; Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, *140*, A1133–A1138.
- (2) Mermin, N. D. Thermal Properties of the Inhomogeneous Electron Gas. *Phys. Rev.* **1965**, *137*, A1441–A1443.
- (3) Anantharaman, A.; Cancès, E. Existence of minimizers for Kohn-Sham models in quantum chemistry. *Ann. I. H. Poincaré-An.* **2009**, *26*, 2425–2455.
- (4) Argaez, C.; Melgaard, M. Existence of a Minimizer for the Quasi-Relativistic Kohn-Sham Model. *Electron. J. Differ. Equations* **2012**, *2012* (18), 1–20.
- (5) Wagner, L. O.; Stoudenmire, E. M.; Burke, K.; White, S. R. Guaranteed convergence of the Kohn-Sham equations. *Phys. Rev. Lett.* **2013**, *111*, 093003.
- (6) Hantsch, F. Existence of Minimizers in Restricted Hartree-Fock Theory. *Electron. J. Differ. Equations* **2014**, *2014* (44), 1–16.
- (7) Wagner, L. O.; Baker, T. E.; Stoudenmire, E. M.; Burke, K.; White, S. R. Kohn-Sham calculations with the exact functional. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2014**, *90*, 045109.
- (8) Gontier, D. Existence of minimizers for Kohn-Sham within the local spin density approximation. *Nonlinearity* **2015**, *28*, 57–76.
- (9) Penz, M.; Laestadius, A.; Tellgren, E. I.; Ruggenthaler, M. Guaranteed Convergence of a Regularized Kohn-Sham Iteration in Finite Dimensions. *Phys. Rev. Lett.* **2019**, *123*, 037401.
- (10) Kelly, C. T. *Iterative Methods for Linear and Nonlinear Equations*; SIAM: Philadelphia, 1995.
- (11) Bertsekas, D. P. *Nonlinear Programming*, 2nd ed.; Athena Scientific: Nashua, NH, 2004; p 780.
- (12) Nocedal, J.; Wright, S. *Numerical Optimization*, 2nd ed.; Springer-Verlag: New York, 2006; p 664.
- (13) Martinez, J. M. Practical quasi-Newton methods for solving nonlinear systems. *J. Comput. Appl. Math.* **2000**, *124*, 97–121.
- (14) Polyak, B. *Introduction to Optimization*; Optimization Software, Inc.: New York, 2020; p 236.
- (15) Yuan, Y. X. Recent advances in trust region algorithms. *Math. Program.* **2015**, *151*, 249–281.
- (16) Francisco, J. B.; Martinez, J. M.; Martinez, L. Globally convergent trust-region methods for self-consistent field electronic structure calculations. *J. Chem. Phys.* **2004**, *121*, 10863–78.
- (17) Thogersen, L.; Olsen, J.; Yeager, D.; Jorgensen, P.; Salek, P.; Helgaker, T. The trust-region self-consistent field method: towards a black-box optimization in Hartree-Fock and Kohn-Sham theories. *J. Chem. Phys.* **2004**, *121*, 16–27.
- (18) Thogersen, L.; Olsen, J.; Kohn, A.; Jorgensen, P.; Salek, P.; Helgaker, T. The trust-region self-consistent field method in Kohn-Sham density-functional theory. *J. Chem. Phys.* **2005**, *123*, 074103.
- (19) Francisco, J. B.; Martinez, J. M.; Martinez, L. Density-based globally convergent trust-region methods for self-consistent field electronic structure calculations. *J. Math. Chem.* **2006**, *40*, 349–377.
- (20) Yang, C.; Meza, J. C.; Wang, L.-W. A Trust Region Direct Constrained Minimization Algorithm for the Kohn-Sham Equation. *SIAM J. Sci. Comput.* **2007**, *29*, 1854–1875.
- (21) Marks, L. D.; Luke, D. R. Robust mixing for ab initio quantum mechanical calculations. *arXiv (Condensed Matter.MATERIALS SCIENCE)*, 2008, 0801.3098, ver. 1. <https://arxiv.org/abs/0801.3098> (accessed 2021-08).
- (22) Marks, L. D.; Luke, D. R. Robust mixing for ab initio quantum mechanical calculations. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2008**, *78*, 075114.
- (23) Powell, M. J. D. A New Algorithm for Unconstrained Optimization. In *Nonlinear Programming*; Rosen, J. B., Mangasarian, O. L., Ritter, K., Eds.; Academic Press: 1970; pp 31–65.
- (24) Blaha, P.; Schwarz, K.; Madsen, G. K. H.; Kvasnicka, D.; Luitz, J.; Laskowsji, R.; Tran, F.; Marks, L. D. *An Augmented Plane Wave + Local Orbitals Program for Calculating Crystal Properties*; Technische Universität Wien: 2018.
- (25) Blaha, P.; Schwarz, K.; Tran, F.; Laskowski, R.; Madsen, G. K. H.; Marks, L. D. WIEN2k: An APW+lo program for calculating the properties of solids. *J. Chem. Phys.* **2020**, *152*, 074101.
- (26) Marks, L. D. Fixed-Point Optimization of Atoms and Density in DFT. *J. Chem. Theory Comput.* **2013**, *9*, 2786–2800.
- (27) Broyden, C. G. A Class of Methods for Solving Nonlinear Simultaneous Equations. *Math. Comput.* **1965**, *19*, 577–593.
- (28) Walker, H. F.; Ni, P. Anderson Acceleration for Fixed-Point Iterations. *SIAM J. Numer. Anal.* **2011**, *49*, 1715–1735.
- (29) Potra, F. A.; Engler, H. A characterization of the behavior of the Anderson acceleration on linear problems. *Linear Algebra Appl.* **2013**, *438*, 1002–1011.
- (30) Toth, A.; Kelley, C. T. Convergence Analysis for Anderson Acceleration. *SIAM J. Numer. Anal.* **2015**, *53*, 805–819.
- (31) Anderson, D. G. M. Comments on “Anderson Acceleration, Mixing and Extrapolation. *Numer. Algorithms* **2019**, *80*, 135–234.
- (32) Evans, C.; Pollock, S.; Rebholz, L. G.; Xiao, M. A Proof That Anderson Acceleration Improves the Convergence Rate in Linearly Converging Fixed-Point Methods (But Not in Those Converging Quadratically). *SIAM J. Numer. Anal.* **2020**, *58*, 788–810.
- (33) Anderson, D. G. Iterative Procedures for Nonlinear Integral Equations. *J. Assoc. Comput. Mach.* **1965**, *12*, 547–560.
- (34) Eyert, V. A comparative study on methods for convergence acceleration of iterative vector sequences. *J. Comput. Phys.* **1996**, *124*, 271–285.
- (35) Pratapa, P. P.; Suryanarayana, P. Restarted Pulay mixing for efficient and robust acceleration of fixed-point iterations. *Chem. Phys. Lett.* **2015**, *635*, 69–74.
- (36) Banerjee, A. S.; Suryanarayana, P.; Pask, J. E. Periodic Pulay method for robust and efficient convergence acceleration of self-consistent field iterations. *Chem. Phys. Lett.* **2016**, *647*, 31–35.
- (37) Pulay, P. Convergence Acceleration of Iterative Sequences - the Case of Scf Iteration. *Chem. Phys. Lett.* **1980**, *73*, 393–398.
- (38) Pulay, P. Improved Scf Convergence Acceleration. *J. Comput. Chem.* **1982**, *3*, 556–560.
- (39) Csaszar, P.; Pulay, P. Geometry Optimization by Direct Inversion in the Iterative Subspace. *J. Mol. Struct.* **1984**, *114*, 31–34.
- (40) Woods, N. *On the Nature of Self-Consistency in Density Functional Theory*. Master’s Thesis, University of Cambridge, 2018.
- (41) Woods, N. D.; Payne, M. C.; Hasnip, P. J. Computing the self-consistent field in Kohn-Sham density functional theory. *J. Phys.: Condens. Matter* **2019**, *31*, 453001.
- (42) Bendt, P.; Zunger, A. Simultaneous Relaxation of Nuclear Geometries and Electric Charge-Densities in Electronic-Structure Theories. *Phys. Rev. Lett.* **1983**, *50*, 1684–1688.
- (43) Yu, R.; Singh, D.; Krakaue, H. All-electron and pseudopotential force calculations using the linearized-augmented-plane-wave method. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1991**, *43*, 6411–6422.
- (44) Kohler, B.; Wilke, S.; Scheffler, M.; Kouba, R.; AmbroschDraxl, C. Force calculation and atomic-structure optimization for the full-potential linearized augmented plane-wave code WIEN. *Comput. Phys. Commun.* **1996**, *94*, 31–48.
- (45) Madsen, G. K. H.; Blaha, P.; Schwarz, K.; Sjostedt, E.; Nordstrom, L. Efficient linearization of the augmented plane-wave method. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2001**, *64*, 195134.
- (46) Tran, F.; Kunes, J.; Novak, P.; Blaha, P.; Marks, L. D.; Schwarz, K. Force calculation for orbital-dependent potentials with FP-(L)APW plus lo basis sets. *Comput. Phys. Commun.* **2008**, *179*, 784–790.
- (47) Rohwedder, T.; Schneider, R. An analysis for the DIIS acceleration method used in quantum chemistry calculations. *J. Math. Chem.* **2011**, *49*, 1889–1914.

- (48) Garza, A. J.; Scuseria, G. E. On the equivalence of LIST and DIIS methods for convergence acceleration. *J. Chem. Phys.* **2015**, *142*, 164104.
- (49) Zhang, J.; O'Donoghue, B.; Boyd, S. Globally Convergent Type-I Anderson Acceleration for Nonsmooth Fixed-Point Iterations. *SIAM J. Optimiz.* **2020**, *30*, 3170–3197.
- (50) Lin, L.; Yang, C. Elliptic Preconditioner for Accelerating the Self-Consistent Field Iteration in Kohn-Sham Density Functional Theory. *SIAM J. Sci. Comput.* **2013**, *35*, S277–S298.
- (51) Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C. *Introduction to Algorithms*; MIT Press: Boston, MA, 2009; p 1291.
- (52) Blügel, S. *First Principles Calculations of the Electronic Structure of Magnetic Overlays on Transition Metal Surfaces*; Zentralbibliothek der Kernforschungsanlage Jülich GmbH: 1988; p 125.
- (53) Oren, S. S.; Spedicato, E. Optimal Conditioning of Self-Scaling Variable Metric Algorithms. *Math. Program.* **1976**, *10*, 70–90.
- (54) Spedicato, E.; Greenstadt, J. Some Classes of Variationally Derived Quasi-Newton Methods for Systems of Non-Linear Algebraic Equations. *Numer. Math. (Heidelberg)* **1978**, *29*, 363–380.
- (55) Tikhonov, A. N. Solution of incorrectly formulated problems and the regularization method. *Dokl. Akad. Nauk SSSR* **1963**, *151*, 501–504.
- (56) Golub, G. H.; Heath, M.; Wahba, G. Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter. *Technometrics* **1979**, *21*, 215–223.
- (57) Hoerl, A. E.; Kennard, R. W. Ridge Regression - Biased Estimation for Nonorthogonal Problems. *Technometrics* **1970**, *12*, 55–67.
- (58) Schnabel, R. B.; Frank, P. D. Tensor Methods for Nonlinear Equations. *SIAM J. Numer. Anal.* **1984**, *21*, 815–843.
- (59) Feng, D.; Frank, P. D.; Schnabel, R. B. Local Convergence Analysis of Tensor Methods for Nonlinear Equations. *Math. Program.* **1993**, *62*, 427–459.
- (60) Bouaricha, A.; Schnabel, R. B. Algorithm 768: TENSOLVE: A software package for solving systems of nonlinear equations and nonlinear least-squares problems using tensor methods. *ACM Trans. Math. Softw.* **1997**, *23*, 174–195.
- (61) Bouaricha, A.; Schnabel, R. B. Tensor methods for large sparse systems of nonlinear equations. *Math. Program.* **1998**, *82*, 377–400.
- (62) Bouaricha, A.; Schnabel, R. B. Tensor Methods for Large, Sparse Nonlinear Least Squares Problems. *SIAM J. Sci. Comput.* **1999**, *21*, 1199–1221.
- (63) Custodio, A. L.; Dennis, J. E.; Vicente, L. N. Using simplex gradients of nonsmooth functions in direct search methods. *IMA J. Numer. Anal.* **2008**, *28*, 770–784.
- (64) Hare, W.; Jarry-Bolduc, G. Calculus Identities for Generalized Simplex Gradients: Rules and Applications. *SIAM J. Optimiz.* **2020**, *30*, 853–884.
- (65) Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.* **1944**, *2*, 164–168.
- (66) Marquardt, D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441.
- (67) Woods, N. D. Private communication, 2020.
- (68) Clark, S. J.; Segall, M. D.; Pickard, C. J.; Hasnip, P. J.; Probert, M. I. J.; Refson, K.; Payne, M. C. First principles methods using CASTEP. *Z. Kristallogr. - Cryst. Mater.* **2005**, *220*, 567–570.
- (69) Srivastava, G. P. Broyden Method for Self-Consistent Field Convergence Acceleration. *J. Phys. A: Math. Gen.* **1984**, *17*, L317–L321.
- (70) Srivastava, G. P. Correction: Broyden's method for self-consistent field convergence acceleration. *J. Phys. A: Math. Gen.* **1984**, *17*, 2737–2737.
- (71) Ip, C. M.; Todd, M. J. Optimal Conditioning and Convergence in Rank One Quasi-Newton Updates. *SIAM J. Numer. Anal.* **1988**, *25*, 206–221.
- (72) Conn, A. R.; Gould, N. I. M.; Toint, Ph. L. Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Math. Program.* **1991**, *50*, 177–195.
- (73) Khalfan, H. F.; Byrd, R. H.; Schnabel, R. B. A Theoretical and Experimental Study of the Symmetric Rank-One Update. *SIAM J. Optimiz.* **1993**, *3*, 1–24.
- (74) Byrd, R. H.; Khalfan, H. F.; Schnabel, R. B. Analysis of a symmetric rank-one trust region method. *SIAM J. Optimiz.* **1996**, *6*, 1025–1039.
- (75) Al-Baali, M.; Spedicato, E.; Maggioni, F. Broyden's quasi-Newton methods for a nonlinear system of equations and unconstrained optimization: a review and open problems. *Optim. Method. Softw.* **2014**, *29*, 937–954.
- (76) Goldfarb, D. A Family of Variable-Metric Methods Derived by Variational Means. *Math. Comput.* **1970**, *24*, 23–26.
- (77) Rheinboldt, W. C.; Vandergraft, J. S. On the Local Convergence of Update Methods. *SIAM J. Numer. Anal.* **1974**, *11*, 1069–1085.
- (78) Gragg, W. B.; Stewart, G. W. Stable Variant of Secant Method for Solving Nonlinear Equations. *SIAM J. Numer. Anal.* **1976**, *13*, 889–903.
- (79) Dennis, J. E., Jr.; Schnabel, R. B. Least Change Secant Updates for Quasi-Newton Methods. *SIAM Rev.* **1979**, *21*, 443–459.
- (80) Dennis, J. E., Jr.; Walker, H. F. Convergence Theorems for Least-Change Secant Update Methods. *SIAM J. Numer. Anal.* **1981**, *18*, 949–987.
- (81) Burdakov, O. P. On Superlinear Convergence of Some Stable Variants of the Secant Method. *Z. Angew. Math. Mech.* **1986**, *66*, 615–622.
- (82) Martinez, J. M. Local Convergence Theory of Inexact Newton Methods Based on Structured Least Change Updates. *Math. Comput.* **1990**, *55*, 143–167.
- (83) Gomesruggiero, M. A.; Martinez, J. M.; Moretti, A. C. Comparing Algorithms for Solving Sparse Nonlinear-Systems of Equations. *SIAM J. Sci. Stat. Comput.* **1992**, *13*, 459–483.
- (84) Lopes, V. L. R.; Martinez, J. M. Convergence properties of the inverse column-updating method. *Optim. Method. Softw.* **1995**, *6*, 127–144.
- (85) Spedicato, E.; Huang, Z. Numerical experience with Newton-like methods for nonlinear algebraic systems. *Computing* **1997**, *58*, 69–89.
- (86) Luksan, L.; Spedicato, E. Variable metric methods for unconstrained optimization and nonlinear least squares. *J. Comput. Appl. Math.* **2000**, *124*, 61–95.
- (87) *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*; Bauschke, H. H., Burachik, R. S., Combettes, P. J., Elser, V., Luke, D. R., Wolkowicz, H., Eds.; Springer Optimization and Its Applications 49; Springer: New York, 2011.
- (88) Shanno, D. F.; Phua, K. H. Matrix Conditioning and Non-Linear Optimization. *Math. Program.* **1978**, *14*, 149–160.
- (89) Guerber, T. *Improvement of the MSRIa algorithm in the WIEN2k code*; Université de Rennes: 2019.
- (90) Baldereschi, A.; Tosatti, E. Dielectric Band-Structure of Solids. *Solid State Commun.* **1979**, *29*, 131–135.
- (91) Car, R.; Tosatti, E.; Baroni, S.; Leela-prute, S. Dielectric Band-Structure of Crystals - General-Properties and Calculations for Silicon. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1981**, *24*, 985–999.
- (92) Ho, K.-M.; Ihm, J.; Joannopoulos, J. D. Dielectric matrix scheme for fast convergence in self-consistent electronic-structure calculations. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1982**, *25*, 4260–4262.
- (93) Hybertsen, M. S.; Louie, S. G. Ab initio static dielectric matrices from the density-functional approach. I. Formulation and application to semiconductors and insulators. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1987**, *35*, 5585–5601.
- (94) Castro, J. D. E.; Muniz, R. B.; Oliveira, L. E. A Model Calculation of the Dielectric Band-Structure of Diamond and Silicon. *Phys. Status Solidi B* **1990**, *158*, 743–752.
- (95) Galamic-Mulaomerovic, S.; Hogan, C. D.; Patterson, C. H. Eigenfunctions of the inverse dielectric functions and response functions of silicon and argon. *Phys. Status Solidi A* **2001**, *188*, 1291–1296.
- (96) Wilson, H. F.; Lu, D. Y.; Gygi, F.; Galli, G. Iterative calculations of dielectric eigenvalue spectra. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2009**, *79*, 245106.

- (97) Powell, M. J. D. *A Hybrid Method for Non-Linear Equations*; Gordon and Breach: London, 1970.
- (98) Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; Vetterling, W. T. *Numerical Recipes*; Cambridge University Press: Cambridge, U.K., 1989; p 1482.
- (99) Barzilai, J.; Borwein, J. M. Two-Point Step Size Gradient Methods. *IMA J. Numer. Anal.* **1988**, *8*, 141–148.
- (100) Raydan, M. On the Barzilai and Borwein choice of steplength for the gradient method. *IMA J. Numer. Anal.* **1993**, *13*, 321–326.
- (101) Dai, Y.-H.; Zhang, H. Adaptive two-point stepsize gradient algorithm. *Numer. Algorithms* **2001**, *27*, 377–385.
- (102) Fletcher, R. On the Barzilai-Borwein method. In *Optimization and Control with Applications*; Qi, L., Teo, K., Yang, X., Eds.; Applied Optimization 96; Springer: 2005; pp 235–256..
- (103) Liu, Z.; Liu, H. Several efficient gradient methods with approximate optimal stepsizes for large scale unconstrained optimization. *J. Comput. Appl. Math.* **2018**, *328*, 400–413.
- (104) Kim, J.; Gulans, A.; Draxl, C. Robust mixing in self-consistent linearized augmented planewave calculations. *Electron. Struct.* **2020**, *2*, 037001.
- (105) Winkelmann, M.; Di Napoli, E.; Wortmann, D.; Blugel, S. Kerker mixing scheme for self-consistent muffin-tin based all-electron electronic structure calculations. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2020**, *102*, 195138.
- (106) Blaha, P.; Schwarz, K.; Sorantin, P.; Trickey, S. B. Full-Potential, Linearized Augmented Plane-Wave Programs for Crystalline Systems. *Comput. Phys. Commun.* **1990**, *59*, 399–415.
- (107) Blaha, P.; Singh, D. J.; Sorantin, P. I.; Schwarz, K. Electric-field-gradient calculations for systems with large extended-core-state contributions. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1992**, *46*, 1321–1325.
- (108) Singh, D.; Nordstrom, L. *Planewaves, Pseudopotentials and the LAPW Method*; Springer: 2006; p 134.
- (109) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **1996**, *77*, 3865–3868.
- (110) Daniels, A. D.; Scuseria, G. E. Converging difficult SCF cases with conjugate gradient density matrix search. *Phys. Chem. Chem. Phys.* **2000**, *2*, 2173–2176.
- (111) Woods, N. D. Set of Kohn-Sham DFT inputs (CASTEP format) for which the SCF iterations are difficult to converge. <https://github.com/NickWoods1/scf-xn-testsuite>.
- (112) Smith Pellizzeri, T. M.; McMillen, C. D.; Wen, Y.; Chumanov, G.; Kolis, J. W. Iron Vanadates Synthesized from Hydrothermal Brines: Rb<sub>2</sub>FeV<sub>6</sub>O<sub>16</sub>, Cs<sub>2</sub>FeV<sub>6</sub>O<sub>16</sub>, and SrFe<sub>3</sub>V<sub>18</sub>O<sub>38</sub>. *Eur. J. Inorg. Chem.* **2019**, *2019*, 4538–4545.
- (113) Gatehouse, B. M.; Platts, S. N.; Williams, T. B. Structure of anhydrous titanyl sulfate, titanyl sulfate monohydrate and prediction of a new structure. *Acta Crystallogr., Sect. B: Struct. Sci.* **1993**, *49*, 428–435.
- (114) Cannillo, E.; Mazzi, F.; Rossi, G. The structure type of Joaquinite. *TMPM, Tschermaks Mineral. Petrogr. Mitt.* **1972**, *17*, 233–246.
- (115) Jana, P. P.; Pankova, A. A.; Lidin, S. Au<sub>10</sub>Mo<sub>4</sub>Zn<sub>89</sub>: a fully ordered complex intermetallic compound analyzed by TOPOS. *Inorg. Chem.* **2013**, *S2*, 11110–7.
- (116) Wolf, S.; Fenske, D.; Klopper, W.; Feldmann, C. [Pb{Mn-(CO)<sub>5</sub>}<sub>3</sub>][AlCl<sub>4</sub>]: a lead-manganese carbonyl with AlCl<sub>4</sub>-linked PbMn<sub>3</sub> clusters. *Dalton Trans.* **2019**, *48*, 4696–4701.
- (117) Wada, H.; Sato, A.; Onoda, M.; Adams, S.; Tansho, M.; Ishii, M. Phase transition and crystal structure of silver-ion conductor Ag<sub>12-nM+nS<sub>6</sub></sub> (M = Ti, Nb, Ta). *Solid State Ionics* **2002**, *154-155*, 723–727.
- (118) Lepage, Y.; Marezio, M. Structural Chemistry of Magneli Phases Ti<sub>n</sub>O<sub>2n-1</sub> (4 Less-Than-or-Equal-to N Less-Than-or-Equal-to 9). 4. Superstructure in Ti<sub>4</sub>O<sub>7</sub> at 140 K. *J. Solid State Chem.* **1984**, *S3*, 13–21.
- (119) Bindi, L.; Pham, J.; Steinhardt, P. J. Previously unknown quasicrystal periodic approximant found in space. *Sci. Rep.* **2018**, *8*, 16271.
- (120) Enterkin, J. A.; Subramanian, A. K.; Russell, B. C.; Castell, M. R.; Poepelmeier, K. R.; Marks, L. D. A homologous series of structures on the surface of SrTiO<sub>3</sub>(110). *Nat. Mater.* **2010**, *9*, 245–8.
- (121) Laskowski, R.; Blaha, P. Unraveling the structure of the h-BN/Rh(111) nanomesh with ab initio calculations. *J. Phys.: Condens. Matter* **2008**, *20*, 064207.
- (122) Bertsekas, D. P. *Nonlinear Programming*, 2nd ed.; Solutions Manual. 2004. [www.athenasc.com/nlpso1.pdf](http://www.athenasc.com/nlpso1.pdf).
- (123) Toth, A.; Ellis, J. A.; Evans, T.; Hamilton, S.; Kelley, C. T.; Pawłowski, R.; Slattery, S. Local Improvement Results for Anderson Acceleration with Inaccurate Function Evaluations. *SIAM J. Sci. Comput.* **2017**, *39*, S47–S65.
- (124) Kerker, G. P. Efficient Iteration Scheme for Self-Consistent Pseudopotential Calculations. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1981**, *23*, 3082–3084.
- (125) Raczkowski, D.; Canning, A.; Wang, L. W. Thomas-Fermi charge mixing for obtaining self-consistency in density functional calculations. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2001**, *64*, 121101.
- (126) Anglade, P. M.; Gonze, X. Preconditioning of self-consistent-field cycles in density-functional theory: The extrapolar method. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2008**, *78*, 045126.
- (127) Shiihara, Y.; Kuwazuru, O.; Yoshikawa, N. Real-space Kerker method for self-consistent calculation using non-orthogonal basis functions. *Modell. Simul. Mater. Sci. Eng.* **2008**, *16*, 035004.
- (128) Zhou, Y.; Wang, H.; Liu, Y.; Gao, X.; Song, H. Applicability of Kerker preconditioning scheme to the self-consistent density functional theory calculations of inhomogeneous systems. *Phys. Rev. E: Stat. Phys., Plasmas, Fluids, Relat. Interdiscip. Top.* **2018**, *97*, 033305.
- (129) Kumar, S.; Xu, Q. M.; Suryanarayana, P. On preconditioning the self-consistent field iteration in real-space Density Functional Theory. *Chem. Phys. Lett.* **2020**, *739*, 136983.
- (130) Herbst, M. F.; Levitt, A. Black-box inhomogeneous preconditioning for self-consistent field iterations in density functional theory. *J. Phys.: Condens. Matter* **2021**, *33*, 085503.
- (131) Rondinelli, J. M.; Deng, B.; Marks, L. D. Enhancing structure relaxations for first-principles codes: An approximate Hessian approach. *Comput. Mater. Sci.* **2007**, *40*, 345–353.
- (132) Weinert, M. Solution of Poisson's equation: Beyond Ewald-type methods. *J. Math. Phys.* **1981**, *22*, 2433–2439.