Two Classes of Multisecant Methods for Nonlinear Acceleration [*]

Haw-ren Fang[†] Yousef Saad[†]

June 1, 2007

Abstract

Many applications in science and engineering lead to models which require solving large-scale fixed point problems, or equivalently, systems of nonlinear equations. Several successful techniques for handling such problems are based on quasi-Newton methods that implicitly update the approximate Jacobian or inverse Jacobian to satisfy a certain secant condition.

We present two classes of multisecant methods which allows to take into account a variable number of secant equations at each iteration. The first is the Broyden-like class, of which Broyden's family is a subclass, and Anderson mixing is a particular member. The second class is that of the nonlinear Eirola-Nevanlinna-type methods.

This work was motivated by a problem in electronic structure calculations, whereby a fixed point iteration, known as the self-consistent field (SCF) iteration, is accelerated by various strategies termed 'mixing'.

1 Introduction

We consider the solution of large-scale fixed point problems with applications such as electronic structure calculations [9], or complex transportation systems [3]. The fixed point problem is to find x such that $g(x) = x$ with $g : \mathbb{R}^n \to \mathbb{R}^n$. For example, in the electronic structure problem, x is a certain potential and $g(x)$ is the result of a very complex calculation which delivers another potential. The iteration is 'self-consistent' then the input and output potentials are the same.

As is well-known, computing a fixed point is equivalent to solving the nonlinear system of equations $f(x) = 0$, by simply defining $f(x) \equiv x - g(x)$. The problems of interest to us in this paper are those which have the following characteristics [3]:

[*] Work supported by DOE under grant DE-FG02-03ER25585, by NSF under grant DMR-0325218, and by the Minnesota Supercomputer Institute.

[†] Department of Computer Science and Engineering; University of Minnesota; Minneapolis, MN 55455.

1. The dimensionality of the problem is large.
2. $f(x)$ is continuously differentiable, but the analytic form of its derivative is not readily available, or it is very expensive to compute.
3. The cost of evaluating $f(x)$ is computationally high.
4. The problem is noisy. In other words, the evaluated function values of $f(x)$ usually contain errors.

Another critical consideration, which we now describe, is a practical one which will underline the difference between the problem of solving a nonlinear equation and that of accelerating a sequence of vectors. Specifically, there is limited freedom in the use of f, in that $f(x)$ is the result of some complex calculation which would be too cumbersome to rewrite in the form of a subroutine call. In other words, one can consider that the problem is to accelerate a sequence of vectors produced by a certain complex code. This is in contrast with another common situation where we are given a procedure (subroutine) for computing $f(x)$ and an initial guess. For this reason, there is a very important distinction to be made between the problem of solving a nonlinear system of equations and that of accelerating a vector sequence. From one viewpoint, the two problems are equivalent, since the sequence $x_{k+1} = \bar{f}(x_k)$ is attempting to find the solution to the system $x - \bar{f}(x) = 0$. However, there may be many practical constraints that will not allow the use of certain types of nonlinear techniques, in addition to the unavailability of the Jacobian. For example, in some instances the sequence x_k has physical meaning, and it may not be safe or practical to calculate $\bar{f}(x_k + v)$ for an arbitrary v.

These characteristics restrict us from several existing methods. For example, characteristic 1 suggests using matrix-free limited-memory algorithms. Because of characteristic 2, standard methods, such as Newton's method, which explicitly require the derivatives cannot be directly applied. Characteristic 3 prevents us from line search techniques. Newton-Krylov methods [5, 6] can be made matrix-free by incorporating finite difference schemes but all finite difference schemes are discouraged by characteristics 3 and 4. As a result, algorithms relying on secant equations have become the preferred approach for tackling such problems. These include Anderson mixing [1], Broyden's methods [7, 22, 26], modified or generalized Broyden's methods [12, 16, 28].

This paper is organized as follows. Section 2 gives a brief review of existing methods, including Anderson mixing [1], Broyden's methods [7], generalized Broyden's methods [12], and the nonlinear Eirola-Nevanlinna-like method [29, chapter 7]. Inspired by existing work, we present two classes of multisecant methods in Section 3, where limited-memory algorithms are also presented. Treatments for numerical stability are given in Section 4. Experimental results are reported in Section 5. Concluding remarks are given in Section 6.

2 Quasi-Newton methods

We consider a large system of nonlinear equations $f(x) = 0$ where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. Then we can write

$$f(x + \Delta x) \approx f(x) + J(x)\Delta x, \tag{1}$$

where $J(x)$ is the Jacobian matrix at x. At the iterate x_k, the Newton step Δx_k is determined by

$$J(x_k)\Delta x_k = -f(x_k). \tag{2}$$

Then $f(x_k + \Delta x_k) \approx 0$ if the solution Δx_k to (2) is small enough. The iterations are repeated by setting

$$x_{k+1} := x_k + \Delta x_k = x_k - J(x_k)^{-1} f(x_k) \tag{3}$$

for $k = 1, 2, \ldots$ until a good enough solution is obtained. If $f(x)$ is linear with a nonsingular Jacobian, then the solution is reached in one step.

Newton's method requires that the Jacobian be available at each iteration, which is often not practical. Quasi-Newton methods approximate $J(x_k)$ by J_k, and obtain J_{k+1} from J_k by adding a low-rank matrix at each iteration. They implicitly take advantage of the analytic information without explicitly computing the actual Jacobian.

2.1 Broyden's method

Standard quasi-Newton methods require that property (1) be satisfied by the updated J_{k+1}; in other words, the following *secant condition* is imposed:

$$J_{k+1}\Delta x_k = \Delta f_k, \tag{4}$$

where $\Delta f_k := f(x_{k+1}) - f(x_k)$.

Furthermore, another common requirement is the following so-called *no-change condition*:

$$J_{k+1} q = J_k q, \quad \forall q \quad \text{such that} \quad q^T \Delta x_k = 0, \tag{5}$$

which stipulates that there be no new information from J_k to J_{k+1} along any direction q orthogonal to Δx_k.

Broyden [7] developed a method satisfying both secant condition (4) and the no-change condition (5). By simply imposing these conditions he arrived at the update formula,

$$J_{k+1} = J_k + (\Delta f_k - J_k \Delta x_k)\frac{\Delta x_k^T}{\Delta x_k^T \Delta x_k}. \tag{6}$$

The matrix J_{k+1} in (6) is the unique matrix satisfying both conditions (4) and (5). Dennis and Moré [15] showed that the Broyden update can also be obtained by minimizing $E(J_{k+1}) = \|J_{k+1} - J_k\|_F^2$ with respect to terms of J_{k+1}, subject to the secant condition (4).

It may seem at first that Broyden's first method can be expensive since computing the quasi-Newton step Δx_k requires solving a linear system (2) at each iteration. However, note that, typically, the approximate Jacobian is a small rank modification of a diagonal matrix (or a matrix that is easy to invert) so the cost of the solve is actually not too high as long as the number of steps is not too large.

An alternative, is Broyden's second method which approximates the inverse Jacobian instead of the Jacobian itself. We use G_k to denote the estimated inverse Jacobian at the k-th iteration. The secant condition (4) now reads,

$$G_{k+1}\Delta f_k = \Delta x_k. \tag{7}$$

By minimizing $E(G_{k+1}) = \|G_{k+1} - G_k\|_F^2$ with respect to G_{k+1} subject to (7), one finds this update formula for the inverse Jacobian:

$$G_{k+1} = G_k + (\Delta x_k - G_k \Delta f_k)\frac{\Delta f_k^T}{\Delta f_k^T \Delta f_k}, \tag{8}$$

which is also the only update satisfying both the secant condition (7) and the no-change condition for the inverse Jacobian:

$$G_k q = G_{k+1} q, \quad \forall\ q \quad \text{such that} \quad q^T \Delta f_k = 0. \tag{9}$$

We can also obtain the update formula (6) in terms of $G_k \equiv J_k^{-1}$ by applying the Sherman-Morrison formula:

$$G_{k+1} = G_k + (\Delta x_k - G_k \Delta f_k)\frac{\Delta x_k^T G_k}{\Delta x_k^T G_k \Delta f_k}. \tag{10}$$

This shows, as was explained earlier, that the solve of the Jacobian system associated with Broyden's first approach can be reduced to a set of update operations that are not more costly than those required by the second update. Note however that the above formula requires the inverse of the initial Jacobian.

2.2 Broyden's family

From (8) and (10) it is possible to define *Broyden's family* of updates, in which an update formula takes the general form

$$G_{k+1} = G_k + (\Delta x_k - G_k \Delta f_k) v_k^T, \tag{11}$$

where $v_k^T \Delta f_k = 1$ so that the secant condition (7) holds [7]. Note that the secant condition (7) is equivalent to the condition (4). The pseudo-code of Broyden's two methods is given in Algorithm 1.

Some authors called Broyden's first method Broyden's *good* update, and Broyden's second method, Broyden's *bad* update. These are two particular members in Broyden's family. In addition, listed below are three other types of update that satisfy the secant condition:

Algorithm 1 Broyden's methods for solving $f(x) = 0$.

```
Given the initial guess $x_1$, $G_1$.
$f_1 := f(x_1)$
**for all** $k = 1, 2, \ldots$ until convergence **do**
    $\Delta x_k := -G_k f_k$
    $x_{k+1} := x_k + \Delta x_k$
    $f_{k+1} := f(x_{k+1})$
    $\Delta f_k := f_{k+1} - f_k$
    Select $\hat{v}_k^T \ = \begin{cases} \Delta x_k^T G_k & \text{if} \quad \text{first update;} \\ \Delta f_k^T & \text{if} \quad \text{second update.} \end{cases}$
    $G_{k+1} := G_k + (\Delta x_k - G_k \ \Delta f_k)\frac{\hat{v}_k^T}{\hat{v}_k^T \ \Delta f_k}$
**end for**
```

1. **Direct update of factorizations.** An algorithm of this type (e.g., [17, 20]) implicitly assumes the Jacobian being factorized as a product of two matrices (typically LU factorization), and updates the two matrices separately to satisfy the secant condition at each iteration. The sparsity of the approximate Jacobian or its factorization is sometimes assumed and taken into account for efficiency.

2. **Structured methods.** This type of methods relies on the property that the Jacobian is in the form $J(x) = C(x) + D(x)$, where $C(x)$ is easy to compute but $D(x)$ is not. At each iteration we update D_{k+1} from D_k so that $J_{k+1} := C(x_{k+1}) + D_{k+1}$ satisfies the secant condition.

3. **Column-updating methods.** Martínez introduced a column-updating method on the Jacobian [19], which consists of updating one column of J_k at a time:

   $$J_{k+1} = J_k + (\Delta f_k - J_k \Delta x_k)\frac{e_{j_k}^T}{e_{j_k}^T \Delta x_k},$$

   where $e_{j_k}^T$ is a column vector that is zero except for a 1 in its j_kth position. This scheme is a member of the Broyden family with, using the notation of the algorithm, $\hat{v}_k = e_{j_k}^T G_k$ for the first update and $\hat{v}_k = e_{j_k}^T$ for the second update method which was later proposed by the same author [21].

2.3 Generalized Broyden's method

Inspired by the work of Vanderbilt and Louie [28], Eyert [12] proposed a generalized Broyden's method with a flexible rank of update on the inverse Jacobian, satisfying a set of m secant equations

$$G_k \Delta f_i = \Delta x_i \text{ for } i = k-m, \ldots, k-1, \tag{12}$$

where we assume $\Delta f_{k-m}, \ldots, \Delta f_{k-1}$ are linearly independent and $m \leq n$. The idea of taking multiple secant equations into account is not new. It has been well-studied by several researchers (see, e.g., [2, 13, 14]). Aggregating (12) in matrix form, we can rewrite it as

$$G_k \mathcal{F}_k = \mathcal{X}_k, \tag{13}$$

where

$$\mathcal{F}_k = [\Delta f_{k-m} \ \cdots \ \Delta f_{k-1}], \ \mathcal{X}_k = [\Delta x_{k-m} \ \cdots \ \Delta x_{k-1}] \in \mathbb{R}^{n\times m}. \tag{14}$$

The no-change condition corresponding to (9) is

$$(G_k - G_{k-m})q = 0 \tag{15}$$

for all q orthogonal to the subspace spanned by $\Delta f_{k-m}, \ldots, \Delta f_{k-1}$, the columns of $\mathcal{F}_k$. This means that the null space of $G_k - G_{k-m}$ is the orthogonal of $\mathcal{F}_k$. Since $\text{Null}(X)^\perp = \text{Range}(X^T)$ this is equivalent to the condition that $(G_k - G_{k-m})^T = \mathcal{F}_k Z^T$ for a certain unknown matrix Z. This matrix Z can now be obtained from condition (13). Multiplying $G_k - G_{k-m} = Z\mathcal{F}_k^T$ to the right by $\mathcal{F}_k$, we get:

$$(G_k - G_{k-m})\mathcal{F}_k = Z\mathcal{F}_k^T \mathcal{F}_k \quad \Rightarrow \quad Z = (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)(\mathcal{F}_k^T \mathcal{F}_k)^{-1},$$

where we assume $\mathcal{F}_k$ has full column rank. In the end this yields,

$$G_k = G_{k-m} + (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)(\mathcal{F}_k^T \mathcal{F}_k)^{-1}\mathcal{F}_k^T, \tag{16}$$

a rank-m update formula. Note that $\text{rank}(\mathcal{F}_k) = m$.

The update formula for x_{k+1} is obtained by substituting (16) into (3),

$$\begin{aligned} x_{k+1} &= x_k - G_k f_k \\ &= x_k - G_{k-m} f_k - (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)(\mathcal{F}_k^T \mathcal{F}_k)^{-1}\mathcal{F}_k^T f_k \\ &= x_k - G_{k-m} f_k - (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)\gamma_k, \end{aligned} \tag{17}$$

where the column vector γ_k is obtained by solving the normal equations $(\mathcal{F}_k^T \mathcal{F}_k)\gamma_k = \mathcal{F}_k^T f_k$, which is equivalent to solving the least squares problem

$$\min_\gamma \|\mathcal{F}_k \gamma - f_k\|_2 \ . \tag{18}$$

Note that in (17), if $\mathcal{F}_k$ is square and of full rank, then for any G_{k-m},

$$x_{k+1} = x_k - \mathcal{X}_k \mathcal{F}_k^{-1} f_k, \tag{19}$$

the same form as that in the standard secant method (see, e.g., [14]).

The update matrix G_k in the formula (16) is:

1. The only formula satisfying both the secant condition $G_k\mathcal{F}_k = \mathcal{X}_k$ in (12) and the no-change condition (15).

2. The minimizer of $E(G_k) = \|G_k - G_{k-m}\|_F$ subject to $G_k\mathcal{F}_k = \mathcal{X}_k$.

2.4 The nonlinear Eirola-Nevanlinna-like method

Eirola and Nevanlinna [11] proposed an iterative method to approximate A^{-1} by adding a rank-one update at each iteration while the approximation of the solution to a linear system $Ax = b$ is improved simultaneously. Yang [29, chapter 7] generalized this scheme to solve nonlinear systems. She called this the *nonlinear EN-like method*. The pseudo-code is given in Algorithm 2.

Algorithm 2 Nonlinear Eirola-Nevanlinna-like method.

```
Given the initial guess $x_1$, $G_1$.
for all $k = 1, 2, \ldots$ until convergence do
    $f_k := f(x_k)$
    $p_k := -G_k f_k$
    $q_k := f(x_k + p_k) - f_k$
    $G_{k+1} := G_k + (p_k - G_k q_k)\frac{p_k^T G_k}{p_k^T G_k q_k}$          ▷ (*)
    $x_{k+1} := x_k - G_{k+1} f_k$
end for
```

Note that the nonlinear EN-like method (Algorithm 2) requires two function evaluations per iteration whereas Broyden's method (Algorithm 1) needs only one. Both methods perform a rank-one update per iteration. Yang showed that the nonlinear EN-like method converges twice as fast as Broyden's method in terms of number of iterations [29, chapter 7]. In other words, the two methods should in theory converge with a similar speed, requiring a similar number of function evaluations.

2.5 Anderson mixing

Consider a procedure for solving a large nonlinear system of equations $f(x) = 0$ by an iterative process. The most recent iterates are denoted by $x_{k-m}, \ldots, x_k \in \mathbb{R}^n$ and the corresponding outputs $f_{k-m}, \ldots, f_k \in \mathbb{R}^n$. Assuming evaluating $f(x)$ is expensive and no explicit analytic form of $f(x)$ is available, the challenge is to determine the next estimate x_{k+1} that approximates the solution to $f(x) = 0$ without additional evaluations of $f(x)$.

The Anderson mixing scheme [1] takes the latest m steps into account[1]:

$$\bar{x}_k = x_k - \sum_{i=k-m}^{k-1} \gamma_i^{(k)} \Delta x_i = x_k - \mathcal{X}_k \gamma_k, \tag{20}$$

$$\bar{f}_k = f_k - \sum_{i=k-m}^{k-1} \gamma_i^{(k)} \Delta f_i = f_k - \mathcal{F}_k \gamma_k, \tag{21}$$

[1] Anderson originally formulated his mixing scheme as $\bar{x}_k = x_k + \sum_{j=1}^m \theta_j^{(k)}(x_{k-j} - x_k)$ and $\bar{f}_k = f_k + \sum_{j=1}^m \theta_j^{(k)}(f_{k-j} - f_k)$. The formulation (20) (21) is equivalent to his by taking $\gamma_i^{(k)} = \sum_{j=k-i}^m \theta_j^{(k)}$.

where $\Delta x_i = x_{i+1} - x_i$ and $\Delta f_i = f_{i+1} - f_i$, $\mathcal{X}_k = [\Delta x_{k-m} \ \cdots \ \Delta x_{k-1}]$, $\mathcal{F}_k = [\Delta f_{k-m} \ \cdots \ \Delta f_{k-1}]$, and $\gamma_k = [\gamma^{(k)}_{k-m} \ \cdots \ \gamma^{(k)}_{k-1}]^T$. Expressing the equations in the form $\bar{x}_k = \sum_{j=k-m}^{k} w_j x_j$ and $\bar{f}_k = \sum_{j=k-m}^{k} w_j f_j$, we find that $\sum_{j=k-m}^{k} w_j = 1$. In other words, $\bar{x}_k$ and $\bar{f}_k$ are weighted averages of $x_{k-m}, \ldots, x_k$ and $f_{k-m}, \ldots, f_k$, respectively.

The arguments $\gamma_i = [\gamma^{(k)}_{k-m} \ \cdots \ \gamma^{(k)}_{k-1}]^T$ are determined by minimizing

$$E(\gamma_k) = \langle \bar{f}_k, \bar{f}_k \rangle = \|f_k - \mathcal{F}_k \gamma_k\|_2^2, \tag{22}$$

whose solution can (but should not in practice) be obtained by solving the normal equations

$$(\mathcal{F}_k^T \mathcal{F}_k)\gamma_k = \mathcal{F}_k^T f_k. \tag{23}$$

Combining (20), (21), and (23), we obtain

$$\begin{aligned} x_{k+1} &= \bar{x}_k + \beta_k \bar{f}_k \\ &= x_k + \beta_k f_k - (\mathcal{X}_k + \beta_k \mathcal{F}_k)\gamma_k \\ &= x_k + \beta_k f_k - (\mathcal{X}_k + \beta_k \mathcal{F}_k)(\mathcal{F}_k^T \mathcal{F}_k)^{-1} \mathcal{F}_k^T f_k, \end{aligned} \tag{24}$$

where we assume $\mathcal{F}_k^T \mathcal{F}_k$ is nonsingular. In particular, if no previous iterate is taken into account (i.e., $m = 0$), then (24) reads,

$$x_{k+1} = x_k + \beta f_k. \tag{25}$$

This scheme is referred to as *simple mixing*.

The update formula (24) is the same as (17) by setting $G_{k-m} = -\beta_k I$. In this respect Anderson mixing implicitly forms an approximate inverse Jacobian G_k that minimizes $\|G_k + \beta_k I\|_F$ subject to (13). In the context of mixing, generalized Broyden's second method is equivalent to Anderson mixing. This equivalence relation was shown by Eyert [12]. Note that if $\mathcal{F}_k$ is square and nonsingular, then (24) matches the formula (19) of the standard secant method.

3 Two classes of multisecant methods

Sections 3.1 and 3.2 derive generalized Broyden's family and Anderson's family, respectively. Sections 3.3 and 3.4 give the Broyden-like class and the Eirola-Nevanlinna-like class of multisecant methods, respectively. Section 3.5 presents the hybrid methods.

3.1 Generalized Broyden's family

At the kth iteration a quasi-Newton method minimizes the change of the approximate Jacobian J_k or the inverse Jacobian G_k in the Frobenius norm ($G_k \equiv J_k^{-1}$). The authors [3, 12] working on the generalized quasi-Newton methods favored minimizing the change of G_k rather than J_k in the Frobenius norm. An observation made it possible to obtain an update formula in terms of G_k that minimizes the change of J_k in the Frobenius norm.

Following the notation in Section 2.3 and in a similar vein to derive the formula (16), we get the update formula for the approximate Jacobian J_k:

$$J_k = J_{k-m} + (\mathcal{F}_k - J_{k-m}\mathcal{X}_k)(\mathcal{X}_k^T\mathcal{X}_k)^{-1}\mathcal{X}_k^T, \tag{26}$$

where $\mathcal{F}_k$ and $\mathcal{X}_k$ are defined in (14). The J_k obtained in the above expression is:

1. The only formula satisfies both the secant condition $J_k\mathcal{X}_k = \mathcal{F}_k$ and the no-change condition $J_k q = J_{k-m}q$ for q orthogonal to the subspace spanned by the columns of $\mathcal{X}_k$.

2. The minimizer of $E(J_k) = \|J_k - J_{k-m}\|_F$ subject to $J_k\mathcal{X}_k = \mathcal{F}_k$.

The formula (26) can be expressed in terms of $G_k \equiv J_k^{-1}$ by applying the Woodbury formula:

$$G_k = G_{k-m} + (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)(\mathcal{X}_k^T G_{k-m}\mathcal{F}_k)^{-1}\mathcal{X}_k^T G_{k-m}. \tag{27}$$

Given a set of secant equations represented by $\mathcal{X}_k$ and $\mathcal{F}_k$, we call Type-I methods, those methods that update G_k by (27), by minimizing the change of the approximate Jacobian in the Frobenius norm. Type-I methods differ from each other by how the set of secant equations is chosen. We call Type-II methods, those methods that update G_k by (16) by minimizing the change of the approximate inverse Jacobian in the Frobenius norm. Likewise, Type-II methods differ from each other by how the set of secant equations is chosen.

Now we can write down the *generalized Broyden family*, in which an update algorithm is in the form

$$G_k = G_{k-m} + (\mathcal{X}_k - G_{k-m}\mathcal{F}_k)V_k^T, \tag{28}$$

where $V_k^T\mathcal{F}_k = I$ so that the secant condition $G_k\mathcal{F}_k = \mathcal{X}_k$ in (13) holds. The generalized Broyden's first method (27) and generalized Broyden's second method (16) are two particular members in this family. Note that if $\mathcal{F}_k$ is square and nonsingular, then $V_k^T = \mathcal{F}_k^{-1}$ is unique and the resulting formula $G_k = \mathcal{X}_k\mathcal{F}_k^{-1}$ corresponds to the regular secant method (19).

The column-updating methods by Martínez [19, 21] in Broyden's family (11) can also be generalized to be m-column-updating methods. The advantage is that the cost of updating approximate Jacobian or inverse Jacobian is reduced. In the context of mixing, the function evaluations are much more expensive and therefore the convergence rate in terms of number of iterations is important. Hence we use the optimal update formulas (16) and (27) in the sense of least change.

3.2 Anderson's family

Recall that Anderson mixing implicitly forms the approximate inverse Jacobian G_k and minimizes $\|G_k+\beta I\|_F$ subject to the available secant equations [12]. The generalized Broyden's second method is a particular member in the generalized Broyden's family (28), where $V_k^T = (\mathcal{F}_k^T\mathcal{F}_k)^{-1}\mathcal{F}_k$. Likewise, we can replace $(\mathcal{F}_k^T\mathcal{F}_k)^{-1}\mathcal{F}_k$ in (24) by V_k^T and obtain *Anderson's family*:

$$x_{k+1} = x_k + \beta_k f_k - (\mathcal{X}_k + \beta_k\mathcal{F}_k)V_k^T f_k, \tag{29}$$

where $V_k \in \mathbb{R}^{n\times m}$ satisfies $V_k^T\mathcal{F}_k = I$ implicitly for the secant condition (13). There are two particular members in the generalized Broyden family (28): the Type-I method (27) and the Type-II method (16). The latter corresponds to Anderson mixing (24) which is therefore a Type-II method. In a similar vein, to obtain (27), we set $V_k = (\mathcal{X}_k^T\mathcal{F}_k)^{-1}\mathcal{X}_k^T$ in (29) and obtain the Type-I Anderson mixing:

$$x_{k+1} = x_k + \beta_k f_k - (\mathcal{X}_k + \beta_k\mathcal{F}_k)(\mathcal{X}_k^T\mathcal{F}_k)^{-1}\mathcal{X}_k^T f_k, \tag{30}$$

which minimizes $\|J_k + \frac{1}{\beta}I\|_F$ subject to the available secant equations, where J_k is the implicitly formed approximate Jacobian.

3.3 The Broyden-like class

Now we describe the Broyden-like class of multisecant methods. For large-scale problems, we are interested in limited-memory algorithms. Consider solving a nonlinear system $f(x) = 0$ and suppose we have the latest m iterates available, which are denoted by $x_1,\dots,x_m$. Let $\Delta x_i = x_{i+1} - x_i$ for $i = 1,\dots,m-1$. Partition $\Delta x_1,\dots,\Delta x_{m-1}$ into k groups,

$$\begin{aligned}
\mathcal{X}_1 &= [\Delta x_1,\dots,\Delta x_{z_1}],\\
\mathcal{X}_2 &= [\Delta x_{z_1+1},\dots,\Delta x_{z_2}],\\
&\vdots\\
\mathcal{X}_k &= [\Delta x_{z_{k-1}+1},\dots,\Delta x_{z_k}],
\end{aligned}$$

where z_i be the index of the last entry in the ith group for $i = 1,\dots,k$; $z_0 = 0$ and $z_k = m-1$. Also partition $\Delta f_1,\dots,\Delta f_{m-1}$ into $\mathcal{F}_1,\dots,\mathcal{F}_k$ accordingly, where $\Delta f_i = f_{i+1} - f_i$ with $f_i = f(x_i)$. We use $s_i := z_i - z_{i-1}$ to denote the sizes of the groups for $i = 1,\dots,k$. Note that the indexing here is different from that in Section 2. We iteratively approximate the inverse Jacobian at the (z_i+1)st iterate for $i = 1,\dots,k$ by

$$G_{i+1} = G_i + (\mathcal{X}_i - G_i\mathcal{F}_i)V_i^T, \tag{31}$$

where $V_i^T\mathcal{F}_i = I$ for the secant condition. The update follows the formula of the generalized Broyden family (28). In the context of mixing, the base case is

$$G_1 = -\beta I,$$

where β is the mixing parameter. The next iterate is set by

$$x_{m+1} = x_m - G_{k+1} f_m. \tag{32}$$

The choice of V_i satisfying $V_i^T \mathcal{F}_i = I$ in (31) is left for free. In practice, we set V_i^T as $(\mathcal{X}_i^T G_i \mathcal{F}_i)^{-1} \mathcal{X}_i^T G_i$ or $(\mathcal{F}_i^T \mathcal{F}_i)^{-1} \mathcal{F}_i^T$, that minimize $E_1(G_{i+1}) = \|G_{i+1}^{-1} - G_i^{-1}\|_F$ or $E_2(G_{i+1}) = \|G_{i+1} - G_i\|_F$ as in (27) and (16), respectively. In both cases, $V_i \in \mathbb{R}^{n \times s_i}$ is in the form

$$V_i^T = M_i^{-1} N_i^T, \tag{33}$$

where $M_i \in \mathbb{R}^{s_i \times s_i}$ and $N_i \in \mathbb{R}^{n \times s_i}$. The two optimal choices are displayed in Table 1.

Table 1: Two optimal choices of $V_i^T = M_i^{-1} N_i^T$ in (31) satisfying $V_i^T \mathcal{F}_i = I$.

Type	I	II
M_i	$\mathcal{X}_i^T G_i \mathcal{F}_i$	$\mathcal{F}_i^T \mathcal{F}_i$
N_i^T	$\mathcal{X}_i^T G_i$	$\mathcal{F}_i^T$
Correspondence	(27)	(16)

Conceptually, there are two dimensions in this class of methods: the choices of update arguments such as those displayed in Table 1, and the partitioning of the available iterates. Four extreme subclasses are listed below.

1. If we partition the iterates into the groups each of which has one iterate, then the resulting subclass is Broyden's family.

2. If we put all available iterates into only one group, then we get Anderson's family.

3. If we use the Type-I update in Table 1, then the resulting subclass can be seen as an interpolation between Broyden's first method and the Type-I Anderson mixing.

4. If we use the Type-II update, then the resulting subclass can be seen as an interpolation between Broyden's second method and the standard Anderson mixing, a Type-II method.

Now we show how to avoid storing the potentially large matrices $G_i \in \mathbb{R}^{n \times n}$ for large-scale problems with $n \gg m$. The sketch is similar to that described in [22, section 5]. Let

$$E_i = \mathcal{X}_i - G_i \mathcal{F}_i. \tag{34}$$

Substituting (34) into (31) gets

$$G_i = G_{i-1} + E_{i-1} V_{i-1}^T$$

$$
\begin{aligned}
&= G_{i-2} + E_{i-2}V_{i-2}^T + E_{i-1}V_{i-1}^T \\
&\quad \vdots \\
&= G_1 + \sum_{j=1}^{i-1} E_j V_j^T \\
&= -\beta I + \sum_{j=1}^{i-1} E_j V_j^T \qquad (35)
\end{aligned}
$$

for $i = 2, \ldots, k+1$. The matrices G_i need not be explicitly stored. We need G_i only to compute $G_i\mathcal{F}_i$ in (34) and $G_{k+1}f_m$ in (32), and also for V_i if it depends on G_i. Substituting (35) into (34) gets

$$E_i = \mathcal{X}_i + \beta\mathcal{F}_i - \sum_{j=1}^{i-1} E_j(V_j^T\mathcal{F}_i) \qquad (36)$$

for $i = 1, \ldots, k$. The computation is economic for large-scale problems with $n \gg m$. The next iterate x_{m+1} in (32) can also be computed in a similar way:

$$x_{m+1} = x_m - G_{k+1}f_m = x_m + \beta f_m - \sum_{j=1}^{k} E_j(V_j^T f_m).$$

Using the Type-II update (see Table 1), the computation of V_i is straightforward from $\mathcal{F}_i$. On the other hand, the Type-I update involves G_i to compute V_i. By (35),

$$N_i^T = \mathcal{X}_i^T G_i = -\beta\mathcal{X}_i^T + \sum_{j=1}^{i-1}(\mathcal{X}_i^T E_j)V_j^T.$$

After obtaining N_i, we compute $M_i = N_i^T\mathcal{F}_i$ and then $V_i^T = M_i^{-1}N_i^T$ for $i = 1, \ldots, k$.

In practice, we fix the group sizes $s_1, s_2, \ldots$ from one Newton's iteration to another. As a result, $E_1, E_2, \ldots$ and $V_1, V_2, \ldots$ can be reused across iterations. Note that all $\mathcal{X}_i, \mathcal{F}_i, E_i, V_i$ are of the same size $n \times s_i$. We need $\mathcal{X}_i$ and $\mathcal{F}_i$ only for computing E_i in (36) and V_i displayed in Table 1. After E_i and V_i are computed, $\mathcal{X}_i$ and $\mathcal{F}_i$ are no longer required. In this respect $\mathcal{X}_i$ and $\mathcal{F}_i$ can share the storage with E_i and V_i for $i = 1, \ldots, k$. The storage required is listed below.

1. Two column vectors of size n for x_m and f_m.
2. An $n \times (m-1)$ matrix for $\mathcal{X}_1, \ldots, \mathcal{X}_k$ (shared with $E_1, \ldots, E_k$).
3. An $n \times (m-1)$ matrix for $\mathcal{F}_1, \ldots, \mathcal{F}_k$ (shared with $V_1, \ldots, V_k$).
4. For the Type-I update we also store the last group N_k, since its computation involves G_k.

3.4 The Eirola-Nevanlinna-like class

The nonlinear EN-like method proposed by Yang [29, chapter 7] is a Type-I member that corresponds to Broyden's first method. They have similar algorithmic structures, as shown Algorithms 1 and 2. Broyden's family was obtained by generalizing Broyden's methods to satisfy the secant condition. Likewise, we can obtain the *nonlinear EN-like family* by replacing (*) in Algorithm 2 by

$$G_{k+1} = G_k + (p_k - G_k q_k) v_k^T,$$

where $v_k^T q_k = 1$ for the secant condition.

Generalizing the EN-like family to a class of multisecant methods is also possible. We use the update formula in the form of (28):

$$G_k = G_{k-m} + (\mathcal{P}_k - G_{k-m}\mathcal{Q}_k) V_k^T, \tag{37}$$

where $\mathcal{P}_k = [p_{k-m} \ \cdots \ p_{k-1}]$, and $\mathcal{Q}_k = [q_{k-m} \ \cdots \ q_{k-1}]$ and $V_k^T \mathcal{Q}_k = I$ to satisfy the secant condition $G_k \mathcal{Q}_k = \mathcal{P}_k$. Note that p_i, q_i are defined in Algorithm 2 for $i = k-m, \ldots, k-1$. In particular, choosing

$$V_k = (\mathcal{P}_k^T G_{k-m} \mathcal{Q}_k)^{-1} (\mathcal{P}_k^T G_{k-m}) \tag{38}$$

results in a Type-I method, whereas choosing

$$V_k = (\mathcal{Q}_k^T \mathcal{Q}_k)^{-1} \mathcal{Q}_k^T \tag{39}$$

corresponds to the Type-II update. Note that m is the number of secant equations to be satisfied and also the rank of the update. Setting $m = 1$ results in the EN-like family.

Yang hinted at a limited-memory version of the nonlinear EN-like algorithm [29, page 124]. Here we give a solution for large-scale problems. For all algorithms in the EN-like class, it is straightforward to adapt the scheme in Section 3.3 to avoid explicitly storing the approximate inverse Jacobian G_k. Now we discuss the memory efficiency. Like the methods in Broyden-like class, only one secant equation is created per iteration using a method in the EN-like class. Hence the required memory is unchanged for the same number of iterations. The methods in the EN-like class require two function evaluations per iteration whereas the methods in the Broyden-like class need only one. Therefore, the required memory is halved for the same number of function evaluations.

For the same amount of function evaluations, the methods in the EN-like class take only half of the secant conditions of those in the Broyden-like class into account to approximate the Jacobian or inverse Jacobian. Intuitively, the EN-like class seems not to take the full advantage of the information of the available iterates. However, empirical evidences showed that the EN-like methods are often comparable to and sometimes faster than the Broyden-like methods in terms of number of function evaluations to converge. At this stage, little is known about theoretical convergence properties.

3.5 Hybrid methods

Broyden's first method (10) and Broyden's second method (8) are two optimal updating schemes to minimize $E_1(G_{k+1}) = \|G_{k+1}^{-1} - G_k^{-1}\|_F$ and $E_2(G_{k+1}) = \|G_{k+1} - G_k\|_F$ subject to the secant equation (7), respectively. Instead of using one updating method straight for all iterations, Martínez developed a scheme to choose the update (8) or (10) at each iteration [23]. This method outperforms Broyden's first and second methods in his small numerical experiments.

Likewise, using a multisecant algorithm presented in Section 3.3 or 3.4, we can choose either a Type-I or a Type-II update (displayed in Table 1) from one iteration to another. Inspired by Martínez' work [23], we develop a scheme to choose $V_i = (\mathcal{F}_i^T\mathcal{F}_i)^{-1}\mathcal{F}_i^T$ or $V_i = (\mathcal{X}_i^T G_i\mathcal{F}_i)^{-1}\mathcal{X}_i^T G_i$ for (31), the update of a Broyden-like method at each iteration.

To simplify the discussion, we assume that all groups of iterates have the same size fixed across iterations, except for the last group that may be smaller due to fewer iterates. Let i be the index of the last group. Since all updates satisfy the secant condition, we have $G_i\mathcal{F}_{i-1} = \mathcal{X}_{i-1}$ for $i > 1$ (i.e., there are at least two groups). When the last group $\mathcal{X}_i$ is smaller than the others, we trim some columns of $\mathcal{X}_{i-1}$ to form $\widehat{\mathcal{X}}_{i-1}$ of the same size of $\mathcal{X}_i$. We also trim the corresponding columns of $\mathcal{F}_{i-1}$ to get $\widehat{\mathcal{F}}_{i-1}$. In practice we keep the latest secant equations. By abuse of notation, we still write $\mathcal{X}_{i-1}$ for $\widehat{\mathcal{X}}_{i-1}$ and $\mathcal{F}_{i-1}$ for $\widehat{\mathcal{F}}_{i-1}$, and then $G_i\mathcal{F}_{i-1} = \mathcal{X}_{i-1}$.

Substituting $G_i\mathcal{F}_{i-1} = \mathcal{X}_{i-1}$ into the result of multiplying (31) by $\mathcal{F}_{i-1}$, we obtain

$$G_{i+1}\mathcal{F}_{i-1} - \mathcal{X}_{i-1} = (\mathcal{X}_i - G_i\mathcal{F}_i)V_i^T F_{i-1},$$

which represents the secant error $G_{i+1}\mathcal{F}_{i-1} - \mathcal{X}_{i-1}$ governed by $V_i^T F_{i-1}$, whose norm $\|V_i^T F_{i-1}\|$ can be used as a criterion to decide which V_i to use. More precisely, we choose $V_i = (\mathcal{F}_i^T\mathcal{F}_i)^{-1}\mathcal{F}_i^T$ if

$$\|(\mathcal{F}_i^T\mathcal{F}_i)^{-1}(\mathcal{F}_i^T\mathcal{F}_{i-1})\| < \|(\mathcal{X}_i^T G_i\mathcal{F}_i)^{-1}(\mathcal{X}_i^T\mathcal{X}_{i-1})\|,$$

otherwise we use $V_i = (\mathcal{X}_i^T G_i\mathcal{F}_i)^{-1}\mathcal{X}_i^T G_i$. To reduce cost we replace the above criterion by one based on the inequality:

$$\frac{\|\mathcal{F}_i^T\mathcal{F}_{i-1}\|}{\|\mathcal{F}_i^T\mathcal{F}_i\|} < \frac{\|\mathcal{X}_i^T\mathcal{X}_{i-1}\|}{\|\mathcal{X}_i^T G_i\mathcal{F}_i\|}\ . \tag{40}$$

Alternatively, we may consider the secant error in the form $G_{i+1}^{-1}\mathcal{X}_{i-1}^T - \mathcal{F}_{i-1}$, which leads to another decision criterion

$$\frac{\|\mathcal{F}_i^T\mathcal{F}_{i-1}\|}{\|\mathcal{F}_i^T G_i^{-1}\mathcal{X}_i\|} < \frac{\|\mathcal{X}_i^T\mathcal{X}_{i-1}\|}{\|\mathcal{X}_i^T\mathcal{X}_i\|}. \tag{41}$$

Any norm can be employed in (40) and (41). In practice we use the Frobenius norm.

Three remarks are given as follows. First, when a limited-memory algorithm is required for large-scale problems, we use (35) to compute $G_i\mathcal{F}_i$ in (40) so that

G_i does not need an explicit storage. On the other hand, the criterion (41) is discouraged due to the factor G_i^{-1}. Second, this scheme has no effect on the methods in Anderson's family, since all secant equations are put in one group and therefore the secant error is undefined. Third, this scheme is also applicable to the methods in the EN-like class.

Given a nonsingular square matrix A, an iterative method to solve a nonlinear system of equations is said to be invariant with respect to A in the range space, if applying this method to $f(x) = 0$ and $Af(x) = 0$ results in the same iterates $x_1, x_2, \ldots$. Likewise, a method is called invariant with respect to A in the domain space, if the iterates $\{x_i\}$ and $\{y_i\}$ resulting from applying this method for solving $f(x) = 0$ and $f(Ay) = 0$ respectively, are related by $x_i = Ay_i$, provided that $x_1 = Ay_1$. From (2) and (3), Newton's method is invariant with respect to linear transformation in both domain and range spaces, whereas a Type-I method with update (26) is invariant in the range space, and a Type-II method with update (16) is invariant in the domain space. The invariance property is not guaranteed with a hybrid method.

4 Practical details

The multisecant methods which we implemented and tested (see Section 5 on numerical experiments) can be categorized as one of Broyden-like or EN-like, as well as Type-I, Type-II, or hybrid update. A parameter of the method which is denoted by s, is the size of the groups of the secant equations. Several of these methods have appeared in the literature, as is displayed in Table 2. Note that $s = \infty$ in the table means that Anderson mixing implicitly forms an approximate inverse Jacobian subject to 'all' secant equations.

Table 2: Catalog of the multisecant methods.

Method	Ref.	Class	Type	Group size
Anderson mixing	[1]	Broyden	II	$s = \infty$
Broyden's first meth.	[7]	Broyden	I	$s = 1$
Broyden's second meth.	[7]	Broyden	II	$s = 1$
Martínez' scheme	[23]	Broyden	Hybrid	$s = 1$
Nonlinear EN-like meth.	[29]	EN	I	$s = 1$

The regularized Householder QR factorization with complete pivoting (see Section 4.1) has been utilized in our implementation to improve numerical stability.

The formulas of the multisecant methods involve inverting matrices. For Type-II methods, these matrices are symmetric positive semidefinite, whereas for Type-I methods no special structure is guaranteed (see, e.g., Table 1).

So far we have assumed that these matrices are nonsingular and therefore their inverses are well-defined. In practice if these matrices can become singular or ill-conditioned, so a careful treatment is required for numerical stability. The simplest solution is to perform restarting whenever a singular or ill-conditioned matrices is detected. Some of these issues are discussed next.

4.1 Type-II update

All the Type-II methods introduced in this article involve a factor of the form $(A^TA)^{-1}A^T$ (see, e.g., Table 1), or require to solve a least-squares linear system of the form (22) which can be solved by various means, the simplest of which is the normal equations (23).

When A does not have full column rank, it is nature to replace $(A^TA)^{-1}A^T$ from a Type-II update formula by a function that maps a given vector b to an output x that satisfies $A^TAx = A^Tb$.

Consider the least-squares systems to be solved in the rank-m update schemes seen earlier. For notational convenience we rewrite these least-squares systems (18) (22) as

$$\min_x \|b - Ax\|_2 \tag{42}$$

and the corresponding normal equations system (23) as

$$A^TAx = A^Tb, \tag{43}$$

where $A \in \mathbb{R}^{n\times m}$ with $n > m$. Solving the system in this way can cause instability in Anderson mixing when A is close to rank-deficient, a condition that is bound to take place near convergence.

The method of choice for solving very ill-conditioned least-squares system is a truncated form of the SVD, which is briefly discussed at the end of this section. Since the SVD is somewhat expensive, it is common to use instead a Householder QR approach with column pivoting.

The decomposition is denoted by $AP = QR$, where $Q \in \mathbb{R}^{n\times m}$ consists of orthonormal columns, $R \in \mathbb{R}^{m\times m}$ is upper triangular, and $P \in \mathbb{R}^{n\times n}$ is a permutation matrix for pivoting. The objective function (42) becomes $\|b - QR(P^Tx)\|_2$, the minimizer of which is identical to that of

$$\min_{\tilde{x}} \|Q^Tb - R\tilde{x}\|_2,$$

where $\tilde{x} = P^Tx$. In the case of Householder QR with pivoting, the matrix R is of the form

$$R = \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}, \tag{44}$$

where R_{11} is nonsingular and upper triangular and $R_{22} = 0$. Such a factorization is called a *Rank-revealing QR* [4]. Writing $\tilde{x} = \binom{\tilde{x}_1}{\tilde{x}_2}$ and $Q^Tb = \binom{\tilde{b}_1}{\tilde{b}_2}$ then the least-squares system to solve becomes

$$\min_{\tilde{x}_1,\tilde{x}_2} \left\| \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{bmatrix} - \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} \right\|_2 .$$

As is known, there are infinitely many solutions to the above problem, and the one with smallest 2-norm is given by

$$\tilde{x}_2 = 0; \qquad \tilde{x}_1 = R_{11}^{-1}\tilde{b}_1 \ .$$

The resulting solution is

$$\hat{x} = PR^{+}Q^{T}b, \tag{45}$$

where R^{+} is the Moore-Penrose pseudo-inverse of R. When A is ill-conditioned, the solution is 'regularized', meaning that R_{22} in (44) also contains the elements whose magnitudes are smaller than some tolerance. In practice we set the tolerance to be the product of the machine epsilon and the maximum magnitude of the diagonal elements in R.

Another option is to use the Singular Value Decomposition of A, denoted by $A = U\Sigma V^{T}$, where $U \in \mathbb{R}^{n\times m}$ consists of orthonormal columns, Σ is diagonal, and V is orthogonal. The pseudo-inverse solution to (43) is given by

$$\hat{x} = V\Sigma^{+}U^{T}b, \tag{46}$$

where Σ^{+} is from Σ with every nonzero entry replaced by its reciprocal. This is known to be the (unique) least-squares solution when A is of full rank and the solution of smallest 2-norm when A is not of full rank.

When A is ill-conditioned, the pseudo-inverse solution is often 'regularized', meaning that the singular values below a certain threshold are dropped. The solution obtained in this manner is more stable but also more computationally expensive than the solution (45) by the QR decomposition.

4.2 Type-I update

All Type-I methods involve inverting a generally nonsymmetric matrix, denoted by C, followed by a multiplication of C^{-1} and another matrix or vector (see, e.g., Table 1). When C is singular or ill-conditioned, the practical implementations of Section 4.1 for Type-II methods are also applicable here.

Consider $C^{-1}d$, where d is a column vector. When C is nonsingular, it is natural to choose y as any minimizer of $\|Cy-d\|_2$ to replace $C^{-1}d$. The smallest 2-norm minimizer is $\hat{y} = C^{+}d$, where C^{+} is the Moore-Penrose pseudo-inverse of C. This method is stable but somewhat expensive since the computation involves the SVD of C into $C = U\Sigma V^{T}$. The resulting formula is $\hat{y} = V\Sigma^{+}U^{T}d$, in the same form of (46).

Alternatively, we may use the Householder QR decomposition of C with column pivoting, denoted by $CP = QR$, where Q is orthogonal, R is upper triangular, and P is a permutation matrix for pivoting. The minimizer of $\|Cy-d\|_2$ from this decomposition is $\hat{y} = PR^{+}Q^{T}d$, in the same form of (45). The computation is less expensive than SVD.

In summary, the Type-I update formula involves a factor as the inverse of a matrix, denoted by C^{-1}, which in practice is replaced by $V\Sigma^{+}U^{T}$ using the SVD of $C = U\Sigma V^{T}$, or by $PR^{+}Q^{T}$ using the rank-revealing QR decomposition of C, $CP = QR$. For both numerical implementations, the decomposition is 'regularized' when C is ill-conditioned. This is as discussed in the case of Type-II update.

4.3 Restarting

Let f_{old} and f_{new} be two consecutive function values evaluated in sequence. If $\|f_{new}\|$ is too large relative to $\|f_{old}\|$, then the linear model or its approximation is not reliable, so the iteration should be restarted.

A restart simply consists of ignoring the previous directions and taking x_1 and f_1 to be the latest approximation and residual x_{old} and f_{old} obtained. Our codes perform a restart when $\|f_{old}\| < r\|f_{new}\|$, where r is a restarting parameter. Keeping r small prevents too many restarts and results in using more secant equations and speeding-up convergence. However, for challenging problems it may be necessary to increase the value of r to improve reliability. Our experiments seem to indicate that a good choice of the restarting factor is usually between 0.1 and 0.3.

One may ask why standard "global convergence strategies" such as damping, or trust-region techniques (see, e.g., [10, 24]), are not attempted here. The primary reason is that these methods tend to be expensive in the context of acceleration. For example, a linesearch back-tracking technique will often require several attempts before settling with a point that is very close to the previous one. The end result is that the convergence will be slow and the number of function evaluations is high. Recall that a function evaluation in electronic structures consists of one very expensive SCF iteration. Similar considerations apply for trust region techniques. In essence, the idea of restarting is another mechanism which decides on a course of action depending on whether or not the local linear model, upon which many of the global convergence strategies are built, is trusted. If it is not, then fewer (or no) previous directions are kept.

5 Experiments

We have tested the multisecant algorithms described in this article on a variety of problems. Here we report the results of the experiments on a variant of Bratu's problem (see Section 5.1), two problems using RSDFT which is a MATLAB implementation of a real-space density function theory method in electronic structure (see Section 5.2), and two problems using PARSEC, the Pseudopotential Algorithm for Real-Space Electronic Calculations which is Fortran-90 code for electronic structure (see Section 5.3).

5.1 A variant of Bratu's problem

Consider the partial differential equation

$$u_{xx} + u_{yy} + \alpha u_x + \lambda e^u = 0, \qquad (47)$$

where u is a function of $(x, y) \in [0, 1]^2$, and α, λ are two scalars. This equation differs from the standard Bratu problem by the addition of the convection term αu_x which destroys symmetry. Dirichlet boundary conditions are imposed such that $u(x, y) = 0$ for (x, y) on the boundary of the unit square domain.

We solved this problem by a finite difference method. More precisely, we used a matrix $U = [u_{ij}] \in \mathbb{R}^{m\times m}$ to present the approximate solution, where $u_{ij} \approx u(ih, jh)$, $h = \frac{1}{m+1}$. Then the finite difference approximations of u_{xx}, u_{yy}, u_x are well-defined for $(x, y) = (ih, jh)$, $i, j = 1, \ldots, m$. Substituting these approximations into (47), we obtain a system of nonlinear equations of U involving $m \times m$ variables, denoted by $F(U) = 0$.

Table 3: Number of function evaluations to achieve $\|F\|_2 < 10^{-8}$, $n = 400$.

Class	Type	$s = 1$	Best Performance	$s = \infty$
Broyden	I	91	65 ($s = 21\!:\!26, 28, 30\!:\!55$)	79
	Hybrid-I	71	65 ($s = 16\!:\!18, 21\!:\!29, 32\!:\!55$)	
	II	71	65 ($s = 16\!:\!18, s \geq 21$)	65
	Hybrid-II	71	65 ($s = 16\!:\!18, 21\!:\!29, s \geq 32$)	
EN	I	115	69 ($s = 17, 20$)	79
	Hybrid-I	77	69 ($s = 16, 17, 19, 22, 23, 25, 27, 28$)	
	II	78	69 ($s = 17, 20, 21, 27, 32, s \geq 34$)	69
	Hybrid-II	78	69 ($s = 17, 20, 21, 27, 32, s \geq 34$)	

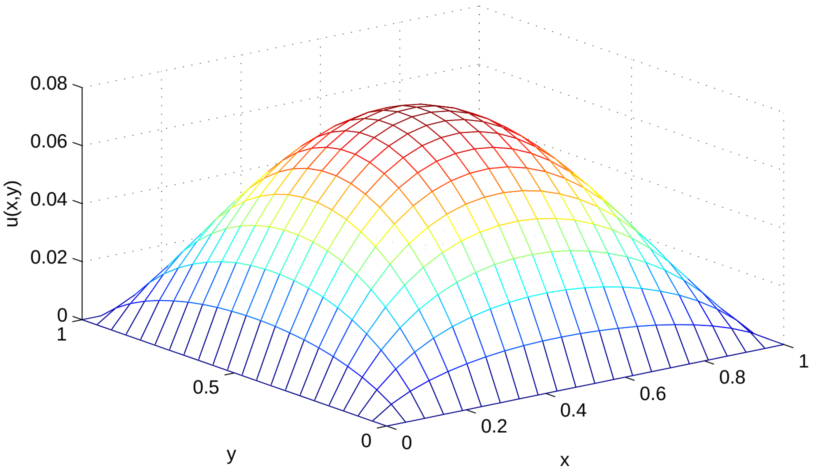

Figure 1: Computed solution of a variant of Bratu's problem (47).

Table 3 lists the numbers of function evaluations to achieve $\|F\|_2 < 10^{-8}$ for the test with $\alpha = 1$, $\lambda = 1$, and $m = 20$, resulting in a problem of size $n = 20 \times 20 = 400$. The restarting factor used was $r = 0.1$. This problem is not well-scaled; a good choice for the mixing parameter was $\beta = 5 \times 10^{-4}$ for both classes of methods. As discussed in Section 3.5, the hybrid type of update chooses Type-I or Type-II update at each iteration depending on which update results in a smaller secant error, which is defined only when the number of iterations is larger than the group size of secant equations s. We use Hybrid-I/II to indicate that Type-I/II update is employed whenever the secant error is undefined, respectively. The 3D-plot of the computed solution is given in Figure 1.

We further investigated the case of 'finer' grids $n = 100 \times 100 = 10,000$, where $\beta = 2 \times 10^{-5}$ was a good choice of the mixing parameter. The restarting factor used was $r = 0.3$. For each type of update we tried the group sizes $s = 1, 2, 5, 10, 20, 50, 100, 200, \infty$. The numbers of function evaluations to achieve $\|F\|_2 < 10^{-6}$ were displayed in Table 4. Both Broyden's first method and the Type-I nonlinear EN-like method did not converge within 500 iterations and are marked as 'N/A'.

Table 4: Number of function evaluations to achieve $\|F\|_2 < 10^{-6}$, $n = 10,000$.

Class	Type	$s = 1$	Best Performance	$s = \infty$
Broyden	I	N/A	277 ($s = 200$)	408
	Hybrid-I	306	273 ($s = 100, 200$)	
	II	300	273 ($s = 50, 100, 200, \infty$)	273
	Hybrid-II	307	273 ($s = 50, 100, 200, \infty$)	
EN	I	N/A	290 ($s = 100$)	396
	Hybrid-I	332	286 ($s = 100$)	
	II	325	285 ($s = 50, 200, \infty$)	285
	Hybrid-II	332	285 ($s = 50, 200, \infty$)	

In both tests shown in Tables 3 and 4, a Type-II method outperformed its corresponding Type-I method, and the hybrid scheme improved the Type-I methods. For the Type-I methods a suitable group size s may outperform both end methods (i.e., $s = 1, \infty$).

5.2 RSDFT

Here we present experimental results on two problems using RSDFT, a MATLAB implementation of a real-space density functional theory approach to electronic structure. The first problem involves $n = 157,464$ variables (Na atom), whereas the second has $n = 79,507$ variables (Si atom). How close the current estimate is to the solution is measured by the relative residual 2-norm, i.e., $\frac{\|f(x_i)\|_2}{\|x_i\|_2}$.

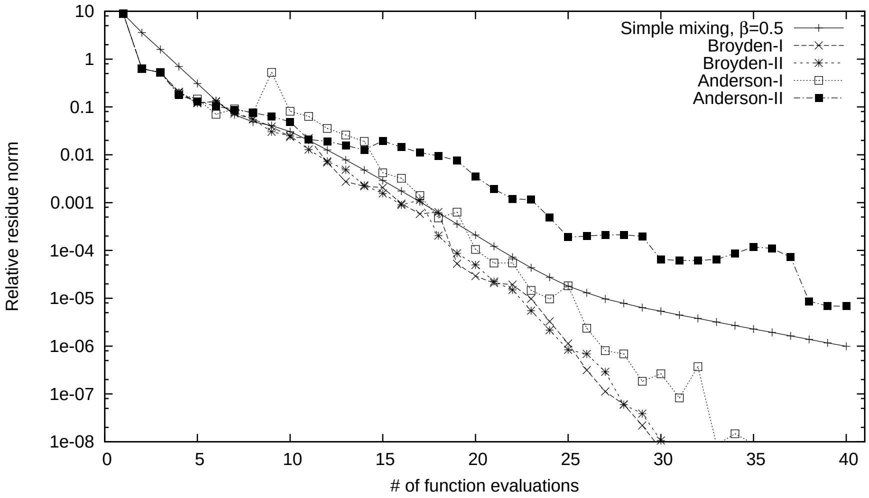

Figure 2: Result of methods in Broyden-like class, RSDFT, Na atom, $\beta = 1.0$.

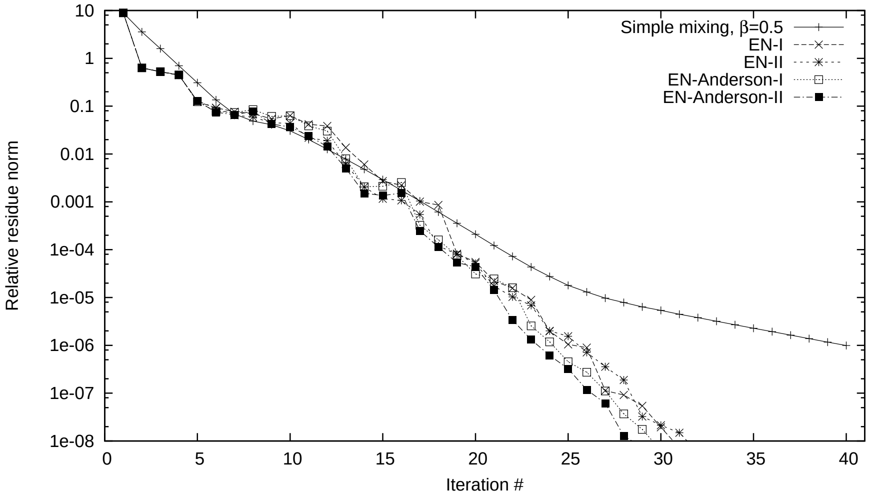

Figure 3: Results of methods in EN-like class, RSDFT, Na atom, $\beta = 1.0$.

In all plots we use Broyden-I and Broyden-II to denote Broyden's first and second methods, respectively. As discussed in Section 3.2, Anderson mixing is a Type-II method, which has a Type-I variant. They are denoted by Anderson-II and Anderson-I, respectively. As discussed in Section 2.4, the nonlinear EN-like method proposed by Yang [29, chapter 7] is a Type-I method, which has a Type-II variant. We denote them by EN-I and EN-II, respectively. The methods corresponding to the Anderson-I/II in the EN-like class (i.e., $s = \infty$) are denoted by EN-Anderson-I/II, respectively. For these two problems the performance of the intermediate methods in both classes was comparable and therefore is not reported. Results of simple mixing, defined in (25), are also included for comparison.

For the first problem (Na atom), the mixing parameter $\beta = 1.0$ was a good choice for both classes of multisecant methods, while simple mixing worked well with $\beta = 0.5$. We used the restarting factor $r = 0.3$. Figures 2 and 3 show the result of tests with the methods in Broyden-like and EN-like class, respectively. In this experiment both Broyden's first and second methods exhibited good performance, whereas Anderson mixing is less efficient and it is outperformed by simple mixing.

The Type-I variant of Anderson mixing, Anderson-I, performed significantly better. Compared with Broyden's methods, the nonlinear EN-like algorithms (EN-I/II) saved 50% of memory without slowing down the convergence. The EN-Anderson algorithms also improved the convergence compared with the Anderson algorithms, in addition to saving 50% memory in the mixing.

For the second problem (Si atom), the mixing parameter $\beta = 0.5$ was a good choice for both classes of multisecant methods, while simple mixing worked well with $\beta = 0.3$. Figures 4 and 5 show the result of tests with the methods in Broyden-like and EN-like class, respectively. In this experiment both Broyden's first and second methods performed well, whereas Anderson algorithms were less efficient but still better than simple mixing. Compared with Broyden's methods, nonlinear EN-like algorithms (EN-I/II) saved 50% memory in mixing without slowing down the convergence. EN-Anderson algorithms not only outperformed Anderson mixing but also saved 50% memory in mixing.

5.3 PARSEC

PARSEC [8, 18, 25] is a comprehensive electronic structure calculation code written in FORTRAN-90 which was developed over a period of more than a decade. It uses a real-space finite difference approach and employs pseudopotentials. The simulation in PARSEC employs Density Functional Theory and as such the calculation consists of a self-consistent (SCF) iteration with the potential. In many cases, self-consistency is not too difficult to achieve, requiring 10 to 20 iterations to converge. Metallic systems, such as iron clusters, lead to much more difficult convergence, requiring on occasion a few hundred iterations. Iron clusters have been the subject of an extensive recent study in [27].

In the following we report on a few experiments with two such problems: Fe1 and Fe43, which involve $n = 118,238$ and $n = 220,490$ variables, respectively.

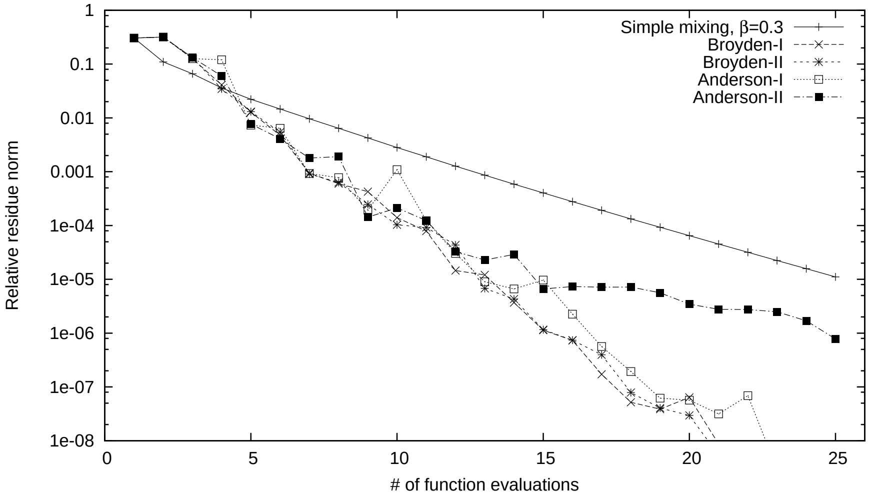

Figure 4: Result of methods in Broyden-like class, RSDFT, Si atom, $\beta = 0.5$.

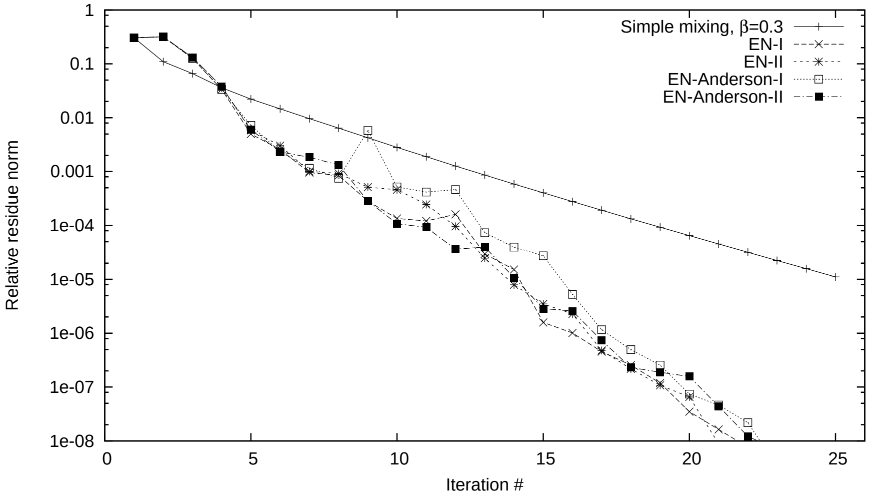

Figure 5: Result of methods in EN-like class, RSDFT, Si atom, $\beta = 0.5$.

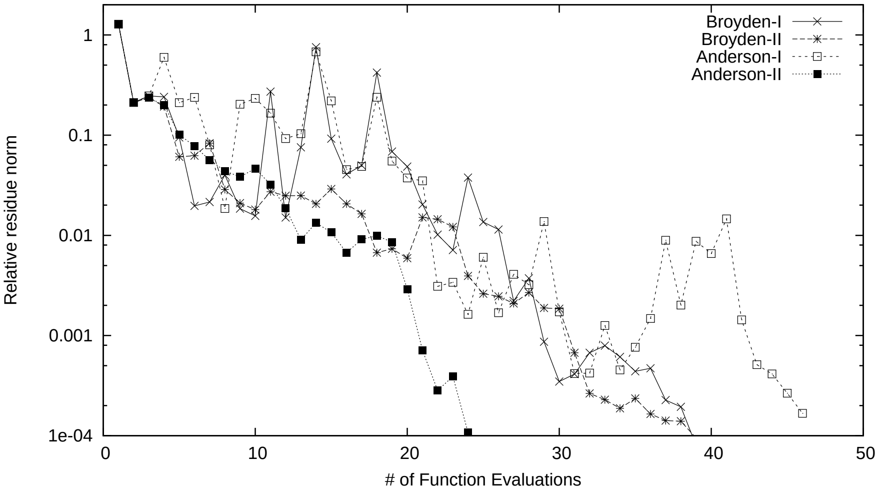

Figure 6: Result of methods in Broyden-like class, PARSEC, Fe1, $\beta = 0.1$.

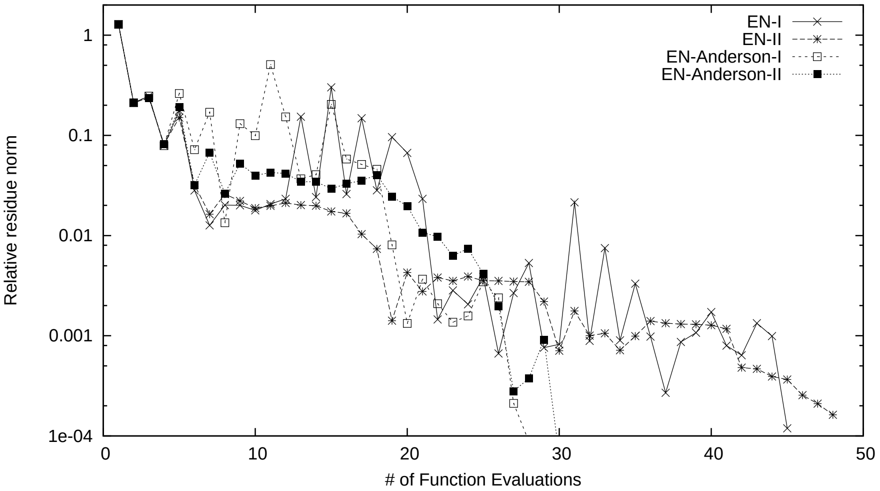

Figure 7: Result of methods in EN-like class, PARSEC, Fe1, $\beta = 0.1$.

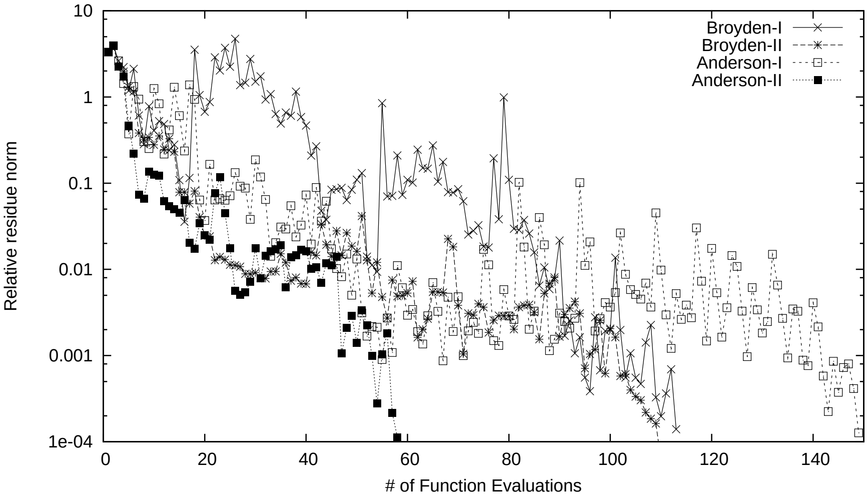

Figure 8: Result of methods in Broyden-like class, PARSEC, Fe43, $\beta = 0.1$.

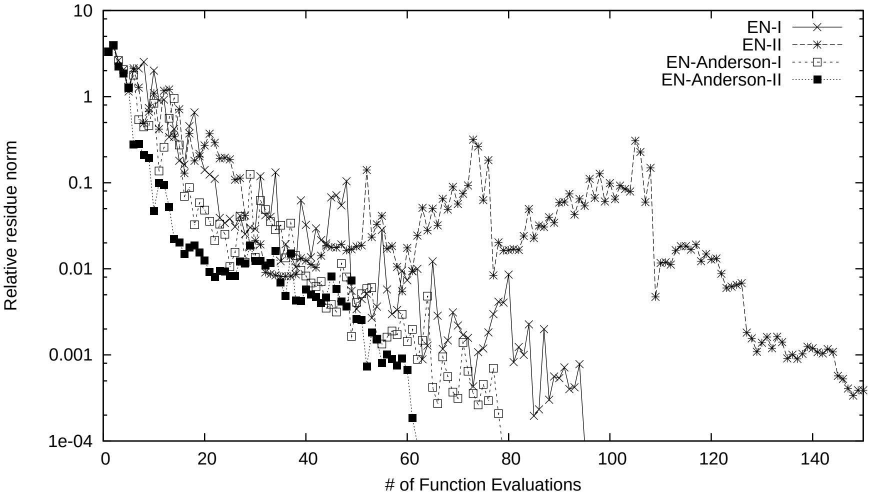

Figure 9: Result of methods in EN-like class, PARSEC, Fe43, $\beta = 0.1$.

Note that metallic systems include spin as a variable (unlike non-magnetic materials). This means that each problem consists of two coupled sub-problems of size $n/2$ each (one for spin-up and the other for spin down).

Figures 6 and 7 show the results of the Fe1 problem, whereas Figures 8 and 9 show the results of the Fe43 problem, by the methods in Broyden-like and EN-like classes, respectively. In both tests simple mixing, defined in (25), was not useful. We follow the notation used in Figures 2–5 in Section 5.2. The mixing parameter used was $\beta = 0.1$, and the restarting factor was set as $r = 0.3$, except that in the case EN-I applied to the Fe1 problem, we used $r = 0.4$ to improve convergence.

For both Fe1 and Fe43, Anderson mixing (a Type-II method) achieved the best performance. On average, the EN-like methods, performed as well as the Broyden-like methods. Restarting sometimes played an important role to ensure convergence. The performances of the intermediate methods and the hybrid methods were generally in a comparable range, so details are omitted.

6 Concluding remarks

We presented two classes of multisecant methods, from observing the mathematical difference between Broyden's methods [7] and Anderson mixing [1]: Broyden's methods updates its approximate Jacobian or inverse Jacobian from the previous iterate subject to the latest 'one' secant equation, whereas Anderson mixing implicitly forms an approximate inverse Jacobian from 'all' available secant equations [12]. By allowing a flexible number of secant equations (denoted by s), we obtain a Broyden-like class of multisecant methods. Likewise, we generalized the nonlinear EN-like method by Yang [29, chapter 7] to the EN-like class of multisecant methods.

In the Broyden-like class of multisecant methods, Broyden's family is a subclass and Anderson mixing is a particular member. Each method in the EN-like class, has a corresponding method in the Broyden-like class, and vice versa. In particular, the nonlinear EN-like method by Yang corresponds to Broyden's first method.

We call the update formulas to minimize the change of the approximate Jacobian (and inverse Jacobian) the Type-I (and Type-II) methods, respectively. This feature inspired some methods from existing ones. For example, Anderson mixing is a Type-II method, and we gave its Type-I variant. The nonlinear EN-like method by Yang is a Type-I method, and we presented its Type-II variant.

Broyden's methods and Anderson mixing are at two extreme ends ($s = 1$ and $s = \infty$, respectively) in the Broyden-like class. Experiments showed that sometimes the intermediate methods can outperform the two end methods (e.g., Tables 3 and 4 for the variant of Bratu's problem).

We also generalized Martínez' scheme [23], to choose the Type-I or Type-II update depending on the secant errors at each iteration, resulting in the hybrid methods presented in Section 3.5. In practice, hybrid methods may improve on the worse of the Type-I and Type-II methods (e.g., Tables 3 and 4 for the variant of Bratu's problem).

No method can outperform the others in all cases:

- Sometimes Broyden's methods work better (e.g., Figure 2 for Na problem); sometimes Anderson mixing works better (e.g., Tables 3 and 4 for the variant of Bratu's problem).
- While methods in the EN-like class showed better stability for the Na problem (see Figures 2 and 3), methods in the Broyden-like class performed better for the variant of Bratu's problem (see Tables 3 and 4). The two classes of methods showed comparable performances for the Fe1 and Fe43 problems (see Figures 6–9).
- Anderson mixing worked best for both Fe1 and Fe43 problems (see Figures 6–9). It was also the worst multisecant method for both Na and Si problems (see Figures 2–5).

Experience with specific applications, and knowledge of the problems may help decide which method and which parameters (e.g., mixing parameters and restarting factor) to use.

References

[1] D. G. Anderson. Iterative procedures for non-linear integral equations. *Assoc. Comput. Mach.*, 12(547):547–560, 1965.

[2] J. G. P. Barnes. An algorithm for solving nonlinear equations based on the secant method. *Computer Journal*, 8:66–72, 1965.

[3] M. Bierlaire and F. Crittin. Solving noisy, large-scale fixed point problems and systems of nonlinear equations. *Transportation Sci.*, 40(1):44–63, February 2006.

[4] A. Björk. *Numerical Methods for Least-Squares Problems.* SIAM publications, Philadelphia, PA, 1996.

[5] P. N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comput.*, 11(3):450–481, 1990.

[6] P. N. Brown and Y. Saad. Convergence theory of nonlinear Newton-Krylov algorithms. *SIAM J. Optim.*, 4(2):297–330, 1994.

[7] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, 19:577–593, 1965.

[8] J. R. Chelikowsky, L. Kronik, I. Vasiliev, M. Jain, and Y. Saad. Using real space pseudopotentials for the electronic structure problem. In C. Le Bris and P. G. Ciarlet, editors, *Handbook for numerical analysis, volume X*, pages 613–635. Elsevier Science, 2003.

[9] J.R. Chelikowsky and S.G. Louie. *Quantum Theory of Real Materials.* Kluwer Press, Amsterdam, 1996.

[10] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall, Englewood Cliffs, NJ, 1983.

[11] T. Eirola and O. Nevanlinna. Accelerating with rank-one updates. *J. of Linear Algebra with Applications*, 121:511–520, 1989. Linear algebra and applications (Valencia, 1987).

[12] V. Eyert. A comparative study on methods for convergence acceleration of iterative vector sequences. *J. Computational Phys.*, 124:271–285, 1996.

[13] D. M. Gay and R. B. Schnabel. Solving systems of nonlinear equations by Broyden's method with projected updates. In O. Mangasarian, R. Meyer, and S. Robinson, editors, *Nonlinear Programming 3*, pages 245–281. Academic Press, New York, 1978.

[14] W. B. Gragg and G. W. Stewart. A stable variant of the secant method for solving nonlinear equations. *SIAM J. Numer. Anal.*, 13(6):889–903, 1976.

[15] Jr. J. E. Dennis and J. J. Moré. Quasi-Newton methods: Motivation and theory. *SIAM Rev.*, 19(1):46–89, 1977.

[16] D. D. Johnson. Modified Broyden's method for accelerating convergence in self-consistent calculations. *Phys. Rev. B*, 38(18):12807–12813, December 1988.

[17] J. E. Dennis Jr. and E. S. Marwil. Direct secant updates of matrix factorizations. *Math. Comp.*, 38(158):459–476, 1982.

[18] L. Kronik, A. Makmal, M. L. Tiago, M. M. G. Alemany, M. Jain, X. Huang, Y. Saad, and J. R. Chelikowsky. PARSEC the pseudopotential algorithm for real-space electronic structure calculations: recent advances and novel applications to nano-structure. *Phys. Stat. Sol. (B)*, 243(5):1063 1079, 2006.

[19] J. M. Martínez. A quasi-Newton method with modification of one column per iteration. *Computing*, 33(3-4):353–362, 1984.

[20] J. M. Martínez. A family of quasi-Newton methods for nonlinear equations with direct secant updates of matrix factorizations. *SIAM J. Numer. Anal.*, 27:1034–1049, 1990.

[21] J. M. Martínez. An inverse column-updating method for solving large-scale nonlinear systems of equations. *Optim. Methods Software*, 1:129–140, 1992.

[22] J. M. Martínez. Practical quasi-Newton methods for solving nonlinear systems. *J. Comput. Appl. Math.*, 124:98–121, 2000.

[23] J. M. Martínez and L. S. Ochi. Sobre dois métodos de Broyden. *Nat. Apl. Comput.*, 1:135–141, 1982.

[24] W. Murray P. E. Gill and M. Wright. *Practical Optimization*. Academic Press, New York, 1981.

[25] Y. Saad, Y. Zhou, C. Bekas, M. L. Tiago, and J. R. Chelikowsky. Diagonalization methods in PARSEC. *Physica Status Solidi (b)*, 243(9):2188–2197, 2006.

[26] G. P. Srivastava. Broyden's method for self-consistent field convergence acceleration. *Phys. Rev. A*, 17:L317–L321, 1984.

[27] M. L. Tiago, Y. Zhou, M. M. G. Alemany, Y. Saad, and J. R. Chelikowsky. The evolution of magnetism in iron from the atom to the bulk. *Physical Review Letters*, 97:147201–4, 2006.

[28] D. Vanderbilt and S. G. Louie. Total energies of diamond (111) surface reconstructions by a linear combination of atomic orbitals method. *Phys. Rev. B*, 30(10):6118–6130, 1984.

[29] U. M. Yang. *A family of preconditioned iterative solvers for sparse linear systems*. PhD thesis, University of Illinois at Urbana-Champaign, Department of Computer Science, 1995.