SIAM J. NUMER. ANAL. © 2011 Society for Industrial and Applied Mathematics
Vol. 49, No. 4, pp. 1715–1735

ANDERSON ACCELERATION FOR FIXED-POINT ITERATIONS[*]

HOMER F. WALKER[†] AND PENG NI[†]

Abstract. This paper concerns an acceleration method for fixed-point iterations that originated in work of D. G. Anderson [*J. Assoc. Comput. Mach.*, 12 (1965), pp. 547–560], which we accordingly call *Anderson acceleration* here. This method has enjoyed considerable success and wide usage in electronic structure computations, where it is known as *Anderson mixing*; however, it seems to have been untried or underexploited in many other important applications. Moreover, while other acceleration methods have been extensively studied by the mathematics and numerical analysis communities, this method has received relatively little attention from these communities over the years. A recent paper by H. Fang and Y. Saad [*Numer. Linear Algebra Appl.*, 16 (2009), pp. 197–221] has clarified a remarkable relationship of Anderson acceleration to quasi-Newton (secant updating) methods and extended it to define a broader *Anderson family* of acceleration methods. In this paper, our goals are to shed additional light on Anderson acceleration and to draw further attention to its usefulness as a general tool. We first show that, on linear problems, Anderson acceleration without truncation is "essentially equivalent" in a certain sense to the generalized minimal residual (GMRES) method. We also show that the *Type* 1 variant in the Fang–Saad Anderson family is similarly essentially equivalent to the Arnoldi (full orthogonalization) method. We then discuss practical considerations for implementing Anderson acceleration and illustrate its performance through numerical experiments involving a variety of applications.

Key words. acceleration methods, fixed-point iterations, generalized minimal residual method, Arnoldi (full orthogonalization) method, iterative methods, expectation-maximization algorithm, mixture densities, alternating least-squares, nonnegative matrix factorization, domain decomposition

AMS subject classifications. 65H10, 65F10

DOI. 10.1137/10078356X

1. Introduction.

We consider a general fixed-point problem and the associated fixed-point iteration as follows.

Problem FP. Given $g : \mathbb{R}^n \to \mathbb{R}^n$, solve $x = g(x)$.

ALGORITHM FPI. FIXED-POINT ITERATION.

```
Given $x_0$.
For $k = 0, 1, \ldots$
    Set $x_{k+1} = g(x_k)$.
```

Fixed-point problems abound in computational science and engineering, although they may not always be regarded or treated as such. There is a natural duality between Problem FP and a nonlinear-equations problem: solve $f(x) \equiv g(x) - x = 0$. Many problems that could be viewed as fixed-point problems are instead posed as nonlinear-equations problems in order to take advantage of the many well-refined algorithms for the latter problems. Most of these algorithms are modeled on Newton's method, and their major strength is rapid (typically superlinear or quadratic) local convergence. Additionally, robustness is often enhanced by globalization procedures that improve the likelihood of convergence from arbitrary initial approximations. (See, e.g., Dennis

[*]Received by the editors January 22, 2010; accepted for publication (in revised form) May 17, 2011; published electronically August 16, 2011. A preliminary version of this paper appeared as Worcester Polytechnic Institute Mathematical Sciences Department Research Report MS-9-21-45. This work was supported in part by U.S. National Science Foundation grant DMS 0915183 and by U.S. Department of Energy award DE-SC0004880.

http://www.siam.org/journals/sinum/49-4/78356.html

[†]Department of Mathematical Sciences, Worcester Polytechnic Institute, Worcester, MA 01609-2280 (walker@wpi.edu, davisni@wpi.edu).

and Schnabel [14] or Kelley [26] for extensive treatments of these methods.) Notwithstanding the advantages of these sophisticated algorithms, there are often overriding reasons for casting problems in fixed-point form and employing fixed-point iteration to solve them numerically.

The major concern usually associated with fixed-point iteration is that the iterates may not converge or, if they do, exhibit only linear convergence, which may be unacceptably slow. Acceleration methods can potentially alleviate slow convergence and, in some cases, divergence as well (see the numerical experiments in section 5). Our interest here is in a particular acceleration method originating in work of Anderson [1],[1] which we refer to as *Anderson acceleration* and formulate as follows.

Algorithm AA. Anderson Acceleration.

```
Given $x_0$ and $m \geq 1$.
Set $x_1 = g(x_0)$.
For $k = 1, 2, \ldots$
    Set $m_k = \min\{m, k\}$.
    Set $F_k = (f_{k-m_k}, \ldots, f_k)$, where $f_i = g(x_i) - x_i$.
    Determine $\alpha^{(k)} = (\alpha_0^{(k)}, \ldots, \alpha_{m_k}^{(k)})^T$ that solves
```

$$\min_{\alpha=(\alpha_0,\ldots,\alpha_{m_k})^T} \|F_k\alpha\|_2 \text{ s.t. } \sum_{i=0}^{m_k} \alpha_i = 1. \tag{1.1}$$

```
    Set $x_{k+1} = \sum_{i=0}^{m_k} \alpha_i^{(k)} g(x_{k-m_k+i})$.
```

In practice, each m_k may be further modified, e.g., to maintain acceptable conditioning of F_k (see section 4). The original formulation in [1] allows a more general step,

$$x_{k+1} = (1 - \beta_k) \sum_{i=0}^{m_k} \alpha_i^{(k)} x_{k-m_k+i} + \beta_k \sum_{i=0}^{m_k} \alpha_i^{(k)} g(x_{k-m_k+i}), \tag{1.2}$$

where $\beta_k > 0$ is a relaxation parameter. It is appropriate here to consider only $\beta_k = 1$, which gives the step in Algorithm AA.[2] Also in [1], the least-squares problem (1.1) is formulated as an equivalent unconstrained least-squares problem:

$$\min_{(\theta_1,\ldots,\theta_{m_k})} \left\| f_k + \sum_{i=1}^{m_k} \theta_i (f_{k-i} - f_k) \right\| 2.$$

The form (1.1) seems better aligned with current views of the algorithm; however, neither form may be preferred in an actual implementation (see section 4).

A number of acceleration methods have been proposed and studied over the years. Within the mathematics and numerical analysis communities, most attention has been given to the *vector-extrapolation methods*, principally the *polynomial methods*, which include the reduced-rank extrapolation (RRE), minimal-polynomial extrapolation (MPE), and modified minimal-polynomial extrapolation (MMPE) methods, and to the *vector* and *topological ε-algorithms*. The literature on these methods is vast,

[1] Methods that appear to be mathematically equivalent have been derived independently by others in the context of specific applications, namely, by Carlson and Miller [11] and Miller [30] for accelerating modified-Newton (chord) iterations in method-of-lines applications, and by Washio and Oosterlee [43] and Oosterlee and Washio [33] for accelerating nonlinear multigrid methods.

[2] Choosing $\beta_k = 1$ is necessary for the theoretical results in sections 2–3 and sufficient for the experiments in section 5. For examples of other choices that may be useful in practice, see the experiments in Fang and Saad [19].

and a complete list of references would be inappropriate here. For overviews and references to original sources, see the book by Brezinski and Redivo-Zaglia [4] and also a relatively recent survey by Brezinski [3]; the survey by Jbilou and Sadok [25] and earlier work by those authors [23], [24]; and the survey by Smith, Ford, and Sidi [42].

Anderson acceleration differs mathematically from the vector-extrapolation methods and ε-algorithms and, in fact, belongs to a distinct category of methods developed by a different community of researchers. This category includes structurally similar methods developed for electronic structure computations, notably those by Pulay [35], [36] (known as *Pulay mixing* within the materials community and *direct inversion on the iterative subspace*, or DIIS, among computational chemists), and a number of other "mixing" methods; see Kresse and Furthmüller [28], [29], Le Bris [5], and Yang et al. [45] for overviews. (In these applications, "mixing" derives from "charge mixing," and Anderson acceleration is known as "Anderson mixing.") Also in this category are the mathematically related methods of Eirola and Nevanlinna [16] and Yang [46] and certain other methods based on quasi-Newton updating. A recent paper by Fang and Saad [19] summarizes these and a great deal of related work in addition to providing a number of new developments. Of particular relevance here, Fang and Saad [19] clarify a remarkable relationship of Anderson acceleration to quasi-Newton updating (specifically *multisecant* updating) originally noted by Eyert [18] and proceed from this relationship to define a broad family of acceleration methods that includes Anderson acceleration. We return to this subject in section 3.

Our goal in this paper is to shed additional light on Anderson acceleration and its behavior. We also hope to draw further attention to the method as a useful general tool for accelerating fixed-point iterations. In section 2, we show that, on linear problems, it is "essentially equivalent" to the generalized minimal residual (GMRES) method (Saad and Schultz [40]) in a sense given in Theorem 2.2.[3] In section 3, we review the relationship of Anderson acceleration to multisecant updating methods outlined in [19], [18] and, for further perspectives, show that a particular member of the *Anderson family* of methods defined in [19] (the *Type* I method) is, on linear problems, essentially equivalent in the same sense to the Arnoldi method (also known as the full orthogonalization method, or FOM) (Saad [38]). In section 4, we discuss practical considerations for implementing Anderson acceleration. In section 5, we report on numerical experiments that show the performance of the method in a variety of applications. In section 6, we provide a concluding summary.

In the following, the Euclidean norm $\|\cdot\|_2$ on $\mathbb{R}^n$ is used throughout. One can easily extend the results to allow any inner-product norm on $\mathbb{R}^n$. The Frobenius norm on matrices is denoted by $\|\cdot\|_F$.

2. Anderson acceleration on linear problems.

In this section, we consider the case in which g in Problem FP is linear; specifically, we assume that $g(x) = Ax+b$ for some $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. The rationale for Anderson acceleration is readily apparent in this case: At the kth step, with $\sum_{i=0}^{m_k} \alpha_i^{(k)} = 1$, one has

$$x_{k+1} = \sum_{i=0}^{m_k} \alpha_i^{(k)} g(x_{k-m_k+i}) = g\left(\sum_{i=0}^{m_k} \alpha_i^{(k)} x_{k-m_k+i}\right) = g(x_{\min}),$$

where $x_{\min} \equiv \sum_{i=0}^{m_k} \alpha_i^{(k)} x_{k-m_k+i}$ has minimal fixed-point residual within the affine subspace containing $\{x_{k-m_k}, \ldots, x_k\}$.

[3] The authors of [11], [30], [43], [33] also recognized very close relationships of their methods to GMRES but did not provide precise details.

Our main goal is to establish a certain equivalence between Anderson acceleration without truncation, i.e., with $m_k = k$ for each k, and GMRES applied to the equivalent linear system $(I-A)x = b$ starting with the same x_0. Our notation is as follows: For each j, x_j^{GMRES} and $r_j^{\text{GMRES}} \equiv b-(I-A)x_j^{\text{GMRES}}$ denote the jth GMRES iterate and residual, respectively, and $\mathcal{K}_j \equiv \text{span}\,\{r_0^{\text{GMRES}}, (I-A)r_0^{\text{GMRES}}, \dots, (I-A)^{j-1}r_0^{\text{GMRES}}\}$ denotes the jth Krylov subspace generated by $(I-A)$ and r_0^{GMRES}. When helpful, we also denote the jth iterate of Anderson acceleration by x_j^{AA}. Our major assumptions are as follows.

Assumption 2.1.

- $g(x) = Ax + b$ for $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$.
- Anderson acceleration is not truncated, i.e., $m_k = k$ for each k.
- $(I-A)$ is nonsingular.
- GMRES is applied to $(I-A)x = b$ with initial point $x_0 = x_0^{\text{AA}}$.

Our main result is the following theorem, which shows that, with the above assumption and an additional "nonstagnation" assumption, Anderson acceleration and GMRES are "essentially equivalent" in the sense that the iterates produced by either method can be readily obtained from those of the other method.[4] It follows that Anderson acceleration with truncation, i.e., with $m_k = \min\{m, k\}$, can be regarded as essentially equivalent in the same sense to truncated GMRES (Saad [39]). Additionally, one can formulate a "restarted" variant of Algorithm AA, in which the method proceeds without truncation for m steps and then is restarted using $\sum_{i=0}^m \alpha_i^{(m)} x_i^{\text{AA}}$ as the new x_0. Under the assumptions of the theorem, this new x_0 is x_m^{GMRES}, and so this restarted variant of Algorithm AA is essentially equivalent in the same sense to GMRES(m) (Saad and Schultz [40]).

THEOREM 2.2. *Suppose that Assumption* 2.1 *holds and that, for some* $k > 0$, $r_{k-1}^{\text{GMRES}} \neq 0$ *and also* $\|r_{j-1}^{\text{GMRES}}\|_2 > \|r_j^{\text{GMRES}}\|_2$ *for each* j *such that* $0 < j < k$. *Then* $\sum_{i=0}^k \alpha_i^{(k)} x_i^{\text{AA}} = x_k^{\text{GMRES}}$ *and* $x_{k+1}^{\text{AA}} = g(x_k^{\text{GMRES}})$.[5]

Remark 2.3. The theorem is phrased to allow the case $k = 1$, in which $\{j : 0 < j < k\} = \emptyset$, and also the possibility that GMRES stagnates at the kth step, i.e., that $\|r_{k-1}^{\text{GMRES}}\|_2 = \|r_k^{\text{GMRES}}\|_2$. (See Remark 2.7 and Proposition 2.8 below.)

Proof. For $i = 1, \dots, k$, we define $z_i \equiv x_i^{\text{AA}} - x_0$. To prove the theorem, it suffices to prove the following claims.

Claim 1. For $1 \le j \le k$, if $\{z_1, \dots, z_j\}$ is a basis of $\mathcal{K}_j$, then $\sum_{i=0}^j \alpha_i^{(j)} x_i^{\text{AA}} = x_j^{\text{GMRES}}$ and $x_{j+1}^{\text{AA}} = g(x_j^{\text{GMRES}})$.

Claim 2. For $1 \le j \le k$, $\{z_1, \dots, z_j\}$ is a basis of $\mathcal{K}_j$.

We first prove Claim 1. We have that

$$f_0 = g(x_0) - x_0 = Ax_0 + b - x_0 = b - (I-A)x_0 = r_0^{\text{GMRES}}. \tag{2.1}$$

Additionally, for $i = 1, \dots, k$,

$$\begin{aligned} f_i &= g(x_i^{\text{AA}}) - x_i^{\text{AA}} = g(x_0 + z_i) - (x_0 + z_i) \\ &= g(x_0) - x_0 + (A-I)z_i = r_0^{\text{GMRES}} - (I-A)z_i. \end{aligned} \tag{2.2}$$

[4]Essentially the same result, with a different proof, is given in Ni [31].

[5]We thank an anonymous referee for noting that the conclusion $x_{k+1}^{\text{AA}} = g(x_k^{\text{GMRES}})$ can easily be recast as $x_{k+1}^{\text{AA}} = x_k^{\text{GMRES}} + r_k^{\text{GMRES}}$, which provides another view of the relationship of Anderson acceleration to GMRES. The corresponding conclusions of Corollary 2.10, Theorem 3.2, and Corollary 3.5 can be recast similarly.

It follows from (2.1) and (2.2) that, for $1 \le j \le k$ and any $\alpha = (\alpha_0, \dots, \alpha_j)^T$,

$$F_j \alpha = \sum_{i=0}^{j} \alpha_i f_i = \left(\sum_{i=0}^{j} \alpha_i \right) r_0^{\mathrm{GMRES}} - (I - A) \left(\sum_{i=1}^{j} \alpha_i z_i \right). \tag{2.3}$$

With (2.3), one easily verifies that if $\{z_1, \dots, z_j\}$ is a basis of $\mathcal{K}_j$, then $\alpha^{(j)} = (\alpha_0^{(j)}, \alpha_1^{(j)}, \dots, \alpha_j^{(j)})^T$, with $\alpha_0^{(j)} = 1 - \sum_{i=1}^{j} \alpha_i^{(j)}$, solves the minimization problem

$$\min_{\alpha=(\alpha_0,\dots,\alpha_j)^T} \|F_j \alpha\|_2 \quad \text{s.t.} \quad \sum_{i=0}^{j} \alpha_i = 1 \tag{2.4}$$

if and only if $(\alpha_1^{(j)}, \dots, \alpha_j^{(j)})^T$ solves the GMRES minimization problem

$$\min_{(\alpha_1,\dots,\alpha_j)^T} \|r_0^{\mathrm{GMRES}} - (I - A) \left(\sum_{i=1}^{j} \alpha_i z_i \right) \|_2. \tag{2.5}$$

Consequently, the solution $\alpha^{(j)} = (\alpha_0^{(j)}, \dots, \alpha_j^{(j)})^T$ of (2.4) satisfies

$$\sum_{i=0}^{j} \alpha_i^{(j)} x_i^{\mathrm{AA}} = \alpha_0^{(j)} x_0 + \sum_{i=1}^{j} \alpha_i^{(j)} (x_0 + z_i) = \left(\sum_{i=0}^{j} \alpha_i^{(j)} \right) x_0 + \sum_{i=1}^{j} \alpha_i^{(j)} z_i$$

$$= x_0 + \sum_{i=1}^{j} \alpha_i^{(j)} z_i = x_j^{\mathrm{GMRES}}, \tag{2.6}$$

which yields

$$x_{j+1}^{\mathrm{AA}} = \sum_{i=0}^{j} \alpha_i^{(j)} g(x_i^{\mathrm{AA}}) = g\left(\sum_{i=0}^{j} \alpha_i^{(j)} x_i^{\mathrm{AA}} \right) = g\left(x_j^{\mathrm{GMRES}} \right),$$

and Claim 1 is proved.

We now prove Claim 2. We have that $z_1 = x_1^{\mathrm{AA}} - x_0 = g(x_0) - x_0 = r_0^{\mathrm{GMRES}}$, which is nonzero since $\|r_0^{\mathrm{GMRES}}\|_2 \ge \|r_{k-1}^{\mathrm{GMRES}}\|_2 > 0$. Thus $\{z_1\}$ is a basis of $\mathcal{K}_1$. If $k = 1$, then the proof is complete.

Suppose that $k > 1$ and, as an inductive hypothesis, that $\{z_1, \dots, z_j\}$ is a basis of $\mathcal{K}_j$ for some j such that $1 \le j < k$. With Claim 1 and (2.6), we have that

$$z_{j+1} = x_{j+1}^{\mathrm{AA}} - x_0 = g(x_j^{\mathrm{GMRES}}) - x_0 = A x_j^{\mathrm{GMRES}} + b - x_0$$

$$= b - (I - A) x_j^{\mathrm{GMRES}} + x_j^{\mathrm{GMRES}} - x_0 = r_j^{\mathrm{GMRES}} + \sum_{i=1}^{j} \alpha_i^{(j)} z_i.$$

Since $r_j^{\mathrm{GMRES}} \in \mathcal{K}_{j+1}$ and $\sum_{i=1}^{j} \alpha_i^{(j)} z_i \in \mathcal{K}_j$, we have that $z_{j+1} \in \mathcal{K}_{j+1}$. Moreover, since $\|r_{j-1}^{\mathrm{GMRES}}\|_2 > \|r_j^{\mathrm{GMRES}}\|_2 \ge \|r_{k-1}^{\mathrm{GMRES}}\|_2 > 0$ by assumption, it follows from Lemma 2.4 that $r_j^{\mathrm{GMRES}} \notin \mathcal{K}_j$. Consequently, z_{j+1} cannot depend linearly on $\{z_1, \dots, z_j\}$, and we conclude that $\{z_1, \dots, z_{j+1}\}$ is a basis of $\mathcal{K}_{j+1}$. This completes the induction and the proof of Claim 2. ∎

LEMMA 2.4. *Suppose that GMRES is applied to $Mx = b$ with nonsingular M. If $\|r_{j-1}^{\text{GMRES}}\|_2 > \|r_j^{\text{GMRES}}\|_2 > 0$ for some $j > 0$, then $r_j^{\text{GMRES}} \notin \mathcal{K}_j$.*

Proof. For convenience, we define $\mathcal{K}_0 \equiv \{0\}$. For $\ell \geq 0$, we denote r_ℓ^{GMRES} simply by r_ℓ and denote the orthogonal projection onto $M(\mathcal{K}_\ell)^\perp$ by π_ℓ. Since $M(\mathcal{K}_\ell)^\perp \subseteq M(\mathcal{K}_{\ell-1})^\perp$ for $\ell = 1, \ldots, j$, an easy induction verifies that $r_j = \pi_j r_{j-1}$.

Suppose that $r_j \neq 0$ for some $j > 0$. If $r_j \in \mathcal{K}_j$, then $r_j \in \mathcal{K}_j \cap M(\mathcal{K}_j)^\perp \subseteq \mathcal{K}_j \cap M(\mathcal{K}_{j-1})^\perp$. Since $\mathcal{K}_j \cap M(\mathcal{K}_{j-1})^\perp$ is a one-dimensional subspace containing r_{j-1}, we have that $r_j = \lambda r_{j-1}$ for some scalar λ. Then $r_j = \pi_j r_j = \lambda \pi_j r_{j-1} = \lambda r_j$. Since $r_j \neq 0$, it follows that $\lambda = 1$ and $r_j = r_{j-1}$ and, consequently, that $\|r_{j-1}^{\text{GMRES}}\|_2 = \|r_j^{\text{GMRES}}\|_2$. ☐

Remark 2.5. If the assumptions of the theorem hold and $r_k^{\text{GMRES}} = 0$, then $x_k^{\text{GMRES}} = g(x_k^{\text{GMRES}})$, i.e., x_k^{GMRES} is a fixed point of g. Then the theorem implies that $x_{k+1}^{\text{AA}} = g(x_k^{\text{GMRES}}) = x_k^{\text{GMRES}}$ is also a fixed point of g, and a practical implementation of Algorithm AA would declare success and terminate at step $k+1$. We note that the equivalence of the least-squares problems (2.4) and (2.5) implies that (2.4) has a unique solution and, thus, x_{k+1}^{AA} is uniquely defined in this case, even though F_k is rank-deficient according to the following proposition.

PROPOSITION 2.6. *Suppose that the assumptions of Theorem 2.2 hold. Then* $\operatorname{rank} F_k \geq k$, *and* $\operatorname{rank} F_k = k$ *if and only if* $r_k^{\text{GMRES}} = 0$.

Proof. We use the developments in the proof of Theorem 2.2. From (2.3) with $j = k$, we have that $F_k \alpha = 0$ for $\alpha = (\alpha_0, \ldots, \alpha_k)^T$ if and only if

$$(I - A)\left(\sum_{i=1}^{k} \alpha_i z_i\right) = \left(\sum_{i=0}^{k} \alpha_i\right) r_0^{\text{GMRES}}. \tag{2.7}$$

Since $\{z_1, \ldots, z_k\}$ is linearly independent and $(I - A)$ is nonsingular, it follows from (2.7) that $\sum_{i=0}^{k} \alpha_i = 0$ if and only if $\alpha = 0$. Consequently, in considering nontrivial solutions of $F_k \alpha = 0$, we can assume without loss of generality that $\sum_{i=0}^{k} \alpha_i = 1$.

Suppose that $\alpha = (\alpha_0, \ldots, \alpha_k)^T$ and $\bar{\alpha} = (\bar{\alpha}_0, \ldots, \bar{\alpha}_k)^T$ satisfy $F_k \alpha = F_k \bar{\alpha} = 0$ and $\sum_{i=0}^{k} \alpha_i = \sum_{i=0}^{k} \bar{\alpha}_i = 1$. Then, from (2.7), their difference $\alpha - \bar{\alpha}$ satisfies

$$(I - A)\left(\sum_{i=1}^{k} (\alpha_i - \bar{\alpha}_i) z_i\right) = 0.$$

As above, it follows that $\alpha = \bar{\alpha}$. One concludes that the dimension of the null-space of F_k is at most one; hence, $\operatorname{rank} F_k \geq k$.

We have that $\operatorname{rank} F_k = k$ if and only if there is an $\alpha^{(k)} = (\alpha_0^{(k)}, \ldots, \alpha_k^{(k)})$ such that $F_k \alpha^{(k)} = 0$ and $\sum_{i=0}^{k} \alpha_i^{(k)} = 1$. With (2.7), one verifies that this holds if and only if

$$(I - A)\left(\sum_{i=1}^{k} \alpha_i^{(k)} z_i\right) = r_0^{\text{GMRES}},$$

which (cf. (2.5)) holds if and only if $r_k^{\text{GMRES}} = 0$. ☐

Remark 2.7. If the assumptions of the theorem hold and GMRES stagnates at the kth step with $r_{k-1}^{\text{GMRES}} = r_k^{\text{GMRES}} \neq 0$, then the proposition below shows that Algorithm AA without truncation determines $x_{k+1}^{\text{AA}} = x_k^{\text{AA}}$. It follows that $f_k = f_{k+1}$, i.e., the final two columns of F_{k+1} are the same, and the least-squares problem (1.1) at step $(k+1)$ is rank-deficient.[6] Consequently, one would expect near-stagnation of GMRES at some step to result in ill-conditioning of the least-squares problem at the next step. This points up a potential numerical weakness of Anderson acceleration relative to GMRES, which does not break down upon stagnation before the solution has been found. As a result, conditioning of the least-squares problem is a central consideration in implementing Algorithm AA, as noted further in section 4.

PROPOSITION 2.8. *Suppose that the assumptions of Theorem* 2.2 *hold and that* $r_{k-1}^{\mathrm{GMRES}} = r_k^{\mathrm{GMRES}} \neq 0$. *Then* $x_{k+1}^{\mathrm{AA}} = x_k^{\mathrm{AA}}$.

Proof. It follows from Proposition 2.6 that $\operatorname{rank} F_k = k+1$; consequently, (1.1) (with $m_k = k$) has a unique solution. Then, since $r_{k-1}^{\mathrm{GMRES}} = r_k^{\mathrm{GMRES}}$, the solution must be $\alpha^{(k)} = (\alpha_0^{(k-1)}, \dots, \alpha_{k-1}^{(k-1)}, 0)^T$, which yields

$$x_{k+1}^{\mathrm{AA}} = \sum_{i=0}^{k-1} \alpha_i^{(k-1)} g(x_i^{\mathrm{AA}}) = x_k^{\mathrm{AA}}. \qquad \square$$

For the remainder of this section, we shift notation slightly. We consider solving a linear system $Ax = b$ numerically using a classical stationary iteration based on an operator splitting $A = M - N$, where M and N are in $\mathbb{R}^{n\times n}$ and M is nonsingular. With this splitting, the system $Ax = b$ is recast as the fixed-point equation $x = g(x) \equiv M^{-1}Nx + M^{-1}b$, and the stationary iteration is fixed-point iteration with this g.

The following corollary of Theorem 2.2 asserts that, with this g, Anderson acceleration without truncation is essentially equivalent in the sense of Theorem 2.2 to GMRES applied to the left-preconditioned system $M^{-1}Ax = M^{-1}b$, starting with the same x_0. Our notation is as before, except that the jth GMRES residual is now $r_j^{\mathrm{GMRES}} \equiv M^{-1}b - M^{-1}Ax_j^{\mathrm{GMRES}}$ for each j. Our assumptions are as follows.

Assumption 2.9.

- $A = M - N$, where $A \in \mathbb{R}^{n\times n}$ and $M \in \mathbb{R}^{n\times n}$ are nonsingular.
- $g(x) = M^{-1}Nx + M^{-1}b$ for $b \in \mathbb{R}^n$.
- Anderson acceleration is not truncated, i.e., $m_k = k$ for each k.
- GMRES is applied to the left-preconditioned system $M^{-1}Ax = M^{-1}b$ with initial point $x_0 = x_0^{\mathrm{AA}}$.

COROLLARY 2.10. *Suppose that Assumption* 2.9 *holds and that, for some* $k > 0$, $r_{k-1}^{\mathrm{GMRES}} \neq 0$ *and also* $\|r_{j-1}^{\mathrm{GMRES}}\|_2 > \|r_j^{\mathrm{GMRES}}\|_2$ *for each* j *such that* $0 < j < k$. *Then* $\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{\mathrm{AA}} = x_k^{\mathrm{GMRES}}$ *and* $x_{k+1}^{\mathrm{AA}} = g(x_k^{\mathrm{GMRES}})$.

Proof. Apply Theorem 2.2 with A and b replaced by $M^{-1}N$ and $M^{-1}b$, respectively. $\square$

We hasten to note that we do not recommend applying Anderson acceleration to stationary iterations as a general alternative to preconditioned GMRES because of the potential numerical weakness brought out in Remark 2.7 and Proposition 2.8. However, there may be special circumstances in which this alternative may be advantageous, as noted in section 5.3.1.

[6]In this case, the least-squares problem with F_{k+1} fails to have a unique solution. In principle, one can continue the algorithm by specifying a particular solution in some way; however, it is not hard to verify that every solution leads to $x_{k+2}^{\mathrm{AA}} = x_{k+1}^{\mathrm{AA}} = x_k^{\mathrm{AA}}$. Extending this reasoning, one sees that $x_k^{\mathrm{AA}} = x_{k+1}^{\mathrm{AA}} = x_{k+2}^{\mathrm{AA}} = \dots$, i.e., the algorithm can make no further progress. Thus a practical implementation should terminate if successive iterates are the same (or nearly the same in floating-point arithmetic).

3. Anderson acceleration and multisecant updating.

We begin by recalling the relationship of Anderson acceleration to quasi-Newton updating. This was first shown by Eyert [18]; we follow the clarified derivation given by Fang and Saad [19], with a few very minor differences.[7] Proceeding as in [19], we write the least-squares problem (1.1) in the equivalent form

$$\min_{\gamma=(\gamma_0,\dots,\gamma_{m_k-1})^T} \|f_k - \mathcal{F}_k\gamma\|_2, \tag{3.1}$$

where $\mathcal{F}_k = (\Delta f_{k-m_k}, \dots, \Delta f_{k-1})$ with $\Delta f_i = f_{i+1} - f_i$ for each i, and α and γ are related by $\alpha_0 = \gamma_0$, $\alpha_i = \gamma_i - \gamma_{i-1}$ for $1 \le i \le m_k - 1$, and $\alpha_{m_k} = 1 - \gamma_{m_k-1}$. With the solution of (3.1) denoted by $\gamma^{(k)} = (\gamma_0^{(k)}, \dots, \gamma_{m_k-1}^{(k)})^T$, the next iterate of Algorithm AA is given by

$$\begin{aligned} x_{k+1} &= g(x_k) - \sum_{i=0}^{m_k-1} \gamma_i^{(k)}[g(x_{k-m_k+i+1}) - g(x_{k-m_k+i})] \\ &= x_k + f_k - (\mathcal{X}_k + \mathcal{F}_k)\gamma^{(k)}, \end{aligned}$$

where $\mathcal{X}_k = (\Delta x_{k-m_k}, \dots, \Delta x_{k-1})$ with $\Delta x_i = x_{i+1} - x_i$ for each i. Assuming that $\mathcal{F}_k$ is full-rank and substituting the normal-equation characterization $\gamma^{(k)} = (\mathcal{F}_k^T\mathcal{F}_k)^{-1}\mathcal{F}_k^T f_k$ in this expression, one obtains the quasi-Newton form (cf. Dennis and Schnabel [14])

$$x_{k+1} = x_k - G_k f_k, \tag{3.2}$$

in which

$$G_k \equiv -I + (\mathcal{X}_k + \mathcal{F}_k)(\mathcal{F}_k^T\mathcal{F}_k)^{-1}\mathcal{F}_k^T \tag{3.3}$$

is regarded as an approximate inverse of the Jacobian of $f(x) \equiv g(x) - x$.

It is observed in [19] that G_k satisfies the inverse multisecant condition $G_k\mathcal{F}_k = \mathcal{X}_k$ and, moreover, that $\|G_k + I\|_F$ is minimal among all matrices satisfying this condition.[8] Thus (3.3) can be viewed as the *second Broyden update* [7] of $-I$ satisfying $G_k\mathcal{F}_k = \mathcal{X}_k$.

Fang and Saad [19] define an *Anderson family* of methods of the form (3.2) that includes the method with G_k given by (3.3) as a particular case, designated the *Type* II method in [19]. Another case of particular interest is the *Type* I method, in which

$$G_k \equiv -I + (\mathcal{X}_k + \mathcal{F}_k)(\mathcal{X}_k^T\mathcal{F}_k)^{-1}\mathcal{X}_k^T, \tag{3.4}$$

where we assume for now that $\mathcal{X}_k^T\mathcal{F}_k$ is nonsingular. With the Sherman–Morrison–Woodbury formula [41], [44], this is seen to correspond to the approximate Jacobian

$$J_k \equiv G_k^{-1} = -I + (\mathcal{F}_k + \mathcal{X}_k)(\mathcal{X}_k^T\mathcal{X}_k)^{-1}\mathcal{X}_k^T,$$

[7]The main differences are that we take $\beta_k = 1$ in (1.2), whereas a more general value $\beta_k = \beta$ is allowed in [19], and for each i, we define $f_i = g(x_i) - x_i$, whereas $f_i = x_i - g(x_i)$ in [19].

[8]This minimal-norm property is easily seen by modestly extending the projection calculus developed in Dennis and Schnabel [13] and Dennis and Walker [15]. Indeed, one has that $\mathcal{Q}_k \equiv \{M \in \mathbb{R}^{n\times n} : M\mathcal{F}_k = \mathcal{X}_k\}$ is an affine subspace of $\mathbb{R}^{n\times n}$ with parallel subspace $\mathcal{N}_k \equiv \{M \in \mathbb{R}^{n\times n} : M\mathcal{F}_k = 0\}$ and normal (in $\|\cdot\|_F$) element $\mathcal{X}_k(\mathcal{F}_k^T\mathcal{F}_k)^{-1}\mathcal{F}_k^T$. Writing $G_k = -I\left[I - \mathcal{F}_k(\mathcal{F}_k^T\mathcal{F}_k)^{-1}\mathcal{F}_k^T\right] + \mathcal{X}_k(\mathcal{F}_k^T\mathcal{F}_k)^{-1}\mathcal{F}_k^T$, one easily verifies that the first term on the right is the $\|\cdot\|_F$-orthogonal projection of $-I$ onto $\mathcal{N}_k$, and it follows that G_k is the $\|\cdot\|_F$-orthogonal projection of $-I$ onto $\mathcal{Q}_k$.

which satisfies the direct multisecant condition $J_k \mathcal{X}_k = \mathcal{F}_k$ and is such that $\|J_k + I\|_F$ is minimal among all matrices satisfying this condition. Thus J_k can be regarded as the *first Broyden update* [7] of $-I$ satisfying $J_k \mathcal{X}_k = \mathcal{F}_k$.

For additional perspectives on Anderson acceleration and the Anderson family, we consider the Type I method further in the remainder of this section. It is useful to note that, with G_k given by (3.4), (3.2) becomes

$$x_{k+1} = x_k + f_k - (\mathcal{X}_k + \mathcal{F}_k)(\mathcal{X}_k^T \mathcal{F}_k)^{-1} \mathcal{X}_k^T f_k. \tag{3.5}$$

With $\gamma^{(k)} = (\gamma_0^{(k)}, \ldots, \gamma_{m_k-1}^{(k)})^T$ and $\alpha^{(k)} = (\alpha_0^{(k)}, \ldots, \alpha_{m_k}^{(k)})$ now defined by

$$\gamma^{(k)} = (\mathcal{X}_k^T \mathcal{F}_k)^{-1} \mathcal{X}_k^T f_k, \tag{3.6}$$

$$\alpha_0^{(k)} = \gamma_0^{(k)}, \quad \alpha_i^{(k)} = \gamma_i^{(k)} - \gamma_{i-1}^{(k)} \text{ for } 1 \le i \le m_k - 1, \quad \alpha_{m_k}^{(k)} = 1 - \gamma_{m_k-1}^{(k)}, \tag{3.7}$$

equation (3.5) yields

$$\begin{aligned} x_{k+1} &= g(x_k) - \sum_{i=0}^{m_k-1} \gamma_i^{(k)} [g(x_{k-m_k+i+1}) - g(x_{k-m_k+i})] \\ &= \sum_{i=0}^{m_k} \alpha_i^{(k)} g(x_{k-m_k+i}). \end{aligned} \tag{3.8}$$

Note that $\sum_{i=0}^{m_k} \alpha_i^{(k)} = 1$. It is a straightforward exercise, which we leave to interested readers, to cast the Type I method in a form analogous to that of Algorithm AA.

In the remainder of this section, we assume that g is linear as in section 2. In Theorem 3.2, we show that the Anderson family Type I method without truncation is essentially equivalent in the sense of section 2 to the Arnoldi method applied to the equivalent linear system $(I - A)x = b$ starting with the same x_0. It follows that remarks analogous to those preceding Theorem 2.2 hold to the effect that truncated and restarted versions of the Type I method are similarly essentially equivalent to truncated and restarted versions of the Arnoldi method. Our notational conventions follow those in section 2: For each j, x_j^{Arnoldi} and r_j^{Arnoldi} denote the jth Arnoldi method iterate and residual, respectively; $\mathcal{K}_j$ denotes the jth Krylov subspace generated by $(I-A)$ and r_0^{Arnoldi}; and $x_j^{\text{Type I}}$ denotes the jth iterate of the Type I method. Our major assumptions are as follows.

Assumption 3.1.
- $g(x) = Ax + b$ for $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$.
- The Type I method is not truncated, i.e., $m_k = k$ for each k.
- $(I - A)$ is nonsingular.
- The Arnoldi method is applied to $(I-A)x = b$ with initial point $x_0 = x_0^{\text{Type I}}$.

THEOREM 3.2. *Suppose that Assumption* 3.1 *holds and that, for some* $k > 0$, x_j^{Arnoldi} *is defined for* $0 < j \le k$ *and* $r_{k-1}^{\text{Arnoldi}} \ne 0$. *Then* $\mathcal{X}_j^T \mathcal{F}_j$ *is nonsingular for* $0 < j \le k$, *and, with* $\alpha^{(k)} = (\alpha_0^{(k)}, \ldots, \alpha_{m_k}^{(k)})$ *given by* (3.6)–(3.7), $\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{\text{Type I}} = x_k^{\text{Arnoldi}}$ *and* $x_{k+1}^{\text{Type I}} = g(x_k^{\text{Arnoldi}})$.

Remark 3.3. An iterate x_k^{Arnoldi} of the Arnoldi method is uniquely characterized by the orthogonal-residual condition $r_k^{\text{Arnoldi}} \perp \mathcal{K}_k$, if it is possible to satisfy this condition. However, satisfying this condition may not be possible for some k, in which case x_k^{Arnoldi} is not defined. It has been shown by Brown [6] that x_k^{Arnoldi} is defined if and only if $\|r_{k-1}^{\text{GMRES}}\|_2 > \|r_k^{\text{GMRES}}\|_2$, i.e., GMRES does not stagnate at the kth step.

It follows that the assumption in Theorem 3.2 that x_j^{Arnoldi} is defined for $0 < j \le k$ is equivalent to the assumption that $\|r_{j-1}^{\text{GMRES}}\|_2 > \|r_j^{\text{GMRES}}\|_2$ for $0 < j \le k$. This latter assumption is slightly stronger than the nonstagnation assumption in Theorem 2.2, which does not preclude stagnation at the kth step.

Proof. The proof parallels the proof of Theorem 2.2 but differs significantly in some details. For $i = 1, \dots, k$, we define $z_i \equiv x_i^{\text{Type I}} - x_0$ and make the claims below, from which the theorem follows.

Claim 1. For $1 \le j \le k$, if $\{z_1, \dots, z_j\}$ is a basis of $\mathcal{K}_j$, then $\sum_{i=0}^{j} \alpha_i^{(j)} x_i^{\text{Type I}} = x_j^{\text{Arnoldi}}$ and $x_{j+1}^{\text{Type I}} = g(x_j^{\text{Arnoldi}})$.

Claim 2. For $1 \le j \le k$, $\{z_1, \dots, z_j\}$ is a basis of $\mathcal{K}_j$.

To prove Claim 1, we note that, for $1 \le j \le k$ and any $\gamma = (\gamma_0, \dots, \gamma_{j-1})$,

$$(3.9) \qquad \begin{aligned} f_j - \mathcal{F}_j\gamma &= f_j - \sum_{i=0}^{j-1} \gamma_i (f_{i+1} - f_i) \\ &= \alpha_0 f_0 + \alpha_1 f_1 + \cdots + \alpha_{j-1} f_{j-1} + \alpha_j f_j, \end{aligned}$$

where $\alpha_0 = \gamma_0$, $\alpha_i = \gamma_i - \gamma_{i-1}$ for $1 \le i < j$, and $\alpha_j = 1 - \gamma_{j-1}$. One easily verifies that $f_0 = r_0^{\text{Arnoldi}}$, $\sum_{i=0}^{j} \alpha_i = 1$, and $f_i = f_0 - (I - A) z_i = r_0^{\text{Arnoldi}} - (I - A) z_i$ for $1 \le i \le j$. Then (3.9) yields

$$f_j - \mathcal{F}_j \gamma = r_0^{\text{Arnoldi}} - (I - A) \sum_{i=1}^{j} \alpha_i z_i,$$

which is to say that $f_j - \mathcal{F}_j\gamma$ is the residual associated with $x_0 + \sum_{i=1}^{j} \alpha_i z_i$ resulting from the step $\sum_{i=1}^{j} \alpha_i z_i \in \mathcal{K}_j$. Since

$$\begin{aligned} \mathcal{K}_j &= \text{span}\,\{z_1, z_2, \dots, z_j\} = \text{span}\,\{z_1, z_2 - z_1, \dots, z_j - z_{j-1}\} \\ &= \text{span}\,\{\Delta x_0, \Delta x_1, \dots, \Delta x_{j-1}\}, \end{aligned}$$

the orthogonal-residual condition that uniquely characterizes the Arnoldi step x_j^{Arnoldi}, if it is defined, is

$$(3.10) \qquad \mathcal{X}_j^T f_j - \mathcal{X}_j^T \mathcal{F}_j \gamma = \mathcal{X}_j^T \left[r_0^{\text{Arnoldi}} - (I - A) \sum_{i=1}^{j} \alpha_i z_i \right] = 0.$$

Thus our assumption that x_j^{Arnoldi} is defined implies that there exist a unique γ and corresponding α satisfying (3.10). It follows that $\mathcal{X}_j^T \mathcal{F}_j$ is nonsingular. Moreover, the solution is $\gamma^{(j)} \equiv (\mathcal{X}_j \mathcal{F}_j)^{-1} \mathcal{X}_j^T f_j$, and the corresponding $\alpha^{(j)}$ is such that

$$(3.11) \qquad x_j^{\text{Arnoldi}} = x_0 + \sum_{i=1}^{j} \alpha_i^{(j)} z_i = \sum_{i=0}^{j} \alpha_i^{(j)} x_i^{\text{Type I}},$$

whence $x_{j+1}^{\text{Type I}} = g(x_j^{\text{Arnoldi}})$.

To prove Claim 2, we note that $z_1 = g(x_0) - x_0 = r_0^{\text{Arnoldi}}$, which is nonzero since $r_{k-1}^{\text{Arnoldi}} \ne 0$. Thus $\{z_1\}$ is a basis of $\mathcal{K}_1$, and the proof is complete if $k = 1$.

We suppose that $k > 1$ and make the inductive hypothesis that $\{z_1, \dots, z_j\}$ is a basis of $\mathcal{K}_j$ for some j with $1 \le j < k$. With Claim 1 and (3.11), we have that

$$\begin{aligned} z_{j+1} &= x_{j+1}^{\text{Type I}} - x_0 = g(x_j^{\text{Arnoldi}}) - x_0 = A x_j^{\text{Arnoldi}} + b - x_0 \\ &= b - (I - A) x_j^{\text{Arnoldi}} + x_j^{\text{Arnoldi}} - x_0 = r_j^{\text{Arnoldi}} + \sum_{i=1}^{j} \alpha_i^{(j)} z_i. \end{aligned}$$

We also have that $r_j^{\text{Arnoldi}} \in \mathcal{K}_{j+1} \cap \mathcal{K}_j^{\perp}$ and $r_j^{\text{Arnoldi}} \neq 0$ since $r_{k-1}^{\text{Arnoldi}} \neq 0$. Since $\sum_{i=1}^{j} \alpha_i^{(j)} z_i \in \mathcal{K}_j$, it follows that z_{j+1} cannot depend linearly on $\{z_1, \dots, z_j\}$, and so $\{z_1, \dots, z_{j+1}\}$ is a basis of $\mathcal{K}_{j+1}$. ◻

We conclude this section with a counterpart of Corollary 2.10 in the case in which a stationary iteration based on an operator splitting $A = M - N$ is applied to solve a system $Ax = b$.

Assumption 3.4.

- $A = M - N$, where $A \in \mathbb{R}^{n \times n}$ and $M \in \mathbb{R}^{n \times n}$ are nonsingular.
- $g(x) = M^{-1} N x + M^{-1} b$ for $b \in \mathbb{R}^n$.
- The Type I method is not truncated, i.e., $m_k = k$ for each k.
- The Arnoldi method is applied to the left-preconditioned system $M^{-1} A x = M^{-1} b$ with initial point $x_0 = x_0^{\text{Type I}}$.

COROLLARY 3.5. *Suppose that Assumption* 3.4 *holds and that, for some* $k > 0$, x_j^{Arnoldi} *is defined for* $0 < j \le k$ *and* $r_{k-1}^{\text{Arnoldi}} \neq 0$. *Then, with* $\alpha^{(k)} = (\alpha_0^{(k)}, \dots, \alpha_{m_k}^{(k)})$ *given by* (3.6)–(3.7), $\sum_{i=0}^{k} \alpha_i^{(k)} x_i^{\text{Type I}} = x_k^{\text{Arnoldi}}$ *and* $x_{k+1}^{\text{Type I}} = g(x_k^{\text{Arnoldi}})$.

Proof. Apply Theorem 3.2 with A and b replaced by $M^{-1} N$ and $M^{-1} b$, respectively. ◻

4. Practical considerations.

4. Practical considerations. We focus on the following general issues that arise in implementing Algorithm AA:

- choosing a particular form of the least-squares problem (1.1) among various equivalent forms;
- choosing a solution method for the least-squares problem;
- choosing m and possibly modifying m_k beyond the simple choice $m_k = \min\{m, k\}$.

Other issues, such as choosing stopping criteria, are straightforward to treat using standard approaches or, in some circumstances, better addressed with problem-specific techniques.

We discuss these three general issues in turn, noting the particular choices that we made in the implementation of Anderson acceleration used in the numerical experiments discussed in section 5. A central consideration is the conditioning of the least-squares problem: a particular form of the least-squares problem may affect conditioning; an ill-chosen solution technique (e.g., solving the normal equation) may exacerbate the effects of ill-conditioning; and choosing larger values of m and m_k may worsen conditioning. This consideration is especially important in view of the potential numerical weakness of Anderson acceleration noted in Remark 2.7.

The form of the least-squares problem. Our implementation uses the unconstrained form (3.1) of the least-squares problem, which offers several advantages and, in our experience, no evident disadvantages. This form has been suggested by Kresse and Furthmüller [28] as well by Fang and Saad [19]. It is convenient for storing and updating information from previous iterations and efficiently using it to solve successive least-squares problems over a number of iterations. In experiments reported by Ni and Walker [32], it resulted in slightly better condition numbers than a comparably efficient alternative form proposed in [32] and in much better condition numbers than the form (1.1) when solved directly using a Lagrange-multiplier approach.

The least-squares solution method. Our implementation solves the least-squares problem using QR decomposition, which provides a good balance of accuracy and efficiency for many applications.[9] Since each $\mathcal{F}_k$ in (3.1) is obtained from its predecessor $\mathcal{F}_{k-1}$ by adding a new column on the right and possibly dropping one or more columns on the left (see the discussion of modifying m_k below), the QR decomposition of $\mathcal{F}_k$ can be efficiently obtained from that of $\mathcal{F}_{k-1}$ in $O(m_k n)$ arithmetic operations using standard QR factor-updating techniques (see, e.g., Golub and Van Loan [22]).

We note that Fang and Saad [19] propose solving the least-squares problem using singular-value decomposition and rank-revealing QR decomposition techniques. These are perhaps the most accurate solution methods and are especially useful for treating rank deficiency. However, they are more costly than updated QR decomposition methods and perhaps less necessary in our implementation in view of the strategy outlined below for modifying m_k to maintain acceptable conditioning.

Choosing m and possibly modifying m_k. If m is small, then the secant information used by the method may be too limited to provide desirably fast convergence. However, if m is large, then the least-squares problem may be undesirably badly conditioned, at least without further restrictions on m_k; additionally, outdated secant information from previous iterations may be retained to the point of degrading convergence. Thus the most appropriate choice of m is likely to be application-dependent. In the experiments reported in section 5, effective choices of m ranged from three (with $n = 3$) to 50 (with $n = 16,384$).

For possibly modifying m_k, our implementation follows a strategy used by Yang et al. [45] that is intended to maintain acceptable conditioning of the least-squares problem. In this, the condition number of the least-squares coefficient matrix (which is just the condition number of R in the QR decomposition) is monitored, and left-most columns of the matrix are dropped (and the QR decomposition updated) as necessary to keep the condition number below a prescribed threshold. Dropping columns in this way may also have the benefit of discarding outdated secant information from earlier iterations, albeit at the risk of losing information that may still be of value.

5. Numerical experiments.

In this section, we report on several numerical experiments. These are by no means intended to be exhaustive or definitive for the applications considered. Rather, they are intended only to indicate how Anderson acceleration can be applied and to illustrate its performance in a variety of settings. The first two applications involve only small numbers of unknowns but deal with methods that are often used to extract information from very large data sets. The remaining applications, although of relatively modest scale in the experiments considered here, involve techniques that are used in simulations at the largest scales. All experiments were performed in MATLAB. Since timings in MATLAB may not reflect timings obtained in other computational environments, no timings are reported in these experiments.

[9]Efficiency in solving the least-squares problem may, per se, be a minor issue in some applications, in which the cost of solving the least-squares problem, however inefficiently, is small relative to the cost of evaluating g. This is the case, e.g., in the *self-consistent field* iteration in electronic-structure computations; see [19]. Nevertheless, it may be significant in other cases, and we consider it here.

5.1. The expectation-maximization algorithm for mixture densities. The *expectation-maximization (EM) algorithm*, formalized by Dempster, Laird, and Rubin [12], is widely used in computational statistics for obtaining maximum-likelihood estimates from incomplete data. It is often applied in classification and clustering problems to estimate the unknown parameters in a *mixture density*, i.e., a probability density function used to model a statistical population that consists of a mixture of subpopulations. In this context, the algorithm typically reduces to a simple fixed-point iteration with very appealing properties. From a computational viewpoint, it enjoys very strong global convergence properties, is very easy to implement, involves no derivative information, requires minimal storage, and is highly parallelizable. However, the convergence of the iterates is linear and can be frustratingly slow, especially when the subpopulations are "poorly separated" in a certain sense. See, e.g., Redner and Walker [37] for an extensive treatment of maximum-likelihood estimation of mixture parameters using the EM algorithm.

In this experiment, we considered a mixture density composed of three univariate normal densities. The mixture density is $p(x) = \sum_{i=1}^{3} \alpha_i p_i(x|\mu_i, \sigma_i)$, where

$$p_i(x|\mu_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\,\sigma_i} e^{-(x-\mu_i)^2/(2\sigma_i^2)}, \quad 1 \le i \le 3,$$

and the mixture proportions $\{\alpha_i\}_{i=1}^{3}$ are nonnegative and sum to one. We assumed that the mixture proportions and variances are known and considered estimating the means $\{\mu_i\}_{i=1}^{3}$ from a sample $\{x_k\}_{k=1}^{N}$ of observations on the mixture that are "unlabeled," i.e., their subpopulations of origin are unknown. In this case, an EM iteration on the means is given by

$$\mu_i^+ = \left\{ \sum_{k=1}^{N} x_k \frac{\alpha_i p_i(x_k|\mu_i, \sigma_i)}{p(x_k)} \right\} \bigg/ \left\{ \sum_{k=1}^{N} \frac{\alpha_i p_i(x_k|\mu_i, \sigma_i)}{p(x_k)} \right\}, \quad 1 \le i \le 3,$$

where the current values of $\{\mu_i\}_{i=1}^{3}$ are used with the known proportions and variances to evaluate the expressions on the right-hand side.

We report on a particular trial, in which we observed the performance of the EM algorithm with and without Anderson acceleration as the subpopulations in the mixture became increasingly poorly separated. Specifically, we took $(\alpha_1, \alpha_2, \alpha_3) = (.3, .3, .4)$, $(\sigma_1, \sigma_2, \sigma_3) = (1, 1, 1)$ and generated samples of 100,000 observations on the mixture corresponding to the sets of mean values $(\mu_1, \mu_2, \mu_3) = (0, 2, 4)$, $(0, 1, 2)$, and $(0, .5, 1)$. For these sets of mean values, the component densities are fairly well separated, rather poorly separated, and very poorly separated, respectively. The samples were generated with the MATLAB command `randn` using the same seed in each case. The large sample size was used both to ensure that the samples accurately reflected the underlying mixture densities and to justify obtaining considerable accuracy in the estimates by imposing tight stopping tolerances on the iterations. In this trial, Anderson acceleration was applied with $m = 3$.

The results of this trial are shown in Figure 1. In this figure, the convergence of the EM iterates without Anderson acceleration is shown by the three dashed blue curves, with the bottom curve showing convergence in the best-separated case and the top curve showing convergence in the worst-separated case. Note that the convergence is rather slow in the best-separated case and degrades dramatically as the separation of the component densities worsens. In contrast, the convergence of the accelerated iterates is much faster and is not significantly affected by the worsening separation.

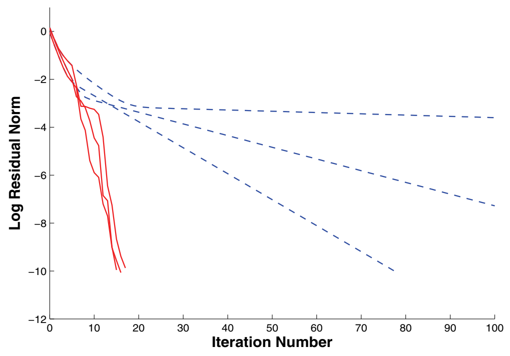

FIG. 1. *Convergence of the EM iterates with (solid red curves) and without (dashed blue curves) Anderson acceleration.*

5.2. Alternating nonnegative least-squares for nonnegative matrix factorization. Nonnegative matrix factorization is an important tool in data mining; see, e.g., Eldén [17] and the references therein. As posed in [17], the problem is as follows: Given $A \in \mathbb{R}^{m\times n}$, find nonnegative matrices $W \in \mathbb{R}^{m\times \ell}$ and $H \in \mathbb{R}^{\ell\times n}$ that solve $\min_{W\geq 0, H\geq 0} \|A - WH\|_F$. A widely used solution technique for this nonlinear least-squares problem is *alternating nonnegative least-squares* (ANNLS), in which one generates sequences of approximate solutions $\{H_k\}$ and $\{W_k\}$, beginning with some $W_0 \geq 0$, by alternately solving the standard nonnegatively constrained linear least-squares problems $H_k = \arg\min_{H\geq 0} \|A - W_k H\|_F$ and $W_{k+1} = \arg\min_{W\geq 0} \|A - WH_k\|_F$. Additionally, a normalization is imposed at each iteration to avoid growth of one factor and decay of another; see [17, sect. 9.2] for details.

Here, we consider ANNLS to be a fixed-point iteration through the assignment $(W_k, H_k) \to (W_{k+1}, H_{k+1})$. In the obvious way, we regard the set of all pairs (W, H) as a vector space with inner-product and norm defined using the Frobenius inner-product and norm on $\mathbb{R}^{m\times\ell}$ and $\mathbb{R}^{\ell\times n}$. In this setting, it is straightforward to apply Anderson acceleration to ANNLS.

In this experiment, we applied ANNLS with and without Anderson acceleration to obtain the nonnegative matrix factorization of the "term-document" matrix A from Example 1.1 of [17], shown on the left in Figure 2. We took $\ell = 3$, so that the W and H factors are 10×3 and 3×5, respectively. We followed the MATLAB procedures given in [17, sect. 9.2] for implementing ANNLS and SVD-based initialization of the iterations (see also Boutsidis and Gallopoulos [2]). We used $m = 3$ in Anderson acceleration. The convergence of the ANNLS iterates with and without Anderson acceleration is shown on the right in Figure 2. In both cases, the iterates converged to factors consistent with those given in [17, p. 110]. We note that, in general, there is a possibility that Anderson acceleration may produce matrices with negative entries; however, that did not occur in this experiment.

5.3. Domain decomposition. Domain decomposition is widely used as a preconditioning technique for solving linear and even nonlinear PDE problems (see, e.g., Knoll and Keyes [27] and Cai and Keyes [8]). It can also be regarded as a fixed-point iteration for obtaining a solution; however, the iterates typically converge too slowly for it to be used in this way (see, e.g., Garbey [20]). In these experiments, we re-

$$A = \begin{pmatrix} 0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1 \\ 1\,0\,1\,0\,0 \\ 1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0 \\ 1\,0\,1\,1\,0 \\ 0\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1 \\ 0\,1\,1\,0\,0 \end{pmatrix}$$

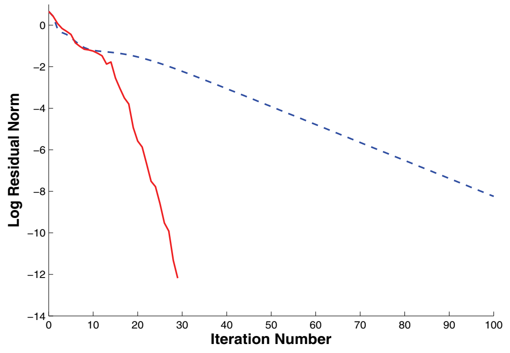

FIG. 2. *Left: the term-document matrix. Right: convergence of the ANNLS iterates with (solid red curve) and without (dashed blue curve) Anderson acceleration.*

port on the effectiveness of Anderson acceleration applied to domain-decomposition iterations for a linear problem and two nonlinear problems. We note that acceleration of domain-decomposition iterations (specifically, additive-Schwarz iterations) has been previously considered by Garbey and collaborators (see [20] and the references therein), who investigated several Aitken-like acceleration procedures and reported encouraging results. To our knowledge, Anderson acceleration has not previously been tried in this setting.

The domain-decomposition technique used in all of our experiments was restricted additive Schwarz (RAS) (Cai and Sarkis [10]) with varying levels of overlap, as noted below. A coarse grid was not used, since scalability studies were not of interest.

5.3.1. A convection-diffusion problem. In this experiment, we observed the performance of RAS with and without Anderson acceleration and also RAS-preconditioned GMRES on a linear convection-diffusion problem, as follows:

$$\begin{aligned} \Delta u + cu + du_x + eu_y &= f \quad \text{in } \mathcal{D} = [0,1] \times [0,1], \\ u &= 0 \quad \text{on } \partial\mathcal{D}. \end{aligned}$$

In the trials reported here, we took $f(x) \equiv -10$ and varied c, d, and e. The problem was discretized using centered differences on a 128×128 grid. RAS was applied on a 4×4 array of subdomains with three grid lines of overlap. All linear subdomain problems were solved using the direct sparse solver in MATLAB. We took $m = 50$ in Anderson acceleration and used 50 as a restart value for GMRES. Since both algorithms terminated in fewer than 50 iterations, this effectively meant that Anderson acceleration was not truncated and GMRES was not restarted.

Representative results are shown in Figure 3. Note that even though Corollary 2.10 does not apply in this case, there is still very close agreement between the iterates produced by RAS with Anderson acceleration and those produced by RAS-preconditioned GMRES. Note also that GMRES requires a product of the "whole-domain" coefficient matrix with a vector at each iteration, but RAS with Anderson acceleration does not. This may be a significant efficiency advantage in circumstances in which these products entail considerable cost. However, as in the remarks at the end of section 2, we caution against applying Anderson acceleration to RAS as a general alternative to RAS-preconditioned GMRES because of the potential numerical weakness of Anderson acceleration brought out in Remark 2.7 and Proposition 2.8.

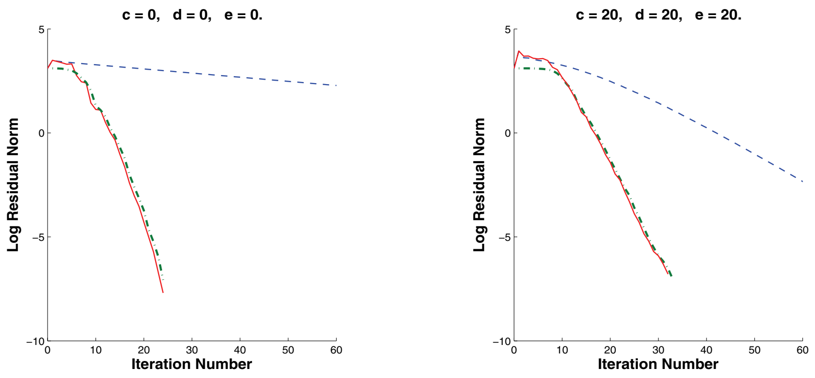

FIG. 3. *The convection-diffusion problem. Convergence of the RAS iterates with (solid red curves) and without (dashed blue curves) Anderson acceleration; convergence of the RAS-preconditioned GMRES iterates (dash-dotted green curve).*

5.3.2. The Bratu problem. The Bratu problem is a nonlinear PDE boundary-value problem, as follows:

$$\begin{aligned} \Delta u + \lambda e^u &= 0 \quad \text{in } \mathcal{D} = [0,1] \times [0,1], \\ u &= 0 \quad \text{on } \partial\mathcal{D}. \end{aligned}$$

This problem is treated in more general form by Fang and Saad [19] and has a long history; see also Glowinski, Keller, and Reinhart [21], Pernice and Walker [34], and the references in those papers. Notwithstanding the exponential nonlinearity, it is not a difficult problem for Newton-like solvers.

In this experiment, we observed the performance of nonlinear RAS with and without Anderson acceleration and also the performance of Newton's method with backtracking applied to the whole-domain problem. As in the previous problem, we used a centered-difference discretization on a 128×128 grid and applied RAS on a 4×4 array of subdomains with three grid lines of overlap. We again took $m = 50$ in Anderson acceleration. We took $\lambda = 6$ in the Bratu problem and used the zero initial approximate solution in all cases. As in the previous experiment, all linear subdomain problems were solved using the direct sparse solver in MATLAB. All nonlinear subdomain problems were solved using Newton's method with a backtracking globalization.

The results are shown in Figure 4. One sees that Anderson acceleration significantly increased the speed of convergence of the RAS iterates. The Newton iterates converged much faster still. However, we note that RAS with Anderson acceleration does note require any action involving the whole-domain Jacobian; additionally, on a parallel machine, the nonlinear subdomain problems could have been solved concurrently at each RAS iteration, which would likely have resulted in significantly shorter time to solution.

5.3.3. A transonic-flow problem. The problem is a one-dimensional compressible-flow problem describing transonic flow in a duct that first narrows and then expands to form a nozzle; see Cai, Keyes, and Young [9] and Young et al. [47] for more details. The PDE problem is

$$\begin{aligned} [A(x)\rho(u)u_x]_x &= 0, \quad 0 < x < 2, \\ u(0) &= 0, \quad u(2) = u_R, \end{aligned}$$

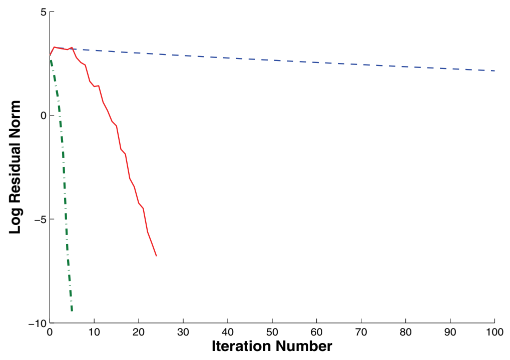

FIG. 4. *The Bratu problem ($\lambda = 6$). Convergence of the nonlinear RAS iterates with (solid red curve) and without (dashed blue curve) Anderson acceleration; convergence of the Newton-backtracking iterates (dash-dotted green curve).*

where $\rho(u) = \left[1 + \frac{\gamma-1}{2}(1-u^2)\right]^{1/(\gamma-1)}$ is the density and $A(x) = 0.4 + 0.6(x-1)^2$ prescribes the cross-section of the duct. The solution u is the potential and u_x is the velocity of the flow. Following [9], [47], we take $u_R = 1.15$ and $\gamma = 1.4$ and apply the finite-difference discretization described in those papers, using the first-order density biasing stabilization outlined in [47]. The resulting discretized problem is very challenging for globalized Newton-like solvers, which tend to require many iterations to resolve the shock.

In this experiment, we observed the performance of nonlinear RAS with and without Anderson acceleration and also the performance of a Newton-GMRES method with a backtracking globalization applied to the whole-domain problem. In this, the Jacobian-vector products needed by GMRES were approximated in "matrix-free" fashion by finite differences of residual values. The nonlinear subdomain problems were also solved with this matrix-free Newton-GMRES-backtracking method. The forcing term $\eta = 10^{-3}$ was used to terminate the GMRES iterations in all cases. In the trial reported here, we used a grid of 512 equally spaced interior grid points. We partitioned the domain into eight subdomains of equal length, with 64 grid points per subdomain, and applied RAS with eight grid lines of overlap. We took $m = 20$ in Anderson acceleration.

The results are shown in Figure 5. We note in particular that RAS with Anderson acceleration required far fewer iterations for convergence than the matrix-free Newton-GMRES-backtracking method. RAS with Anderson acceleration had the additional advantage of not requiring the solution of linear subproblems involving the whole-domain Jacobian.

6. Concluding summary.

Our purpose in the foregoing has been to provide new theoretical insights into Anderson acceleration, to outline useful views on its implementation, and, through experiments involving a range of applications, to illustrate its usefulness as a broadly applicable procedure for accelerating the convergence of fixed-point iterations.

Our theoretical results are given in sections 2 and 3 under the assumption that the fixed-point map is linear. In this case, it is shown in Theorem 2.2 that Anderson acceleration is "essentially equivalent" to GMRES applied to the linear system equivalent to Problem FP in the sense that the iterates produced by one method can

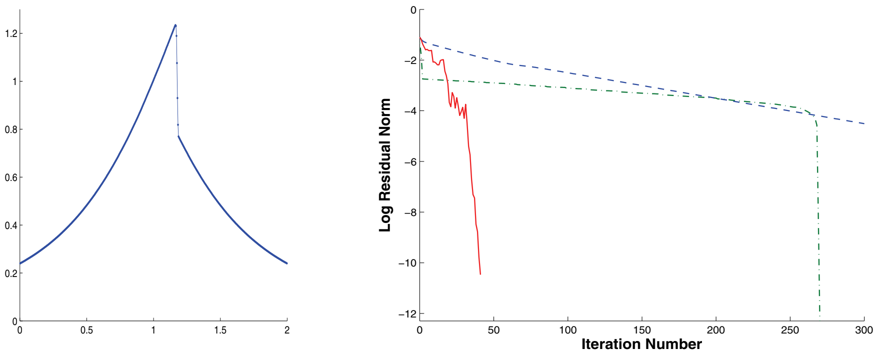

Fig. 5. *The transonic-flow problem. Left: the mach distribution and shock location of the solution on a* 512*-point regular grid. Right: convergence of the RAS iterates with (solid red curve) and without (dashed blue curve) Anderson acceleration; convergence of the "matrix-free" Newton-GMRES-backtracking iterates (dash-dotted green curve).*

be readily obtained from those produced by the other. It follows in Corollary 2.10 that Anderson acceleration applied to a classical stationary iteration defined by an operator splitting is similarly essentially equivalent to GMRES applied to the equivalent left-preconditioned linear system, with left-preconditioning determined by the operator splitting. As noted in Remark 2.7, Anderson acceleration encounters rank deficiency when GMRES stagnates and thus is likely to suffer from ill-conditioning when GMRES convergence is slow. This suggests an inherent potential numerical weakness of the method. Because of this, we caution against applying Anderson acceleration to stationary iterations as a general alternative to preconditioned GMRES; however, as observed in section 5.3.1, this may have advantages in some circumstances. In Theorem 3.2 and Corollary 3.5, we establish results paralleling Theorem 2.2 and Corollary 2.10 that relate the *Type* I method introduced by Fang and Saad [19] to the Arnoldi (full orthogonalization) method.

In section 4, we discuss practical considerations in implementing Anderson acceleration, focusing on choices that we made in the implementation used in the numerical experiments discussed in section 5. Our preferred form of the least-squares problem is (3.1), which is convenient for storing and efficiently updating information from previous iterations. For solving this least-squares problem, our implementation uses QR decomposition, which offers a good balance of accuracy and efficiency for many applications. Efficiency is achieved by updating the QR factors from one iteration to the next at the cost of $O(m_k n)$ arithmetic operations. Acceptable accuracy is maintained by dropping left-most columns of the coefficient matrix (and subsequently updating the QR factors) as necessary to keep the condition number below a desired level.

In section 5, we report on experiments in which Anderson acceleration was applied to fixed-point iterations arising in various applications. The first two experiments involve only small numbers of unknowns, but the methods of interest are often used to analyze very large data sets. The remaining experiments, although of relatively modest scales in the trials reported here, involve domain-decomposition methods that are used in simulations at the largest scales. In all cases, Anderson acceleration significantly accelerated the convergence of the underlying fixed-point iterations. In some cases, its performance was noteworthy in other respects as well. We mention in particular that, in the experiment involving the EM algorithm (section 5.1), the convergence of the accelerated iterates was not significantly affected by worsening separation of the mixture subpopulations, while that of the EM iterates without acceleration was severely degraded, and also that, in the transonic-flow experiment (section 5.3.3), nonlinear RAS with Anderson acceleration required far fewer iterations for convergence than the Newton-GMRES-backtracking method. Additionally, in all of the domain-decomposition experiments, RAS with Anderson acceleration required no operations involving a "whole-domain" coefficient matrix or Jacobian; this may be a significant advantage in cases in which such operations entail considerable expense.

The encouraging experimental results in section 5 and also the established record of success of Anderson acceleration in electronic-structure applications suggest that the method is worthy of consideration in many other applications in which it may be similarly successful. However, in addition to the caution noted above, it should be kept in mind that there are no general guarantees of global or even local convergence of which we are aware. In view of these unresolved issues as well as its evident promise, we feel that the method merits much further study.

Acknowledgments. The authors express appreciation to Juan Meza, Chao Yang, and Yousef Saad for helpful discussions relating to this work, and to the anonymous referees, whose suggestions resulted in a number of improvements to the paper.

REFERENCES

[1] D. G. Anderson, *Iterative procedures for nonlinear integral equations*, J. Assoc. Comput. Mach., 12 (1965), pp. 547–560.

[2] C. Boutsidis and E. Gallopoulos, *SVD based initialization: A head start for nonnegative matrix factorization*, Pattern Recognition, 41 (2008), pp. 1350–1362.

[3] C. Brezinski, *Convergence acceleration during the* 20*th century*, J. Comput. Appl. Math., 122 (2000), pp. 1–21.

[4] C. Brezinski and M. Redivo-Zaglia, *Extrapolation Methods Theory and Practice*, North–Holland, Amsterdam, 1991.

[5] C. Le Bris, *Computational chemistry from the perspective of numerical analysis*, Acta Numer., 14 (2005), pp. 363–444.

[6] P. N. Brown, *A theoretical comparison of the Arnoldi and* GMRES *algorithms*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 58–78.

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

[8] X.-C. Cai and D. E. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 24 (2002), pp. 183–200.

[9] X.-C. Cai, D. E. Keyes, and D. P. Young, *A nonlinear additive Schwarz preconditioned inexact Newton method for shocked duct flows*, in Domain Decomposition Methods in Science and Engineering (Lyon, France, 2000), N. Débit, M. Garbey, R. Hoppe, J. Périaux, D. Keyes, and Y. Kuznetsov, eds., CIMNE, Barcelona, 2000, pp. 345–352.

[10] X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 792–797.

[11] N. N. Carlson and K. Miller, *Design and application of a gradient-weighted moving finite element code* I: *In one dimension*, SIAM J. Sci. Comput., 19 (1998), pp. 728–765.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum-likelihood from incomplete data via the EM algorithm*, J. Royal Statist. Soc. Ser. B, 39 (1977), pp. 1–38.

[13] J. E. Dennis, Jr., and R. B. Schnabel, *Least change secant updates for quasi-Newton methods*, SIAM Rev., 21 (1979), pp. 443–459.

[14] J. E. Dennis, Jr., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Classics Appl. Math. 16, SIAM, Philadelphia, 1996.

[15] J. E. Dennis, Jr., and H. F. Walker, *Convergence theorems for least-change secant update methods*, SIAM J. Numer. Anal., 18 (1981), pp. 949–987.

[16] T. Eirola and O. Nevanlinna, *Accelerating with rank-one updates*, Linear Algebra Appl., 121 (1989), pp. 511–520.
[17] L. Eldén, *Matrix Methods in Data Mining and Pattern Recognition*, Fundam. Algorithms 4, SIAM, Philadelphia, 2007.
[18] V. Eyert, *A comparative study on methods for convergence acceleration of iterative vector sequences*, J. Comput. Phys., 124 (1996), pp. 271–285.
[19] H. Fang and Y. Saad, *Two classes of multisecant methods for nonlinear acceleration*, Numer. Linear Algebra Appl., 16 (2009), pp. 197–221.
[20] M. Garbey, *Acceleration of the Schwarz method for elliptic problems*, SIAM J. Sci. Comput., 26 (2005), pp. 1871–1893.
[21] R. Glowinski, H. B. Keller, and L. Reinhart, *Continuation-conjugate gradient methods for the least squares solution of nonlinear boundary value problems*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 793–832.
[22] G. H. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[23] K. Jbilou and H. Sadok, *Some results about vector extrapolation methods and related fixed-point iterations*, J. Comput. Appl. Math., 36 (1991), pp. 385–398.
[24] K. Jbilou and H. Sadok, *Analysis of some vector extrapolation methods for solving systems of linear equations*, Numer. Math., 70 (1995), pp. 73–89.
[25] K. Jbilou and H. Sadok, *Vector extrapolation methods. Applications and numerical comparison*, J. Comput. Appl. Math., 122 (2000), pp. 149–165.
[26] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers Appl. Math. 16, SIAM, Philadelphia, 1995.
[27] D. A. Knoll and D. E. Keyes, *Jacobian-free Newton–Krylov methods: A survey of approaches and applications*, J. Comput. Phys., 193 (2004), pp. 357–397.
[28] G. Kresse and J. Furthmüller, *Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set*, Comput. Mater. Sci., 6 (1996), pp. 15–50.
[29] G. Kresse and J. Furthmüller, *Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set*, Phys. Rev. B, 54 (1996), pp. 169–186.
[30] K. Miller, *Nonlinear Krylov and moving nodes in the method of lines*, J. Comput. Appl. Math., 183 (2005), pp. 275–287.
[31] P. Ni, *Anderson Acceleration of Fixed-Point Iteration with Applications to Electronic Structure Computations*, Ph.D. thesis, Worcester Polytechnic Institute, Worcester, MA, 2009.
[32] P. Ni and H. F. Walker, *A Linearly Constrained Least-Squares Problem in Electronic Structure Computations*, Tech. Report MS-1-13-46, Mathematical Sciences Department, Worcester Polytechnic Institute, Worcester, MA, 2010.
[33] C. W. Oosterlee and T. Washio, *Krylov subspace acceleration of nonlinear multigrid with application to recirculating flows*, SIAM J. Sci. Comput., 21 (2000), pp. 1670–1690.
[34] M. Pernice and H. F. Walker, NITSOL*: A Newton iterative solver for nonlinear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 302–318.
[35] P. Pulay, *Convergence acceleration of iterative sequences. The case of SCF iteration*, Chem. Phys. Lett., 73 (1980), pp. 393–398.
[36] P. Pulay, *Improved SCF convergence*, J. Comput. Chem., 3 (1982), pp. 556–560.
[37] R. A. Redner and H. F. Walker, *Mixture densities, maximum likelihood and the EM algorithm*, SIAM Rev., 26 (1984), pp. 195–239.
[38] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp., 37 (1981), pp. 105–126.
[39] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[40] Y. Saad and M. H. Schultz, GMRES*: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[41] J. Sherman and W. J. Morrison, *Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix*, Ann. Math. Statistics, 20 (1949), p. 621.
[42] D. A. Smith, W. F. Ford, and A. Sidi, *Extrapolation methods for vector sequences*, SIAM Rev., 29 (1987), pp. 199–233.
[43] T. Washio and C. W. Oosterlee, *Krylov subspace acceleration for nonlinear multigrid schemes*, Electron. Trans. Numer. Anal., 6 (1997), pp. 271–290.
[44] M. A. Woodbury, *Inverting Modified Matrices*, Tech. Report Memorandum Report 42, Statistical Research Group, Princeton University, Princeton, NJ, 1950.
[45] C. Yang, J. C. Meza, B. Lee, and L.-W. Wang, *KSSOLV—a MATLAB toolbox for solving the Kohn–Sham equations*, ACM Trans. Math. Software, 36 (2009), pp. 1–35.
[46] U. M. Yang, *A Family of Preconditioned Iterative Solvers for Sparse Linear Systems*, Ph.D.

thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 1995.

[47] D. P. Young, W. P. Huffman, R. G. Melvin, C. L. Hilmes, and F. T. Johnson, *Nonlinear elimination in aerodynamic analysis and design optimization*, in Large-Scale PDE-Constrained Optimization, L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, eds., Springer-Verlag, Berlin, Heidelberg, New York, 2003, pp. 17–43.