SIAM J. SCI. COMPUT.
Vol. 41, No. 1, pp. A365–A379

CONVERGENCE OF THE EDIIS ALGORITHM FOR NONLINEAR EQUATIONS*

XIAOJUN CHEN† AND C. T. KELLEY‡

Abstract. The Energy Direct Inversion on the Iterative Subspace (EDIIS) algorithm was designed to globalize Anderson acceleration, a method for improving the performance of fixed point iteration. The motivating application is electronic structure computations. In this paper we prove a convergence result for that algorithm and illustrate the theory with a computational example.

Key words. nonlinear equations, Anderson acceleration, EDIIS

AMS subject classifications. 65H10, 81V55

DOI. 10.1137/18M1171084

1. Introduction.

The purpose of this paper is to analyze the convergence of the Energy Direct Inversion on the Iterative Subspace (EDIIS) algorithm [21]. EDIIS is a modification of Anderson acceleration [1] or the Direct Inversion on the Iterative Subspace (DIIS) method [34, 37, 22, 21]. EDIIS relaxes the need for a sufficiently accurate initial iterate. EDIIS is the default solver for the self-consistent field (SCF) iteration in the widely used Gaussian [12] quantum chemistry code. We prove convergence from any starting point in a convex set in which the fixed point map is a contraction and then analyze local convergence. Our local convergence is an improvement of the result in [41] and applies to both EDIIS and Anderson acceleration.

We will begin this introductory section with a review of Anderson acceleration and some of the recent convergence results. We will then describe the EDIIS algorithm. In section 2 we prove our convergence results. Finally, in section 3 we report on a computation which both illustrates the theory and, as is also done in [21], shows how the convergence speeds for EDIIS and Anderson acceleration, while identical in theory, can differ significantly in practice.

Our notational convention is that vectors and vector-valued functions in R^N are in bold. Scalars and elements of infinite dimensional spaces (e.g., integral operators and the functions acted upon by those operators) are in the usual italic math font.

Anderson acceleration [1] is an iterative method for fixed point problems of the form

$$\mathbf{u} = \mathbf{G}(\mathbf{u}), \tag{1.1}$$

*Submitted to the journal's Methods and Algorithms for Scientific Computing section February 15, 2018; accepted for publication (in revised form) November 26, 2018; published electronically January 17, 2019.

http://www.siam.org/journals/sisc/41-1/M117108.html

Funding: The work of the first author was partially supported by Hong Kong Research Grants Council grant PolyU153000/15p. The work of the second author was partially supported by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov) and Simulation of Nuclear Reactors under U.S. Department of Energy Contract DEAC05-00OR22725, Army Research Office grant W911NF-16-1-0504, and National Science Foundation grants DMS-1406349 and OAC-1740309. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

†Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China (xiaojun.chen@polyu.edu.hk).

‡Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205 (tim_kelley@ncsu.edu).

where $\mathbf{u} \in R^N$ and $\mathbf{G} : R^N \to R^N$. The method was designed to accelerate Picard or fixed point iteration, i.e.,

$$\mathbf{u}_{k+1} = \mathbf{G}(\mathbf{u}_k). \tag{1.2}$$

Anderson acceleration was originally designed for integral equations and has been widely used in electronic structure computations (see [9] and many references since then) and is now very common in that field. Anderson acceleration is essentially the same as Pulay mixing [33, 32], DIIS [34, 37, 22, 21], and nonlinear GMRES [25, 30, 45, 4]. Other applications include nuclear reactor design [42, 16], stiff dynamics [13], hydrology [24], and fluid-structure interaction [10, 15, 23], where the method is called interface quasi-Newton.

The analysis of Anderson acceleration is far from complete. In this paper we assume, as do all theoretical results about this algorithm, that the map $\mathbf{G}$ is a contraction. In practice, however, Anderson acceleration does very well for problems in which $\mathbf{G}$ is either definitely not a contraction [41] or not provably a contraction. The results here do not explain those cases.

Anderson acceleration was designed for a problem where Newton's method is not practical because obtaining approximate Jacobians or Jacobian-vector products is too costly. One should expect that Newton's method would perform better when derivative information can be had at a reasonable cost, and we have certainly found that to be the case in our own recent work [16]. Anderson iteration maintains a history of residuals

$$\mathbf{F}(\mathbf{u}) = \mathbf{G}(\mathbf{u}) - \mathbf{u}$$

of size at most $m + 1$, where the *depth* m is an algorithmic parameter. When m is important, we will call the iteration Anderson(m). Anderson(0) is Picard iteration by definition.

The formal description in Algorithm 1.1 is most convenient for analysis and exposition, but not for implementation. We refer the reader to [44, 41, 7, 38, 43, 39] for examples of efficient implementations.

Algorithm 1.1 Anderson Acceleration

`anderson`$(\mathbf{u}_0, \mathbf{G}, m)$
  $\mathbf{u}_1 = \mathbf{G}(\mathbf{u}_0)$; $\mathbf{F}_0 = \mathbf{G}(\mathbf{u}_0) - \mathbf{u}_0$
  for $k = 1, \ldots$ **do**
    Choose $m_k \leq \min(m, k)$
    $\mathbf{F}_k = \mathbf{G}(\mathbf{u}_k) - \mathbf{u}_k$
    Minimize $\| \sum_{j=0}^{m_k} \alpha_j^k \mathbf{F}_{k-m_k+j} \|$ subject to $\sum_{j=0}^{m_k} \alpha_j^k = 1$
    $\mathbf{u}_{k+1} = \sum_{j=0}^{m_k} \alpha_j^k \mathbf{G}(\mathbf{u}_{k-m_k+j})$
  end for

The iteration uses the most recent $m+1$ residuals $\mathbf{F}(\mathbf{u}_j)$ for $k - m_k \leq j \leq k$ where $m_k \leq \min(k, m)$. The key step in the iteration is solving the *optimization problem*

$$\text{Minimize} \left\| \sum_{j=0}^{m_k} \alpha_j^k \mathbf{F}(\mathbf{u}_{k-m_k+j}) \right\| \text{ subject to } \sum_{j=0}^{m_k} \alpha_j^k = 1 \tag{1.3}$$

for the coefficients $\{\alpha_j^k\}$.

Any vector norm can be used in the optimization problem with no change in the convergence theory [41]. In particular, the optimization problem for the coefficients

in either the ℓ^1 or the ℓ^∞ norm can be formulated as a linear programming problem [8]. The optimization problem is easier to solve if one uses the ℓ^2 norm, and that is standard practice. In this case optimization problem for the coefficients can be expressed as a linear least squares problem and solved very inexpensively. One way to do this is to solve the linear least squares problem

$$\text{Minimize} \left\| \mathbf{F}(\mathbf{u}_k) + \sum_{j=0}^{m_k-1} \alpha_j^k (\mathbf{F}(\mathbf{u}_{k-m_k+j}) - \mathbf{F}(\mathbf{u}_k)) \right\|_2^2 \tag{1.4}$$

for $\{\alpha_j^k\}_{j=0}^{m_k-1}$. Then one recovers $\alpha_{m_k}^k$ by

$$\alpha_{m_k}^k = 1 - \sum_{j=0}^{m_k-1} \alpha_j^k.$$

The choice of m_k is, in the original form, simply $\min(m, k)$. One can adapt m_k as the iteration progresses to, for example, enforce well-conditioning of the linear least squares problem (1.4) [44, 39].

One can also [11, 35, 34, 44, 31] show that Anderson acceleration is related to multisecant quasi-Newton methods or, in the case of linear problems, GMRES. None of these results lead to a convergence proof, even in the linear case, unless the available storage is large enough to allow GMRES to take a number of iterations equal to the dimension of the problem. The recent work of one of the authors and his students [41, 40, 39] contains the first convergence theory for Anderson acceleration as it is applied in practice.

1.1. Convergence theory. Theorem 1.1 is one of the convergence results from [41]. That paper also has results for several special cases. We assume that $\mathbf{G}$ is a contraction with contractivity constant $c \in (0, 1)$ in a closed set $D \subset R^N$,

$$\|\mathbf{G}(\mathbf{u}) - \mathbf{G}(\mathbf{v})\| \leq c\|\mathbf{u} - \mathbf{v}\| \tag{1.5}$$

for all $\mathbf{u}, \mathbf{v} \in D$. The contraction mapping theorem implies that $\mathbf{G}$ has a unique fixed point $\mathbf{u}^* \in D$. As is standard, we let $\mathbf{e} = \mathbf{u} - \mathbf{u}^*$ and make the assumption from [41] on the smoothness of $\mathbf{G}$ and the Anderson iteration coefficients.

The convergence of the Picard iteration for a contraction map is q-linear [19] with q-factor c, i.e.,

$$\|\mathbf{e}_k\| \leq c\|\mathbf{e}_{k-1}\|.$$

We will show in this paper that Anderson acceleration is r-linear with r-factor c, which means

$$\|\mathbf{e}_k\| = O(c^k).$$

ASSUMPTION 1.1. $\mathbf{G}$ *is Lipschitz continuously differentiable in the ball*

$$\mathcal{B}(\hat{r}) = \{\mathbf{u} \,|\, \|\mathbf{e}\| \leq \hat{r}\} \subset D$$

for some $\hat{r} > 0$.

There is M_α *such that for all* $k \geq 0$

$$\sum_{j=0}^{m_k} |\alpha_j^k| \leq M_\alpha.$$

The differentiability assumption is needed in the analysis, but not in the formulation or implementation of the algorithm. Our convergence result in section 2.2 relaxes the assumption to continuous differentiability.

THEOREM 1.1 ([41]). *Let Assumption* 1.1 *hold, and let* $c < \hat{r} < 1$. *Then if* $\mathbf{u}_0$ *is sufficiently close to* $\mathbf{u}^*$, *the Anderson iteration converges to* $\mathbf{u}^*$. *In fact, for all* $k \geq 0$,

$$\|\mathbf{F}(\mathbf{u}_k)\| \leq \hat{r}^k \|\mathbf{F}(\mathbf{u}_0)\| \text{ and } \|\mathbf{e}_k\| \leq \left(\frac{1+c}{1-c}\right) \hat{r}^k \|\mathbf{e}_0\|. \tag{1.6}$$

The interpretation of this result is that if the initial data are sufficiently good, then the r-factor for Anderson iteration is no worse than the q-factor of Picard iteration as predicted by the contractivity constant c. While r-linear convergence is weaker than q-linear convergence, Anderson acceleration is often faster than Picard iteration in practice. The requirement that the initial iterate be near the solution is also meaningful in practice [36, 46, 47] and motivated the EDIIS algorithm [21], which is the subject of this paper.

Both Picard iteration and Anderson acceleration can perform better than the prediction (see section 3). In practice, Anderson acceleration is often significantly better than Picard iteration, but there is no theory that explains this under practical (i.e., very limited storage) operating conditions.

1.2. The EDIIS algorithm. Anderson acceleration performs poorly for some applications. One example is electronic structure computations for metallic systems where the HOMO-LUMO gap is small and a good initial iterate is difficult to obtain. In this case both Picard iteration and Anderson acceleration perform poorly [21]. In such cases one can sometimes use a small mixing parameter to ensure convergence, especially when the initial iterate is poor. However, a small mixing parameter may degrade the performance of the iteration—especially when near the solution. The role of the damping parameter β in Picard iteration is simple damping:

$$\mathbf{u}_{k+1} = (1-\beta)\mathbf{u}_k + \beta \mathbf{G}(\mathbf{u}_k).$$

If one applies EDIIS or Anderson acceleration to

$$\mathbf{G}_\beta(\mathbf{u}) = (1-\beta)\mathbf{u} + \beta \mathbf{G}(\mathbf{u}),$$

then [40] one obtains

$$\mathbf{u}_{k+1} = (1-\beta)\sum_{j=0}^{m_k} \alpha_j^k \mathbf{u}_{k-m_k+j} + \beta \sum_{j=0}^{m_k} \alpha_j^k \mathbf{G}(\mathbf{u}_{k-m_k+j}),$$

which is how damping is done in Anderson acceleration [1].

One attempt to solve these problems for small systems is the EDIIS algorithm from [21]. In [21] the authors also formulated the fixed point problem to directly minimize energy (hence the name of the method), but that does not affect the convergence analysis in this paper.

EDIIS differs from Anderson acceleration by imposing a nonnegativity constraint on the coefficients. So, the optimization problem becomes

$$\text{Minimize } \left\| \sum_{j=0}^{m_k} \alpha_j^k \mathbf{F}_{k-m_k+j} \right\| \text{ subject to } \sum_{j=0}^{m_k} \alpha_j^k = 1, \alpha_j^k \geq 0. \tag{1.7}$$

In [21] the authors present an example where EDIIS does well and both Picard and Anderson acceleration fail and another example where Anderson acceleration is successful and EDIIS, while converging, does not perform as well. We present another such example in section 3. One reason EDIIS might perform worse than Anderson acceleration could be that the optimization problem (1.7) for EDIIS has a more restricted feasible set and therefore a larger optimal value.

2. Convergence results. Our global convergence is Theorem 2.1. The proof does not require differentiability, but the convergence speed estimate is very pessimistic with an r-factor of $c^{1/(m+1)}$. We follow the global theorem with a local theorem that shows how the convergence behavior becomes locally r-linear with r-factor c, improving on the local results in [41].

2.1. Global convergence.

THEOREM 2.1. *Let* $\mathbf{G}$ *be a contraction on a convex set* $D \subset R^N$ *with contractivity constant* c. *Let* $\mathbf{u}^*$ *be the unique fixed point of* $\mathbf{G}$ *in* D. *Then for any* $\mathbf{u}_0 \in D$, *EDIIS*(m) *converges to* $\mathbf{u}^*$ *r-linearly with r-factor*

$$\hat{c} = c^{1/(m+1)}.$$

In fact,

$$\|\mathbf{e}_k\| \le \hat{c}^k \|\mathbf{e}_0\|. \tag{2.1}$$

Proof. The proof does not use the optimality properties of the coefficients and only requires that the iteration $\{\mathbf{u}_k\}$ have the form

$$\mathbf{u}_{k+1} = \sum_{j=0}^{m_k} \alpha_j^k \mathbf{G}(\mathbf{u}_{k-m_k+j}), \tag{2.2}$$

where $m_k \le m$, $\alpha_j^k \ge 0$, and $\sum_{j=0}^{m_k} \alpha_j^k = 1$.

We induct on k. Clearly (2.1) holds for both $m_k = 0$, by definition, and $k = 1, m_k = 0$ because the iteration in that case is a single Picard iteration (i.e., one step of Anderson(0)). Assume that the result holds for $k \le K$. Then (2.2) and $\sum_{j=0}^{m_K} \alpha_j^K = 1$ imply that

$$\mathbf{e}_{K+1} = \sum_{j=0}^{m_K} \alpha_j^K (\mathbf{G}(\mathbf{u}_{K-m_K+j}) - \mathbf{u}^*).$$

Note that since $\alpha_j^K \ge 0$, $\sum_{j=0}^{m_K} \alpha_j^K = 1$, $\hat{c} < 1$, and $m_K \le m$, we have

$$\sum_{j=0}^{m_K} \alpha_j^K \hat{c}^{K-m_K+j} \le \hat{c}^{K-m}.$$

Hence

$$\begin{aligned}
\|\mathbf{e}_{K+1}\| &\le \textstyle\sum_{j=0}^{m_K} \alpha_j^K \|\mathbf{G}(\mathbf{u}_{K-m_K+j}) - \mathbf{u}^*\| \\
&\le \textstyle\sum_{j=0}^{m_K} \alpha_j^K c \|\mathbf{u}_{K-m_K+j} - \mathbf{u}^*\| \\
&\le c \textstyle\sum_{j=0}^{m_K} \alpha_j^K \hat{c}^{K-m_K+j} \|\mathbf{e}_0\| \le c\hat{c}^{K-m}\|\mathbf{e}_0\| \le \hat{c}^{K+1}(c\hat{c}^{-m-1})\|\mathbf{e}_0\| = \hat{c}^{K+1}\|\mathbf{e}_0\|.
\end{aligned}$$

∎

Theorem 2.1 implies that for any $\delta > 0$ there is K such that all iterations $\{\mathbf{u}_k\}_{k \geq K}$ are in the set

$$\mathcal{B}(\delta) = \{\mathbf{u} \,|\, \|\mathbf{u} - \mathbf{u}^*\| \leq \delta\}.$$

Hence, starting an Anderson acceleration iteration after sufficiently many EDIIS iterations will result in local convergence at the rate predicted by Theorem 1.1, which is better than (2.1) since $\hat{r}$ can be arbitrarily near c and does not depend on m. However, it is not clear how to decide when to restart. The main result in section 2.2, Theorem 2.2, applies to both EDIIS and Anderson acceleration, generalizes the local convergence result from [41, Theorem 1.1], and says that one can simply continue with the EDIIS iteration and the local convergence estimate for Anderson acceleration will hold.

2.2. Local convergence. Theorem 2.2 is the local convergence result. The theorem generalizes the result in [41] by both weakening the assumptions and improving the r-factor.

We will assume that an iteration begins with a history that lies in $\mathcal{B}(\delta)$ for δ sufficiently small. This history could be either from the EDIIS iteration or from the Anderson acceleration iteration itself. Hence the assumption covers not only EDIIS but also allows us to improve the convergence theory from [41]. We will show that the residuals converge r-linearly to zero with an r-factor of c. Formally our assumption is as follows.

ASSUMPTION 2.1. *$\mathbf{G}$ is a continuously differentiable contraction on $D \subset R^N$ with contractivity constant c and $\mathbf{u}^*$ is the unique fixed point of $\mathbf{G}$ in D.*

The iteration begins with $\{\mathbf{u}_l\}_{l=0}^m \subset \mathcal{B}(\delta) \subset D$. There are real $\{\alpha_j^k\}_{j=0}^{m_k}$ with $0 \leq m_k \leq \min(m, k)$ such that

$$\sum_{j=0}^{m_k} \alpha_j^k = 1,$$

$$\mathbf{u}_{k+1} = \sum_{j=0}^{m_k} \alpha_j^k \mathbf{G}(\mathbf{u}_{k-m_k+j}), \tag{2.3}$$

and

$$\left\| \sum_{j=0}^{m_k} \alpha_j^k \mathbf{F}(\mathbf{u}_{k-m_k+j}) \right\| \leq \|\mathbf{F}(\mathbf{u}_k)\|. \tag{2.4}$$

Finally, there is $\hat{c} \in (c, 1)$ so that

$$\|\mathbf{F}(\mathbf{u}_l)\| \leq \hat{c}^l \|\mathbf{F}(\mathbf{u}_0)\| \quad \textit{for } 0 \leq l \leq m. \tag{2.5}$$

Theorem 2.1 implies that Assumption 2.1 will hold after sufficiently many EDIIS iterations. In the theorem there is no history if $m = 0$ and in that case the iteration is a Picard iteration. While we are motivated by a local iteration from the EDIIS algorithm, the local theory does not require that the coefficients be nonnegative.

Assumption 2.1 weakens the ones in [41] in two ways. The first is that we no longer assume that $\mathbf{G}$ is Lipschitz continuously differentiable. The second is that we do not assume that the coefficients $\{\alpha_j^k\}$ come from any particular optimization problem—only that the linear combination of residuals has norm no larger than that of the most recent residual.

The idea of the analysis is that as the iteration converges, the upper bound for the r-factor will approach c and therefore the r-factor is no larger than c. In the case where there is no history, this fact was implicit in the results from [41]. Adding the history makes the bookkeeping more difficult, and the proof of Theorem 2.2 must account for that.

THEOREM 2.2. *Let Assumption* 2.1 *hold. Assume that there is* M_α *such that*

$$\sum_{j=0}^{m_k} |\alpha_j^k| \le M_\alpha \tag{2.6}$$

for all $k \ge 0$. *Then if* δ *is sufficiently small, the iteration given by* (2.3) *and* (2.4) *converges to* $\mathbf{u}^*$ *and*

$$\limsup_{k\to\infty} \left(\frac{\|\mathbf{F}(\mathbf{u}_k)\|}{\|\mathbf{F}(\mathbf{u}_0)\|} \right)^{1/k} \le c. \tag{2.7}$$

Proof. Let $0 < \epsilon < \hat{c} - c$. We will show that for $\|\mathbf{e}_0\|$ sufficiently small,

$$\limsup_{k\to\infty} \left(\frac{\|\mathbf{F}(\mathbf{u}_k)\|}{\|\mathbf{F}(\mathbf{u}_0)\|} \right)^{1/k} \le c + \epsilon. \tag{2.8}$$

This will complete the proof since ϵ is arbitrary and we can restart the proof once we have m vectors in the history which are near enough to $\mathbf{u}^*$ to reduce ϵ further.

We induct on k. Define $L = (c/\hat{c})^m$. We will show that

$$\|\mathbf{F}(\mathbf{u}_k)\| \le L(c+\epsilon)^k \|\mathbf{F}(\mathbf{u}_0)\| \tag{2.9}$$

for all k. Our assumption on the history that $\|\mathbf{F}(\mathbf{u}_l)\| \le \hat{c}^l \|\mathbf{F}(\mathbf{u}_0)\|$ implies that (2.9) holds for $0 \le k \le m$. Now suppose that (2.9) holds for all $0 \le l \le k$ with $k \ge m$.

We will establish the bound for $k+1$. The analysis has three steps. We first set δ small enough for the iteration to remain in D. We then derive an estimate for $\mathbf{F}(\mathbf{u}_{k+1})$ and finally use that estimate to continue the induction.

Step 1, initialization of δ. Since $\mathbf{G}'$ is continuous in D, there is a nondecreasing function $\rho \in C[0,\infty)$ with $\rho(0) = 0$ so that

$$\|\mathbf{G}'(\mathbf{u}) - \mathbf{G}'(\mathbf{u}^*)\| \le \rho(\|\mathbf{e}\|) \tag{2.10}$$

for all $\mathbf{u} \in D$. This implies that for all $\mathbf{u} \in D$,

$$\mathbf{G}(\mathbf{u}) = \mathbf{G}(\mathbf{u}^*) + \int_0^1 \mathbf{G}'(\mathbf{u}^* + t\mathbf{e})\mathbf{e}\, dt = u^* + \mathbf{G}'(\mathbf{u}^*)\mathbf{e} + \Delta(\mathbf{e}), \tag{2.11}$$

where

$$\|\Delta(\mathbf{e})\| \le \rho(\|\mathbf{e}\|)\|\mathbf{e}\|.$$

Contractivity of $\mathbf{G}$ implies that

$$\|\mathbf{F}(\mathbf{u})\|/(1+c) \le \|\mathbf{e}\| \le \|\mathbf{F}(\mathbf{u})\|/(1-c).$$

Assumption 2.1 implies that

$$\mathcal{B}(\delta) \cap \{\mathbf{u} \,|\, \|\mathbf{F}(\mathbf{u})\| \le \|\mathbf{F}(\mathbf{u}_0)\|\} \subset D.$$

Reduce δ if necessary so that

$$\rho\left(M_\alpha L(c+\epsilon)^{k-m}\delta\frac{1+c}{1-c}\right) \le \frac{c^{m+1}(1-c)}{2M_\alpha}\left(1-\frac{c}{c+\epsilon}\right). \tag{2.12}$$

This implies that

$$\mathbf{w}_k = \sum_{j=0}^{m_k}\alpha_j^k\mathbf{u}_{k-m_k+j} \in D \tag{2.13}$$

for sufficiently small δ because

(2.14)
$$\begin{aligned}\|\mathbf{w}_k-\mathbf{u}^*\| &\le \textstyle\sum_{j=0}^{m_k}|\alpha_j^k|\|\mathbf{e}_{k-m_k+j}\|\\ &\le M_\alpha L(c+\epsilon)^{k-m}\|\mathbf{F}(\mathbf{u}_0)\|/(1-c) \le M_\alpha L(c+\epsilon)^{k-m}\delta(1+c)/(1-c).\end{aligned}$$

Step 2, estimation of $\mathbf{F}(\mathbf{u}_{k+1})$. We may write, for $k \ge m-1$,

$$\begin{aligned}\mathbf{F}(\mathbf{u}_{k+1}) &= \mathbf{G}(\mathbf{u}_{k+1})-\mathbf{u}_{k+1}\\ &= \mathbf{G}(\mathbf{u}_{k+1})-\mathbf{G}(\textstyle\sum_{j=0}^{m_k}\alpha_j^k\mathbf{u}_{k-m_k+j})+\mathbf{G}(\sum_{j=0}^{m_k}\alpha_j^k\mathbf{u}_{k-m_k+j})-\mathbf{u}_{k+1}.\end{aligned}$$

We will estimate the two parts of the sum

$$A_k = \mathbf{G}(\mathbf{u}_{k+1})-\mathbf{G}\left(\sum_{j=0}^{m_k}\alpha_j^k\mathbf{u}_{k-m_k+j}\right)$$

and

$$B_k = \mathbf{G}\left(\sum_{j=0}^{m_k}\alpha_j^k\mathbf{u}_{k-m_k+j}\right)-\mathbf{u}_{k+1}$$

separately.

Using only contractivity of $\mathbf{G}$ and (2.4), we have

(2.15)
$$\begin{aligned}\|A_k\| &= \|\mathbf{G}(\mathbf{u}_{k+1})-\mathbf{G}(\textstyle\sum_{j=0}^{m_k}\alpha_j^k\mathbf{u}_{k-m_k+j})\|\\ &\le c\|\mathbf{u}_{k+1}-\textstyle\sum_{j=0}^{m_k}\alpha_j^k\mathbf{u}_{k-m_k+j}\|\\ &= c\|\textstyle\sum_{j=0}^{m_k}\alpha_j^k(\mathbf{G}(\mathbf{u}_{k-m_k+j})-\mathbf{u}_{k-m_k+j})\|\\ &= c\|\textstyle\sum_{j=0}^{m_k}\alpha_j^k\mathbf{F}(\mathbf{u}_{k-m_k+j})\| \le c\|\mathbf{F}(u_k)\|.\end{aligned}$$

We now estimate B_k. Using (2.12), we have for all $\mathbf{u}\in D$ with

$$\|\mathbf{e}\| \le M_\alpha L(c+\epsilon)^{k-m}\delta(1+c)/(1-c)$$

(2.16)
$$\begin{aligned}\|\Delta(\mathbf{e})\| &\le \rho(\|\mathbf{e}\|)\|\mathbf{F}(\mathbf{u})\|/(1-c)\\ &\le \rho(M_\alpha L(c+\epsilon)^{k-m}\delta(1+c)/(1-c))\|\mathbf{F}(\mathbf{u})\|/(1-c)\\ &\le \tfrac{c^{m+1}}{2M_\alpha}\left(1-\tfrac{c}{c+\epsilon}\right)\|\mathbf{F}(\mathbf{u})\|.\end{aligned}$$

The final stage in the proof is to show that, reducing δ if needed,

$$\|B_k\| \le L(c+\epsilon)^{k+1}\left(1-\frac{c}{c+\epsilon}\right)\|\mathbf{F}(\mathbf{u}_0)\|. \tag{2.17}$$

Recall that

$$\begin{aligned} B_k &= \mathbf{G}(\textstyle\sum_{j=0}^{m_k}\alpha_j^k\mathbf{u}_{k-m_k+j}) - \mathbf{u}_{k+1} \\ &= \mathbf{G}(\textstyle\sum_{j=0}^{m_k}\alpha_j^k\mathbf{u}_{k-m_k+j}) - \textstyle\sum_{j=0}^{m_k}\alpha_j^k\mathbf{G}(\mathbf{u}_{k-m_k+j}). \end{aligned}$$

We use (2.11) to obtain

$$\begin{aligned} \mathbf{G}(\textstyle\sum_{j=0}^{m_k}\alpha_j^k\mathbf{u}_{k-m_k+j}) &= \mathbf{G}(\mathbf{w}_k) = \mathbf{u}^* + \mathbf{G}'(\mathbf{u}^*)\textstyle\sum_{j=0}^{m_k}\alpha_j^k\mathbf{e}_{k-m_k+j} + \Delta(\mathbf{w}_k-\mathbf{u}^*) \\ &= \textstyle\sum_{j=0}^{m_k}\alpha_j^k(\mathbf{u}^* + \mathbf{G}'(\mathbf{u}^*)\mathbf{e}_{k-m_k+j}) + \Delta(\mathbf{w}_k-\mathbf{u}^*) \\ &= \textstyle\sum_{j=0}^{m_k}\alpha_j^k\mathbf{G}(\mathbf{u}_{k-m_k+j}) + \sum_{j=0}^{m_k}\alpha_j^k\Delta(\mathbf{e}_{k-m_k+j}) + \Delta(\mathbf{w}_k-\mathbf{u}^*). \end{aligned}$$

Hence

$$\|B_k\| \le \sum_{j=0}^{m_k}|\alpha_j^k|\|\Delta(\mathbf{e}_{k-m+1})\| + \|\Delta(\mathbf{w}_k-\mathbf{u}^*)\|.$$

We will estimate terms separately. First

$$\begin{aligned} \textstyle\sum_{j=0}^{m_k}|\alpha_j^k|\|\Delta(\mathbf{e}_{k-m+1})\| &\le \tfrac{c^{m+1}}{2M_\alpha}\big(1-\tfrac{c}{c+\epsilon}\big)\textstyle\sum_{j=0}^{m_k}|\alpha_j^k|\|\mathbf{F}(\mathbf{u}_{k-m_k+j})\| \\ &\le \tfrac{c^{m+1}}{2M_\alpha}\big(1-\tfrac{c}{c+\epsilon}\big)\textstyle\sum_{j=0}^{m_k}|\alpha_j^k|L(c+\epsilon)^{k-m_k+j}\|\mathbf{F}(\mathbf{u}_0)\| \\ &\le \tfrac{c^{m+1}}{2}\big(1-\tfrac{c}{c+\epsilon}\big)L(c+\epsilon)^{k-m_k}\|\mathbf{F}(\mathbf{u}_0)\| \\ &\le (L/2)(c+\epsilon)^{k+1}\big(1-\tfrac{c}{c+\epsilon}\big)\|\mathbf{F}(\mathbf{u}_0)\|. \end{aligned} \tag{2.18}$$

Finally, using (2.14) and (2.16),

$$\begin{aligned} \|\Delta(\mathbf{w}_k-\mathbf{u}^*)\| &\le \rho(\|\mathbf{w}_k-\mathbf{u}^*\|)\|\mathbf{F}(\mathbf{w}_k)\|/(1-c) \\ &\le \rho(M_\alpha L(c+\epsilon)^{k-m}\delta(1+c)/(1-c))M_\alpha L(c+\epsilon)^{k-m}\|\mathbf{F}(\mathbf{u}_0)\|/(1-c) \\ &\le (L/2)(c+\epsilon)^{k+1}\big(1-\tfrac{c}{c+\epsilon}\big)\|\mathbf{F}(\mathbf{u}_0)\|. \end{aligned} \tag{2.19}$$

Adding the two estimates (2.18) and (2.19) leads to (2.17).

Step 3, continuation of the induction. Combining (2.15), (2.17), (2.9), and the induction hypotheses, we have

$$\begin{aligned} \|\mathbf{F}(\mathbf{u}_{k+1})\| &\le c\|\mathbf{F}(\mathbf{u}_k)\| + L(c+\epsilon)^{k+1}\big(1-\tfrac{c}{c+\epsilon}\big)\|\mathbf{F}(\mathbf{u}_0)\| \\ &\le \left(Lc(c+\epsilon)^k + L(c+\epsilon)^{k+1}\big(1-\tfrac{c}{c+\epsilon}\big)\right)\|\mathbf{F}(\mathbf{u}_0)\| \\ &\le L(c+\epsilon)^{k+1}\|\mathbf{F}(\mathbf{u}_0)\|. \end{aligned} \tag{2.20}$$

This implies (2.8), which in turn implies (2.7) because ϵ is arbitrary. ∎

Theorem 2.2 and nonsingularity of $\mathbf{F}'(\mathbf{u}^*)$ also imply r-linear convergence of the errors with r-factor c. This extends and sharpens (1.6).

COROLLARY 2.3. *Let the assumptions of Theorem* 2.2 *hold. If* $\mathbf{F}'(\mathbf{u}^*)$ *is nonsingular, then*

$$\limsup_{k\to\infty}\left(\frac{\|\mathbf{e}_k\|}{\|\mathbf{e}_0\|}\right)^{1/k} \le c. \tag{2.21}$$

Proof. We will use Lemma 5.2.1 from [19], which states that if $\mathbf{u}$ is sufficiently near $\mathbf{u}^*$ and $\mathbf{F}'(\mathbf{u}^*)$ is nonsingular, then

$$\frac{\|\mathbf{e}\|}{\|\mathbf{e}_0\|} \le 4\|\mathbf{F}'(\mathbf{u}^*)\|\|\mathbf{F}'(\mathbf{u}^*)^{-1}\|\frac{\|\mathbf{F}(\mathbf{u})\|}{\|\mathbf{F}(\mathbf{u}_0)\|}.$$

Hence

$$\limsup_{k\to\infty}\left(\frac{\|\mathbf{e}_k\|}{\|\mathbf{e}_0\|}\right)^{1/k} \le \lim_{k\to\infty}\left(4\|\mathbf{F}'(\mathbf{u}^*)\|\|\mathbf{F}'(\mathbf{u}^*)^{-1}\|\right)^{1/k}\limsup_{k\to\infty}\left(\frac{\|\mathbf{F}(\mathbf{u}_k)\|}{\|\mathbf{F}(\mathbf{u}_0)\|}\right)^{1/k} \le c,$$

which is (2.21). ∎

3. Numerical example. We will use an example [41] to show how the actual performance of EDIIS and Anderson acceleration can differ, even though the theoretical limiting convergence estimates are identical. Another point of this section is that the solver for the optimization problem can significantly affect the results.

The results in [21] also illustrate this point. Our example is simple enough to directly compare the iteration histories for Picard iteration, EDIIS, and Anderson with the worst-case prediction given by the contractivity constant. We find that when Anderson acceleration performs well, as it does in this example, EDIIS offers no advantage. Moreover, the additional constraint on the optimization problem for the coefficients leads to slower convergence, exactly matching Picard iteration in this case.

The optimization problem for EDIIS requires more care than the linear least squares problem one must solve for Anderson acceleration. The reason for this is that one cannot simply use a QR factorization to solve (1.4). Instead one must apply a more sophisticated iterative solver. The approach of [21] is a direct examination of the boundary of the feasible simplex, which is not practical for a depth much greater than $m = 3$. Since m is small in practice, expressing the optimization problem as a bound-constrained quadratic program is an efficient alternative. References [27, 26] survey the literature on this topic. For example, a bound-constrained quadratic programming code such as the MINQ [29] code is a reasonable choice. However, this approach squares the condition number and can (and did in our testing) result in a singular or nearly singular KKT system and failure of the optimization code's internal linear solvers. The method of [6], while still squaring the condition number, is more robust and terminated without error for this example. The classic method from [14] uses an active set method and the QR factorization to avoid using the normal equations. The approach in [14] performed better in the example here, where the least squares coefficient matrix for the optimization problem is ill-conditioned [41].

The example is the midpoint rule discretization of the Chandrasekhar H-equation [5, 3].

$$\mathcal{F}(H)(\mu) = H(\mu) - \left(1 - \frac{\omega}{2}\int_0^1 \frac{\mu H(\nu)\ d\nu}{\mu+\nu}\right)^{-1} = 0. \tag{3.1}$$

We seek a solution $H^* \in C[0,1]$. When the parameter ω is important we will write H^* as a function $H^*(\mu,\omega)$ of both μ and ω.

The integral equation and its midpoint discretization share the properties that the fixed point map

$$\mathcal{G}(H)(\mu) = \left(1 - \frac{\omega}{2}\int_0^1 \frac{\mu H(\nu)\ d\nu}{\mu+\nu}\right)^{-1}$$

is a contraction for $0 \le \omega < 1$, but not for $\omega = 1$. The Fréchet derivative (and the Jacobian for the discrete case) is singular at the solution for $\omega = 1$, which is a simple fold singularity [17, 28].

In this section we will compare the performance of Picard iteration, Anderson acceleration, and EDIIS for the case $\omega = .5$ on an $N = 100$ point mesh. We terminated the iteration when the residual had decreased by a factor of 10^{-12}.

One interesting result from [41] is that Anderson(m) is more efficient than Newton's method for this example, even in the singular case. In the context of this paper it is important to note that Picard iteration converges faster than one would expect from estimating the contractivity parameter by the spectral radius of the Fréchet derivative of $\mathcal{G}$ at the solution, which is a lower bound for the operator norm of $\mathcal{G}$. From [41]

$$\rho(\mathcal{G}'(H^*)) = 1 - \sqrt{1-\omega} \approx .293.$$

However [18, 20, 2], the solution is analytic in ω and Picard iteration exploits that property to obtain q-linear convergence with q-factor $\le \rho(\mathcal{G}'(H^*))$ and much less for small ω. In fact, if

$$H^*(\mu,\omega) = \sum_{m=0}^{\infty} \omega^m a_m(\mu)$$

is the Taylor expansion of H^* in ω, then the coefficient functions $\{a_m(\mu)\}$ are nonnegative for $0 \le \mu \le 1$. Moreover, the series converges for $\omega = 1$. Hence, if H_k is the kth Picard iteration and $H_0 \equiv 0$, then for all $k \ge 0$ and $\omega, \mu \in [0,1]$,

$$H_k(\mu,\omega) \le H_{k+1}(\mu,\omega) \le H^*(\mu,\omega).$$

All of the above statements about the singularity at $\omega = 1$, the spectral radius of the Fréchet derivative, and the performance of Picard iteration apply to the discrete problem

$$\mathbf{G}(\mathbf{h})_i = \left(1 - \frac{\omega}{2N}\sum_{j=1}^{N} \frac{h_j \mu_i}{\mu_i + \mu_j}\right)^{-1}, \quad 1 \le i \le N. \tag{3.2}$$

In (3.2) $\mu_i = (i - 1/2)/N$ is the ith quadrature node for the N point composite midpoint rule, the vector $\mathbf{h}^*$ is the solution of the discrete problem $\mathbf{h} = \mathbf{G}(\mathbf{h})$, $\mathbf{G}(\mathbf{h}^*)_i$ is the ith component of $\mathbf{G}(\mathbf{h}^*)$, and the ith component of $\mathbf{h}^*$ is $h_i^* \approx H^*(\mu_i)$.

As noted above, the optimization problem (1.7) for EDIIS is harder than the one for Anderson acceleration, and the choice of solver can be important. We compare the method of [14], as implemented in the MATLAB `lsqlin` code with the "active-set" option, with the method from [6], as implemented with the "interior-point" option in `lsqlin`. The method of [6] uses the normal equations and did exhibit problems with ill-conditioning. The computations were done on an Apple Macintosh running

TABLE 3.1
Convergence r-factors.

Anderson	Picard	EDIIS-A	EDIIS-I	$\rho(\mathcal{G}'(H^*))$
No condition limit				
1.06e-02	1.72e-01	1.72e-01	2.62e-01	2.93e-01
Condition limit 10^5				
2.59e-02	1.72e-01	1.72e-01	2.62e-01	2.93e-01

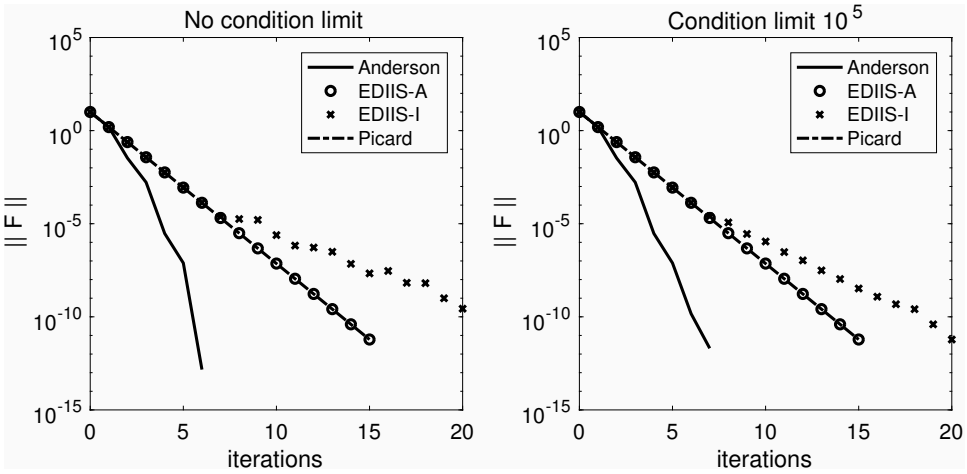

FIG. 3.1. *Residual histories for* $\omega = .5$.

MAC OS 10.13.6 with MATLAB 2017a. The "active-set" option was removed with MATLAB 2017b. The codes that generated Table 3.1 and Figure 3.1 are supplementary materials (see Supplementary_Materials.zip [local/web 5.62KB]) for this paper.

In the left plot of Figure 3.1 we compare Picard iteration, Anderson acceleration, and EDIIS with the active-set option (EDIIS-A) and the interior-point option (EDIIS-I). The depth was $m = 3$ for the Anderson and EDIIS computations. Picard iteration and EDIIS-A are identical. The optimization problem for EDIIS cannot match the results from Anderson acceleration, which has fairly large negative coefficients. Rather, EDIIS-A finds that the coefficients for Picard iteration are optimal.

Table 3.1 compares $\rho(\mathcal{G}'(H^*))$ to the r-factors of the residuals for Anderson acceleration, Picard iteration, and EDIIS. We estimate the r-factors by

$$\left(\frac{\|\mathbf{F}(\mathbf{h}_k)\|}{\|\mathbf{F}(\mathbf{h}_0)\|}\right)^{1/k},$$

where the final iteration upon termination is $\mathbf{h}_k$. Note that, as discussed above, the q-factor for Picard iteration is smaller than the spectral radius. Anderson acceleration also does better than the theory predicts and, in fact, is more efficient than Newton-GMRES [41].

EDIIS-I is the only one of the methods which is sensitive to the ill-conditioning of the optimization problem. We examined this sensitivity by solving the problem twice, once with no limit on the condition number and again by reducing m_k if necessary to limit the condition number to 10^5. This has no effect on EDIIS-A and slightly slows Anderson acceleration down. We show the residual histories in Figure 3.1, where one can clearly see the effect of limiting the condition number. As reported in [41], the

optimization problem becomes more ill-conditioned as the iteration progresses. The figures show that the convergence of EDIIS-I degrades at the 6th iteration, but to a lesser degree when the condition number is limited. Note that the estimated r-factor seems to stabilize near the end and is, in the condition number limited case, back to Picard iteration for the final three iterations, albeit from a worse starting point.

4. Conclusions. The EDIIS algorithm was designed to improve the global convergence properties of the DIIS algorithm, which is also known as Anderson acceleration. We prove global convergence of the iteration and prove a local convergence result that applies to both EDIIS and Anderson acceleration and improves the results in [41]. We observe, as did the inventors of the method [21], that the unmodified version of Anderson acceleration can have better local convergence in practice.

Acknowledgments. The authors are grateful to Elena Jakubikova for making us aware of the global convergence issues in Anderson acceleration, James Nance for pointing out [21], and two very thoughtful referees.

REFERENCES

[1] D. G. Anderson, *Iterative procedures for nonlinear integral equations*, J. ACM, 12 (1965), pp. 547–560, https://doi.org/10.1145/321296.321305.
[2] P. B. Bosma and W. A. DeRooij, *Efficient methods to calculate Chandrasekhar's H-functions*, Astron. Astrophys., 126 (1983), pp. 283–292.
[3] I. W. Busbridge, *The Mathematics of Radiative Transfer*, Cambridge Tracts 50, Cambridge University Press, Cambridge, UK, 1960.
[4] 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, https://doi.org/10.1137/S106482759426955X.
[5] S. Chandrasekhar, *Radiative Transfer*, Dover, New York, 1960.
[6] T. F. Coleman and Y. Li, *On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds*, Math. Program., 67 (1994), pp. 189–224.
[7] A. M. Collier, A. C. Hindmarsh, R. Serban, and C. S. Woodward, *User Documentation for KINSOL* v2.8.0, Tech. report UCRL-SM-208116, Lawrence Livermore National Laboratory, Livermore, CA, 2015.
[8] CVX Research, Inc., *CVX: MATLAB Software for Disciplined Convex Programming, Version* 2.0, http://cvxr.com/cvx, 2012.
[9] P. H. Dedrichs and R. Zeller, *Self-consistency iterations in electronic-structure calculations*, Phys. Rev. B, 28 (1983), pp. 5462–5472.
[10] J. Degroote, K.-J. Bathe, and J. Vierendeels, *Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction*, Comput. & Structures, 97 (2009), pp. 793–801.
[11] H.-R. Fang and Y. Saad, *Two classes of multisecant methods for nonlinear acceleration*, Numer. Linear Algebra Appl., 16 (2009), pp. 197–221, https://doi.org/10.1002/nla.617.
[12] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, Ö. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, and D. J. Fox, *Gaussian* 09*, Revision* A.1, 2009.
[13] D. J. Gardner, C. S. Woodward, D. R. Reynolds, G. Hommes, S. Aubrey, and A. Arsnelis, *Implicit integration methods for dislocation dynamics*, Model. Simul. Mater. Sci. Eng., 23 (2015), 025006.

[14] G. H. Golub and M. A. Saunders, *Linear Least Squares and Quadratic Programming*, Tech. report CS 134, Stanford University, Stanford, CA, 1969.
[15] R. Haelterman, J. Degroote, D. van Heule, and J. Vierendeels, *The quasi-Newton least squares method: A new and fast secant method analyzed for linear systems*, SIAM J. Numer. Anal., 47 (2009), pp. 2347–2368, https://doi.org/10.1137/070710469.
[16] S. Hamilton, M. Berrill, K. Clarno, R. Pawlowski, A. Toth, C. T. Kelley, T. Evans, and B. Philip, *An assessment of coupling algorithms for nuclear reactor core physics simulations*, J. Comput. Phys., 311 (2016), pp. 241–257.
[17] H. B. Keller, *Lectures on Numerical Methods in Bifurcation Theory*, Tata Institute of Fundamental Research, Lectures on Mathematics and Physics, Springer-Verlag, New York, 1987.
[18] C. T. Kelley, *Solution of H-equations by iteration*, SIAM J. Math. Anal., 10 (1979), pp. 844–849, https://doi.org/10.1137/0510080.
[19] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers Appl. Math. 16, SIAM, Philadelphia, 1995, https://doi.org/10.1137/1.9781611970944.
[20] C. T. Kelley and T. W. Mullikin, *Solution by iteration of H-equations in multigroup neutron transport*, J. Math. Phys., 19 (1978), pp. 500–501.
[21] K. N. Kudin, G. E. Scuseria, and E. Cancès, *A black-box self-consistent field convergence algorithm: One step closer*, J. Chem. Phys., 116 (2002), pp. 8255–8261, https://doi.org/10.1063/1.1470195.
[22] L. Lin and C. Yang, *Elliptic preconditioner for accelerating the self-consistent field iteration in Kohn–Sham density functional theory*, SIAM J. Sci. Comput., 35 (2013), pp. S277–S298, https://doi.org/10.1137/120880604.
[23] F. Lindner, M. Mehl, K. Scheufele, and B. Uekermann, *A comparison of various quasi-Newton schemes for partitioned fluid-structure interaction*, in ECCOMAS Coupled Problems in Science and Engineering, B. A. Schrefler, E. Oñate, and M. Papadrakakis, eds., Barcelona, 2015, DIMNE, pp. 477–488.
[24] P. A. Lott, H. F. Walker, C. S. Woodward, and U. M. Yang, *An accelerated Picard method for nonlinear systems related to variably saturated flow*, Adv. Water Res., 38 (2012), pp. 92–101.
[25] K. Miller, *Nonlinear Krylov and moving nodes in the method of lines*, J. Comput. Appl. Math., 183 (2005), pp. 275–287.
[26] J. J. Moré and G. Toraldo, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377–400.
[27] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113, https://doi.org/10.1137/0801008.
[28] T. W. Mullikin, *Some probability distributions for neutron transport in a half-space*, J. Appl. Probability, 5 (1968), pp. 357–374.
[29] A. Neumaier, *MINQ: General Definite and Bound Constrained Indefinite Quadratic Programming*, http://www.mat.univie.ac.at/~neum/software/minq/, 1998.
[30] C. W. Oosterlee and T. Washio, *Krylov subspace acceleration of nonlinear multigrid with application to recirculating flow*, SIAM J. Sci. Comput., 21 (2000), pp. 1670–1690, https://doi.org/10.1137/S1064827598338093.
[31] F. A. Potra and H. Engler, *A characterization of the behavior of the Anderson acceleration on linear problems*, Linear Algegra Appl., 438 (2013), pp. 1002–1011.
[32] P. Pulay, *Convergence acceleration of iterative sequences. The case of SCF iteration*, Chem. Phys. Lett., 73 (1980), pp. 393–398, https://doi.org/10.1016/0009-2614(80)80396-4.
[33] P. Pulay, *Improved SCF convergence acceleration*, J. Comput. Chem., 3 (1982), pp. 556–560, https://doi.org/10.1002/jcc.540030413.
[34] T. Rohwedder and R. Schneider, *An analysis for the DIIS acceleration method used in quantum chemistry calculations*, J. Math. Chem., 49 (2011), pp. 1889–1914, https://doi.org/10.1007/s10910-011-9863-y.
[35] Y. Saad, J. R. Chelikowsky, and S. M. Shontz, *Numerical methods for electronic structure calculations of materials*, SIAM Rev., 52 (2010), pp. 3–54, https://doi.org/10.1137/060651653.
[36] H. B. Schlegel and J. J. W. McDouall, *Do you have SCF stability and convergence problems?*, in Computational Advances in Organic Chemistry: Molecular Structure and Reactivity, C. Ögretir and I. G. Csizmadia, eds., Kluwer, Dordrecht, The Netherlands, 1991, pp. 167–185.

[37] R. Schneider, T. Rohwedder, A. Neelov, and J. Blauert, *Direct minimization for calculating invariant subspaces in density functional computations of the electronic structure*, J. Comput. Math., 27 (2008), pp. 360–387.
[38] *SUNDIALS: SUite of Nonlinear and DIfferential/ALgebraic Solvers*, https://computation.llnl.gov/projects/sundials.
[39] A. Toth, *A Theoretical Analysis of Anderson Acceleration and Its Application in Multiphysics Simulation for Light-Water Reactors*, Ph.D. thesis, North Carolina State University, Raleigh, NC, 2016.
[40] A. Toth, J. A. Ellis, T. Evans, S. Hamilton, C. T. Kelley, R. Pawlowski, and S. Slattery, *Local improvement results for Anderson acceleration with inaccurate function evaluations*, SIAM J. Sci. Comput., 39 (2017), pp. S47–S65, https://doi.org/10.1137/16M1080677.
[41] A. Toth and C. T. Kelley, *Convergence analysis for Anderson acceleration*, SIAM J. Numer. Anal., 53 (2015), pp. 805–819, https://doi.org/10.1137/130919398.
[42] A. Toth, C. T. Kelley, S. Slattery, S. Hamilton, K. Clarno, and R. Pawlowski, *Analysis of Anderson acceleration on a simplified neutronics/thermal hydraulics system*, in Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method, 2015.
[43] A. Toth and R. Pawlowski, *NOX::Solver::AndersonAcceleration Class Reference*, https://trilinos.org/docs/dev/packages/nox/doc/html/classNOX_1_1Solver_1_1AndersonAcceleration.html, 2015.
[44] H. F. Walker and P. Ni, *Anderson acceleration for fixed-point iterations*, SIAM J. Numer. Anal., 49 (2011), pp. 1715–1735, https://doi.org/10.1137/10078356X.
[45] T. Washio and C. Oosterlee, *Krylov subspace acceleration for nonlinear multigrid schemes*, Electron. Trans. Numer. Anal., 6 (1997), pp. 271–290.
[46] D. Young, *Computational Chemistry: A Practical Guide for Applying Techniques to Real World Problems*, Wiley, New York, 2001.
[47] D. Young, *SCF Convergence and Chaos Theory*, http://www.ccl.net/cca/documents/dyoung/topics-orig/converge.html, 2001.