TU Delft

Delft University of Technology

Asymptotic convergence analysis and influence of initial guesses on composite Anderson acceleration

Chen, Kewang; Vuik, Cornelis

DOI
10.1007/s10444-023-10095-3

Publication date
2023

Document Version
Final published version

Published in
Advances in Computational Mathematics

Citation (APA)
Chen, K., & Vuik, C. (2023). Asymptotic convergence analysis and influence of initial guesses on composite Anderson acceleration. *Advances in Computational Mathematics*, *49*(6), Article 94. https://doi.org/10.1007/s10444-023-10095-3

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Advances in Computational Mathematics (2023) 49:_####################_
https://doi.org/10.1007/s10444-023-10095-3

Asymptotic convergence analysis and influence of initial guesses on composite Anderson acceleration

Kewang Chen[1,2] · **Cornelis Vuik**[3]

Received: 17 October 2022 / Accepted: 28 November 2023 / Published online: 13 December 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Although Anderson acceleration AA(m) has been widely used to speed up nonlinear solvers, most authors are simply using and studying the stationary version of Anderson acceleration. The behavior and full potential of the non-stationary version of Anderson acceleration methods remain an open question. Motivated by the hybrid linear solver GMRESR (GMRES Recursive), we recently proposed a set of non-stationary Anderson acceleration algorithms with dynamic window sizes AA(m,AA(n)) for solving both linear and nonlinear problems. Significant gains are observed for our proposed algorithms but these gains are not well understood. In the present work, we first consider the case of using AA(m,AA(1)) for accelerating linear fixed-point iteration and derive the polynomial residual update formulas for non-stationary AA(m,AA(1)). Like stationary AA(m), we find that AA(m,AA(1)) with general initial guesses is also a multi-Krylov method and possesses a memory effect. However, AA(m,AA(1)) has higher order degree of polynomials and a stronger memory effect than that of AA(m) at the k-th iteration, which might explain the better performance of AA(m,AA(1)) compared to AA(m) as observed in our numerical experiments. Moreover, we further study the influence of initial guess on the asymptotic convergence factor of $AA(1, AA(1))$. We show a scaling invariance property of the initial guess x_0 for the AA(1,AA(1)) method in the linear case. Then, we study the root-linear asymptotic convergence factor under scaling of the initial guess and we explicitly indicate the dependence of root-linear asymptotic convergence factors on the initial guess. Lastly, we numerically examine the influence of the initial guess on the asymptotic convergence factor of AA(m) and AA(m,AA(n)) for both linear and nonlinear problems.

Keywords Non-stationary Anderson acceleration · Fixed-point iteration · Dynamic window sizes · Krylov method

Communicated by: Peter Benner

✉ Kewang Chen
kwchen@nuist.edu.cn

Extended author information available on the last page of the article

Mathematics Subject Classification (2010) MSC 65F10 · MSC 65H10

1 Introduction

In this paper, we study convergence acceleration methods for the following fixed-point problem:

$$x = q(x),\ q : R^n \to R^n,$$

which can also be employed to solve nonlinear equation systems $f(x) = 0$. The associated basic fixed-point iteration is given in Algorithm 1.

Algorithm 1 Picard iteration.

Given: x_0.
for $k = 0, 1, 2, \cdots$ **do**
  Set $x_{k+1} = q(x_k)$.
end for

The main concern in real applications is that Algorithm 1 may not converge or it converges unacceptably slowly and may take a very long time to compute an accurate result. Thus, many acceleration methods are proposed to solve this slow convergence issue. In 1962, Anderson [1] proposed a technique for accelerating the convergence of Picard iterations which is originally called the Extrapolation Algorithm. Since then, this acceleration method has enjoyed remarkable success and wide usage in different areas, especially in computational chemistry (where it is known as Pulay mixing) and electronic structure computations (where it is known as Anderson mixing). The technique is now called Anderson acceleration (AA) in the applied mathematics community. Different to the Picard iteration in Algorithm 1, which uses only one previous iterate, the Anderson Acceleration method $AA(m)$ proceeds by linearly recombining a list of previous m iterates in a way such that approximately minimizes the linearized fixed-point residual. We summarize the typical form of Anderson Acceleration with damping in Algorithm 2. Here, f_k is the residual for the kth iteration; m is the window size which indicates how many history residuals will be used in the algorithm. In many papers, this is usually a fixed number during the procedure. Typically, the value of m is not larger than 3 in the early days of applications and now this value could be as large as up to 100. Anderson [2] We point out that $\gamma_k \in (0, 1]$ is a damping factor (or a relaxation parameter) at kth iteration, where

$$\gamma_k = \begin{cases} 1, & \text{AA without damping,} \\ \gamma,\ \text{(a constant independent of } k) & \text{AA with constant damping,} \\ \gamma_k,\ \text{(depending on } k) & \text{AA with dynamic damping.} \end{cases}$$

Algorithm 2 Anderson acceleration: **AA(m)**.

Given: x_0 and $m \geq 1$.
Set: $x_1 = q(x_0)$.
for $k = 0, 1, 2, \cdots$ **do**
Set: $m_k = \min\{m, k\}$.
Set: $F_k = (f_{k-m_k}, \cdots, f_k)$, where $f_i = q(x_i) - x_i$.
Determine: $\beta^{(k)} = \left(\beta_0^{(k)}, \cdots, \beta_{m_k}^{(k)}\right)^T$ that solves

$$\min_{\beta=(\beta_0,\cdots,\beta_{m_k})^T} \|F_k\beta\|_2 \ \ s.t. \sum_{i=0}^{m_k} \beta_i = 1.$$

Set: $x_{k+1} = (1-\gamma_k)\sum_{i=0}^{m_k} \beta_i^{(k)} x_{k-m_k+i} + \gamma_k \sum_{i=0}^{m_k} \beta_i^{(k)} q(x_{k-m_k+i})$.
end for

We can also formulate this constrained optimization problem as an equivalent unconstrained least-squares problem [31]:

$$\min_{(\omega_1,\cdots,\omega_{m_k})^T} \left\| f_k + \sum_{i=1}^{m_k} \omega_i (f_{k-i} - f_k) \right\|_2 \tag{1}$$

One can easily recover the original problem by setting $\omega_0 = 1 - \sum_{i=1}^{m_k} \omega_i$.

Anderson acceleration methods in Algorithm 2 are considered "essentially equivalent" to the direct inversion on the iterative subspace method (DIIS) [19, 27, 28] and the nonlinear GMRES methods. Carlson and Miller [6, 20, 21, 34, 36] They are also in a broad category of methods based on quasi-Newton updating. Eirola and Nevanlinna [12, 14–16, 37] For example, Walker and Ni [34] proved that AA without truncation ($m = \infty$) is equivalent in a certain sense to the GMRES method on linear problems. On nonlinear problems, Fang and Saad [15] had shown a remarkable relationship between AA and quasi-Newton methods, which utilizes the previous iterates to approximate the inverse Jacobian. However, one advantage of AA is that it does not require the expensive computation or approximation of Jacobians or Jacobian-vector products, especially for large-scale problems.

It is only recently that the theoretical results have been obtained on convergence analysis. In 2015, Toth and Kelley [31] first proved the locally r-linearly convergent result for the stationary Anderson acceleration without damping under the condition that the fixed point map is a contraction and the coefficients in the linear combination remain bounded. Later, Evans et al. [13] extended their result to AA with general damping factors. In 2019, Pollock et al. [25] applied AA to the Picard iteration for solving steady incompressible Navier–Stokes equations and proved that the acceleration improves the convergence rate of the Picard iteration. More recently, De Sterck et al. [11] extended the result to a more general fixed-point iteration $x = q(x)$, given knowledge of the spectrum of $q'(x)$ at fixed-point x^* and Wang et al. [35] extended the result to study the asymptotic linear convergence speed of stationary AA applied to Alternating Direction Method of Multipliers (ADMM) method. We note here that the stationary AA in the papers of De Sterck et al. [11] and Wang et al. [35] is stationary

in a different sense: in those papers, the β_i^k of Algorithm 1 are fixed and do not depend on the iteration, so the β_i are stationary. A sharper local convergence estimation and global convergence result of AA remain an active research area. For more related results about Anderson acceleration and its applications, we refer the interested readers to papers [3, 5, 10, 23, 29, 30, 38, 39] and references therein.

Although AA has been used for decades, most authors are simply using and analyzing the stationary version of Anderson acceleration with fixed window size and a constant damping factor. The behavior and potential of the non-stationary versions of the Anderson Acceleration method with dynamic window sizes have not been deeply studied and few results have been reported. Generally, there are two main possible directions for producing non-stationary AA. One is choosing different damping factors γ_k in each iteration, see our recent work on the non-stationary Anderson acceleration algorithm with optimized damping (AAoptD). Chen and Vuik [8] The other way of making AA to be a non-stationary algorithm is to alternate the window size during iterations. Since most efficient linear solvers use composable algorithmic components, [4, 17] similar ideas can be used for AA(m) and AA(n) to solve nonlinear systems. Pollock and Rebholz [24] proposed a strategy to change window sizes based on the residuals. The window size m_k is kept at a small to moderate value (2 to 5) until the residual drops below a given threshold, on the order of 10^{-2} or 10^{-3}, then m_k is increased to a higher steady level, for instance, $m = 10$. This approach is appropriate for problems where the initial residual is moderately scaled. More recently, motivated by the hybrid linear solver GMRESR (GMRES Recursive), [32, 33], we propose a systematic way to dynamically alternate the window size by the multiplicative composite combination, which means we apply stationary AA(m) in the outer loop and apply stationary AA(n) in the inner loop [7]. For these non-stationary AA methods, significant gains are observed over stationary AA for solving the Bratu problem and the convection-diffusion problem, where large windows m are needed. However, the reason why those composite Anderson acceleration methods may work better than the stationary AA methods is not well understood. Motivated by De Sterck and He's recent papers, [9, 10] to throw light on the behavior of non-stationary Anderson acceleration methods, we conduct the asymptotic convergence analysis and study the influence of initial guess on the composite Anderson acceleration methods in the present work.

The rest of the paper is organized as follows. Our motivation and the key procedure of the algorithms are provided in Section 2; In Section 3, for linear problems, we derive polynomial residual update formulas for $AA(m, AA(1))$ and show that it is a multi-Krylov method; In Section 4, we further study the influence of initial guess on the asymptotic convergence factor of $AA(1, AA(1))$ and explicitly indicate the dependence of root-linear asymptotic convergence factors on the initial condition; Some numerical results and discussions are presented in Section 7; conclusions follow in Section 8.

2 Composite Anderson acceleration

In this section, we briefly summarize the motivations and the procedure of the composite Anderson acceleration method proposed in our recent paper [7].

2.1 Motivation

As we know, many efficient linear solvers use composable algorithmic components. Similar ideas can be used for Anderson acceleration method to accelerate nonlinear systems. Van der Vorst and Vuik [32] developed a hybrid method GMRESR (GMRES Recursive) which consists of an outer and inner loop. In the inner loop, one approximates the solution of a linear system by GMRES to find a good search direction. This method was also further investigated by Vuik [33]. If GMRES(m) does not stagnate in m iteration steps, it was proved that GMRES-Recursive(m) converges at least as fast as GMRES(m). On the other hand, it was proved by Walker and Ni [34] that AA without restarting ($m = \infty$) is equivalent in a certain sense to the GMRES method on linear problems. Therefore, motivated by the GMRESR method, to solve nonlinear problems, we propose a systematic way to dynamically alternate the window size in AA by using the multiplicative composition, which means we apply stationary AA(m) first in the outer loop and then apply stationary AA(n) in the inner loop. It is somewhat like using AA(n) to precondition AA(m).

2.2 Multiplicative composition of different window sizes

In this part, we provide a systematic way to dynamically alternate the window size m by multiplicative composition. We start with composite stationary AA(m) with stationary $AA(0)$ (i.e., Picard iteration) in each iteration. This means that after applying one step of $AA(m)$ without damping, we get,

$$x_{k+1/2} = \sum_{i=0}^{m_k} \beta_i^{(k)} q(x_{k-m_k+i}).$$

Then, we take the result $x_{k+1/2}$ as an input $\hat{x}_0 = x_{k+1/2}$ and apply Picard iteration in the inner loop:

$$\hat{x}_{j+1} = q(\hat{x}_j).$$

Putting these two steps together, we have the following non-stationary algorithm $AA(m, AA(0))$ as in Algorithm 3. Suppose we just do a single inner loop iteration in

Algorithm 3 Anderson acceleration with dynamic window-sizes: **AA(m, AA(0))**.

Given: x_0, $iterM$, $iterN$ and $m \geq 1$.
Set: $x_1 = q(x_0)$.
for $k = 1, 2, \cdots, iterM$ **do**
  Set: $\mathbf{x^m_{k+1/2}} \leftarrow$ **apply one step of AA(m;** $\{\mathbf{x_k}\}$**) as given in Algorithm 2.**
  Set: $\hat{x}_0 = x^m_{k+1/2}$
  for $j = 0, 1, 2, \cdots, iterN$ **do**
    Set: $\hat{\mathbf{x}}_{\mathbf{j+1}} \leftarrow$ **apply one step of Picard iteration on** $\hat{x}_j$.
  end for
  Set: $x_{k+1} = \hat{x}_{iterN}$
end for

Algorithm 3, the total amount of work of $AA(m, AA(0))$ in each iteration is much less than that of applying stationary $AA(m)$ twice. Algorithm 3, also means that we may "turn off" the acceleration for a while and then turn on the acceleration. However, the performance can be better than stationary $AA(m)$, see our numerical experiments in Section 7.

More generally, we apply stationary $AA(m)$ in the outer loop and apply stationary $AA(n)$ in the inner loop. So, in each iteration, after applying $AA(m)$, we get,

$$x_{k+1/2} = \sum_{i=0}^{m_k} \beta_i^{(k)} q(x_{k-m_k+i}).$$

Then, we apply $AA(n)$ with the initial guess $x_0 = x_{k+1/2}$ for $iterN$ iterations:

$$x_{k+1} \leftarrow \text{applying AA(n) with initial guess } \mathrm{x}_0 = \mathrm{x}_{\mathrm{k}+1/2}.$$

In other words, the multiplicative composition reads

$$x_{k+1} = AA(m, AA(n)).$$

We summarize this in the following algorithm in Algorithm 4.

Algorithm 4 Anderson acceleration with dynamic window-sizes: **AA(m, AA(n))**.

Given: $x_0, m, n, iterM$ and $iterN$ (with $iterN \geq n$).
Set: $x_1 = q(x_0)$.
for $k = 1, 2, \cdots iterM$ **do**
  Set: $\mathbf{x_{k+1/2}} \leftarrow$ **apply one step of AA(m;** $\{\mathbf{x_k}\}$**) as given in Algorithm 2.**
  Set: $\hat{x}_0 = x_{k+1/2}$
  for $j = 0, 1, 2, \cdots iterN$ **do**
    Set: $\hat{\mathbf{x}}_{\mathbf{j+1}} \leftarrow$ **apply one step of AA(n;**$\{\hat{\mathbf{x}}_{\mathbf{j}}\}$**) as given in Algorithm 2.**
  end for
  Set: $x_{k+1} = \hat{x}_{iterN}$
end for

Remark 2.1 There is a lot of variety here. Let m and n be the window size used in the outer loop and inner loop, respectively. And $iterM$ and $iterN$ be the total numbers of iterations for the outer loop and inner loop, respectively. In the present work, we report some results for the case where $m > n$ and $iterM \gg iterN$, which means the window size used in the inner loop is smaller than that used in the outer loop and the maximum number of iterations of the inner loop is much smaller than that of the outer loop. For example, one can choose $n = 1$ and $iterN = 1$. As we know, $AA(m, AA(n))$ will do extra inner loop iterations, so it is not fair to compare the residuals of $AA(m, AA(n))$ with $AA(m)$ by iterations. Since for many applications, the computation of function evaluation is the most expensive component, thus we could compare the convergence results of $AA(m, AA(1))$ with $AA(m)$ by calculating the residual per function evaluation of $q(x)$. See more discussions in Section 7.

Moreover, we summarize the memory requirements for the algorithms in Table 1. For some problems, memory storage could be crucial. It was shown in our recent paper [7] that the non-stationary AA methods with smaller window sizes usually perform better than the stationary AA algorithm with very large window sizes, which means our proposed non-stationary AA methods may significantly reduce the storage requirements. For more details on the development of these non-stationary AA methods and their performances in solving the Bratu and convection-diffusion problems, we refer the readers to our recent paper [7].

3 AA(m,AA(1)) as a multi-Krylov space method for linear problems

Recall that the order-s Krylov subspace generated by a matrix T and a vector v is the linear subspace spanned by the images of v under the first s power of T:

$$\mathbb{K}_s(T, v) = \left\{ v, Tv, \cdots, T^{s-1}v \right\}. \tag{2}$$

Then, the multi-Krylov space of order-s generated by a matrix T and $m+1$ vectors $\{r_j\}_{j=0}^m$ is: [9]

$$\left\{ \mathbb{K}_s(T, r_j) \right\}_{j=0}^m.$$

In this section, we first establish links between $AA(m, AA(1))$ applied to linear problems and Krylov methods. Then, we compare the residual polynomials obtained by $AA(m, AA(1))$ and $AA(m)$. Let us focus on $AA(m, AA(1))$ applied to the linear case, that is,

$$q(x) = Mx + b,$$

where the fixed point satisfies $Ax^* = b$ with $A = I - M$. We assume that A is nonsingular and we exclude the trivial case that $A = I$ and $M = 0$. For numerical simulations, we also test the case when $q(x)$ is a nonlinear operator.

3.1 Residual polynomials for AA(1,AA(1))

In this part, we derive the polynomial update formulas for the residual of $AA(1, AA(1))$ with inner loop iteration $iterN = 1$. Clearly, $AA(1)$ is a Krylov subspace method [9]. We now derive the residual r_k generated by $AA(1, AA(1))$ iteration. Given x_0, let $x_1 = q(x_0)$. For the out loop iteration, we have

$$r_0 = x_0 - q(x_0) = x_0 - (Mx_0 + b) = (I - M)x_0 - b = Ax_0 - b.$$

Table 1 Memory requirements

Methods	Memory
AA(m)	$m+1$ vectors in memory
AA(m,AA(n))	$m+n+2$ vectors in memory

Since $x_1 = q(x_0) = Mx_0 + b$, we obtain that

$$\begin{aligned} r_1 = x_1 - q(x_1) &= Ax_1 - b \\ &= A(Mx_0 + b) - b \\ &= M(Ax_0 + Ab) - b \\ &= M(Ax_0) + Ab - b \\ &= M(Ax_0 - b + b) + Ab - b \\ &= Mr_0 + Mb + Ab - b \\ &= Mr_0. \end{aligned} \tag{3}$$

We apply $AA(1)$ to $\{x_0, x_1\}$ to get

$$x_{1+1/2} = \left(1 + \beta_1^{out}\right) q(x_1) - \beta_1^{out} q(x_0).$$

For the inner loop, set

$$\bar{x}_0^1 = x_{1+1/2} = \left(1 + \beta_1^{out}\right) q(x_1) - \beta_1^{out} q(x_0),$$

then we have,

$$\begin{aligned} \bar{r}_0^1 = \bar{x}_0^1 - q(\bar{x}_0^1) &= A\bar{x}_0^1 - b \\ &= Ax_{1+1/2} - b \\ &= A\left(1 + \beta_1^{out}\right)(Mx_1 + b) - \beta_1^{out} A(Mx_0 + b) - b \\ &= \left(1 + \beta_1^{out}\right)(AMx_1 + Ab) - \beta_1^{out}(MAx_0 + Ab) - b \\ &= \left(1 + \beta_1^{out}\right) M(Ax_1 - b) - \beta_1^{out} M(Ax_0 - b) + Mb + Ab - b \\ &= \left(1 + \beta_1^{out}\right) Mr_1 - \beta_1^{out} Mr_0 \end{aligned} \tag{4}$$

Next, we calculate

$$\bar{x}_1^1 = q(\bar{x}_0^1)$$

and

$$\bar{r}_1^1 = \bar{x}_1^1 - q(\bar{x}_1^1) = A\bar{x}_1^1 - b = A(M\bar{x}_0^1 + b) - b = M\bar{r}_0^1.$$

Then, we apply $AA(1)$ in the inner loop for $\{\bar{x}_0^1, \bar{x}_1^1\}$ to calculate

$$\bar{x}_2^1 = \left(1 + \beta_1^{in}\right) q(\bar{x}_1^1) - \beta_1^{in} q(\bar{x}_0^1) \tag{5}$$

and

$$\begin{aligned}\bar{r}_2^1 = \bar{x}_2^1 - q(\bar{x}_2^1) &= A\bar{x}_2^1 - b \\ &= \left(1+\beta_1^{in}\right) M\bar{r}_1^1 - \beta_1^{in} M\bar{r}_0^1 \\ &= \left(1+\beta_1^{in}\right) M^2\bar{r}_0^1 - \beta_1^{in} M\bar{r}_0^1 \\ &= \left[\left(1+\beta_1^{in}\right) M^2 - \beta_1^{in} M\right]\bar{r}_0^1\end{aligned} \tag{6}$$

At the end of the inner loop, we set

$$x_2 = \bar{x}_2^1,$$

therefore, we finally obtain

$$\begin{aligned}r_2 = \bar{r}_2^1 &= \left[\left(1+\beta_1^{in}\right) M^2 - \beta_1^{in} M\right]\bar{r}_0^1 \\ &= \left[\left(1+\beta_1^{in}\right) M^2 - \beta_1^{in} M\right]\left[\left(1+\beta_1^{out}\right) Mr_1 - \beta_1^{out} Mr_0\right].\end{aligned} \tag{7}$$

Then, we go back to the outer loop again, we apply $AA(1)$ to $\{x_1, x_2\}$ to get

$$x_{2+1/2} = \left(1+\beta_2^{out}\right) q(x_2) - \beta_2^{out} q(x_1).$$

For the inner loop, we set

$$\bar{x}_0^2 = x_{2+1/2} = \left(1+\beta_2^{out}\right) q(x_2) - \beta_2^{out} q(x_1),$$

then we have,

$$\begin{aligned}\bar{r}_0^2 = \bar{x}_0^2 - q(\bar{x}_0^2) &= A\bar{x}_0^2 - b \\ &= Ax_{2+1/2} - b \\ &= \left(1+\beta_2^{out}\right) Mr_2 - \beta_2^{out} Mr_1\end{aligned} \tag{8}$$

Similarly, we have

$$\bar{x}_1^2 = q(\bar{x}_0^2)$$

and

$$\bar{r}_1^2 = \bar{x}_1^2 - q(\bar{x}_1^2) = A\bar{x}_1^2 - b = A(M\bar{x}_0^2 + b) - b = M\bar{r}_0^2.$$

Again, we apply $AA(1)$ in the inner loop for $\{\bar{x}_0^2, \bar{x}_1^2\}$ to calculate

$$\bar{x}_2^2 = \left(1+\beta_2^{in}\right) q(\bar{x}_1^2) - \beta_2^{in} q(\bar{x}_0^2) \tag{9}$$

and

$$\begin{aligned}\bar{r}_2^2 = \bar{x}_2^2 - q(\bar{x}_2^2) &= A\bar{x}_2^2 - b \\ &= \left(1+\beta_2^{in}\right) M\bar{r}_1^2 - \beta_2^{in} M\bar{r}_0^2 \\ &= \left[\left(1+\beta_2^{in}\right) M^2 - \beta_2^{in} M\right]\bar{r}_0^2.\end{aligned} \tag{10}$$

At the end of the inner loop, set

$$x_3 = \bar{x}_2^2,$$

we obtain

$$\begin{aligned}r_3 = \bar{r}_2^2 &= \left[\left(1+\beta_2^{in}\right) M^2 - \beta_2^{in} M\right]\bar{r}_0^2 \\ &= \left[\left(1+\beta_2^{in}\right) M^2 - \beta_2^{in} M\right]\left[\left(1+\beta_2^{out}\right) Mr_2 - \beta_2^{out} Mr_1\right].\end{aligned} \tag{11}$$

Similarly, one can get, for any $k > 3$

$$r_k = \left[\left(1+\beta_{k-1}^{in}\right) M^2 - \beta_{k-1}^{in} M\right]\left[\left(1+\beta_{k-1}^{out}\right) Mr_{k-1} - \beta_{k-1}^{out} Mr_{k-2}\right]. \tag{12}$$

In summary, we have

$$\begin{aligned}
r_1 &= Mr_0, \quad (M^1) \\
p_1(M) &= M, \\
r_2 &= \left[\left(1+\beta_1^{in}\right) M^2 - \beta_1^{in} M\right]\left[\left(1+\beta_1^{out}\right) Mr_1 - \beta_1^{out} Mr_0\right] \\
&= \left[\left(1+\beta_1^{in}\right) M^2 - \beta_1^{in} M\right]\left[\left(1+\beta_1^{out}\right) Mp_1(M) - \beta_1^{out} M\right] r_0, \quad (M^4) \\
p_2(M) &= \left[\left(1+\beta_1^{in}\right) M^2 - \beta_1^{in} M\right]\left[\left(1+\beta_1^{out}\right) Mp_1(M) - \beta_1^{out} M\right] \\
r_3 &= \left[\left(1+\beta_2^{in}\right) M^2 - \beta_2^{in} M\right]\left[\left(1+\beta_2^{out}\right) Mr_2 - \beta_2^{out} Mr_1\right] \\
&= \left[\left(1+\beta_2^{in}\right) M^2 - \beta_2^{in} M\right]\left[\left(1+\beta_2^{out}\right) M \cdot p_2(M) - \beta_2^{out} M \cdot p_1(M)\right] r_0, \quad (M^7) \\
p_3(M) &= \left[\left(1+\beta_2^{in}\right) M^2 - \beta_2^{in} M\right]\left[\left(1+\beta_2^{out}\right) M \cdot p_2(M) - \beta_2^{out} M \cdot p_1(M)\right], \\
r_4 &= \left[\left(1+\beta_3^{in}\right) M^2 - \beta_3^{in} M\right]\left[\left(1+\beta_3^{out}\right) Mr_3 - \beta_3^{out} Mr_2\right] \\
&= \left[\left(1+\beta_3^{in}\right) M^2 - \beta_3^{in} M\right]\left[\left(1+\beta_3^{out}\right) M \cdot p_3(M) - \beta_3^{out} M \cdot p_2(M)\right] r_0, \quad (M^{10}) \\
p_4(M) &= \left[\left(1+\beta_3^{in}\right) M^2 - \beta_3^{in} M\right]\left[\left(1+\beta_3^{out}\right) M \cdot p_3(M) - \beta_3^{out} M \cdot p_2(M)\right], \\
&\cdots \\
r_k &= \left[\left(1+\beta_{k-1}^{in}\right) M^2 - \beta_{k-1}^{in} M\right]\left[\left(1+\beta_{k-1}^{out}\right) Mr_{k-1} - \beta_{k-1}^{out} Mr_{k-2}\right] \\
&= \left[\left(1+\beta_{k-1}^{in}\right) M^2 - \beta_{k-1}^{in} M\right]\left[\left(1+\beta_{k-1}^{out}\right) M \cdot p_{k-1}(M) - \beta_{k-1}^{out} M \cdot p_{k-2}(M)\right] r_0, \quad (M^{3k-2}), \\
p_k(M) &= \left[\left(1+\beta_{k-1}^{in}\right) M^2 - \beta_{k-1}^{in} M\right]\left[\left(1+\beta_{k-1}^{out}\right) M \cdot p_{k-1}(M) - \beta_{k-1}^{out} M \cdot p_{k-2}(M)\right], \\
&\cdots
\end{aligned} \tag{13}$$

We summarize these results in the following Proposition.

Proposition 1 *AA(1,AA(1)) iteration with general initial guess x_0 is a Krylov method. The residual can be expressed as*

$$r_{k+1} = p_{k+1}(M)r_0, \tag{14}$$

where the residual polynomials satisfy the recurrence relation

$$p_{k+1}(\lambda) = \lambda^2 \left[\left(1+\beta_k^{in}\right)\lambda - \beta_k^{in}\right]\left[\left(1+\beta_k^{out}\right)\cdot p_k - \beta_k^{out}\cdot p_{k-1}\right], \quad k \geq 1 \tag{15}$$

with $p_0(\lambda) = 1$, $p_1(\lambda) = \lambda$. Moreover, $p_k(\lambda)$ is a polynomial with degree at most $3k-2$ and $p_k(1) = 1$ and $p_k(0) = 0$.

Compared with the results from De Sterck and He's paper [9] for $AA(1)$, we obtain the highest possible degree of the residual polynomials for $AA(m, AA(1))$in Table 2.

3.2 Residual polynomials for AA(m,AA(1))

Given initial guesses $\{x_0, x_1, \cdots, x_m\}$, we have related residuals $\{r_0, r_1, \cdots, r_m\}$. For iteration $k = m$, we apply AA(m) to $\{x_0, x_1, \cdots, x_m\}$ to get

$$x_{m+1/2} = \left(1 + \sum_{i=1}^{m} \beta_{(i,m)}^{out}\right) M x_m - \sum_{i=1}^{m} \beta_{(i,m)}^{out} M x_{m-i} + b \tag{16}$$

and

$$r_{m+1/2} = \left(1 + \sum_{i=1}^{m} \beta_{(i,m)}^{out}\right) M r_m - \sum_{i=1}^{m} \beta_{(i,m)}^{out} M r_{m-i}. \tag{17}$$

Then, for the inner loop, we have

$$\bar{x}_0^m = x_{m+1/2}$$

and

$$\bar{r}_0^m = r_{m+1/2} = \left(1 + \sum_{i=1}^{m} \beta_{(i,m)}^{out}\right) M r_m - \sum_{i=1}^{m} \beta_{(i,m)}^{out} M r_{m-i}. \tag{18}$$

Next, we calculate

$$\bar{x}_1^m = q(\bar{x}_0^m)$$

Table 2 Highest order (at most) term of residual polynomials at $3k$-th function evaluations

Methods	Highest order (at most) term in residual polynomials
AA(1)	M^{3k}
AA(1,AA(1))	M^{3k-2}

and

$$\bar{r}_1^m = \bar{x}_1^m - q(\bar{x}_1^m) = A\bar{x}_1^m - b = A(M\bar{x}_0^m + b) - b = M\bar{r}_0^m.$$

Then, we apply $AA(1)$ in the inner loop for $\{\bar{x}_0^m, \bar{x}_1^m\}$ to calculate

$$\bar{x}_2^m = \left(1 + \beta_{(1,m)}^{in}\right) q(\bar{x}_1^m) - \beta_{(1,m)}^{in} q(\bar{x}_0^m) \tag{19}$$

and

$$\begin{aligned} \bar{r}_2^m = \bar{x}_2^m - q(\bar{x}_2^m) &= A\bar{x}_2^m - b \\ &= \left(1 + \beta_{(1,m)}^{in}\right) M\bar{r}_1^m - \beta_{(1,m)}^{in} M\bar{r}_0^m \\ &= \left(1 + \beta_{(1,m)}^{in}\right) M^2\bar{r}_0^m - \beta_{(1,m)}^{in} M\bar{r}_0^m \\ &= \left[\left(1 + \beta_{(1,m)}^{in}\right) M^2 - \beta_{(1,m)}^{in} M\right] \bar{r}_0^m. \end{aligned} \tag{20}$$

At the end of the inner loop, we set

$$x_{m+1} = \bar{x}_2^m,$$

therefore, we finally obtain

$$\begin{aligned} r_{m+1} = \bar{r}_2^m &= \left[\left(1 + \beta_{(1,m)}^{in}\right) M^2 - \beta_{(1,m)}^{in} M\right] \bar{r}_0^m \\ &= \left[\left(1 + \beta_{(1,m)}^{in}\right) M^2 - \beta_{(1,m)}^{in} M\right] \\ &\quad \times \left[\left(1 + \sum_{i=1}^{m} \beta_{(i,m)}^{out}\right) Mr_m - \sum_{i=1}^{m} \beta_{(i,m)}^{out} Mr_{m-i}\right]. \end{aligned} \tag{21}$$

Then, we go back to the outer loop again, for iteration $k = m + 1$, we apply $AA(m)$ to $\{x_1, x_2, \cdots, x_m, x_{m+1}\}$ to get

$$x_{(m+1)+1/2} = \left(1 + \sum_{i=1}^{m} \beta_{(i,m+1)}^{out}\right) Mx_{m+1} - \sum_{i=1}^{m} \beta_{(1,m+1)}^{out} Mx_{m+1-i} + b \tag{22}$$

and

$$r_{(m+1)+1/2} = \left(1 + \sum_{i=1}^{m} \beta_{(i,m+1)}^{out}\right) Mr_{m+1} - \sum_{i=1}^{m} \beta_{(i,m+1)}^{out} Mr_{m+1-i}. \tag{23}$$

Again, for the inner loop, we set

$$\bar{x}_0^{m+1} = x_{(m+1)+1/2} \tag{24}$$

and thus

$$\begin{aligned}\bar{r}_0^{m+1} &= r_{(m+1)+1/2} \\ &= \left(1+\sum_{i=1}^{m}\beta_{(i,m+1)}^{out}\right) M r_{m+1} - \sum_{i=1}^{m}\beta_{(i,m+1)}^{out} M r_{m+1-i}.\end{aligned} \tag{25}$$

Next, we calculate

$$\bar{x}_1^{m+1} = q(\bar{x}_0^{m+1})$$

and

$$\bar{r}_1^{m+1} = \bar{x}_1^{m+1} - q(\bar{x}_1^{m+1}) = A\bar{x}_1^{m+1} - b = M\bar{r}_0^{m+1}.$$

Apply $AA(1)$ to $\{\bar{x}_0^{m+1}, \bar{x}_1^{m+1}\}$ to get

$$\bar{x}_2^{m+1} = \left(1+\beta_{(1,m+1)}^{in}\right) q(\bar{x}_1^{m+1}) - \beta_{(1,m+1)}^{in} q(\bar{x}_0^{m+1}) \tag{26}$$

and

$$\begin{aligned}\bar{r}_2^{m+1} = \bar{x}_2^{m+1} - q(\bar{x}_2^{m+1}) &= A\bar{x}_2^{m+1} - b \\ &= \left(1+\beta_{(1,m+1)}^{in}\right) M\bar{r}_1^{m+1} - \beta_{(1,m+1)}^{in} M\bar{r}_0^{m+1} \\ &= \left(1+\beta_{(1,m+1)}^{in}\right) M^2\bar{r}_0^{m+1} - \beta_{(1,m+1)}^{in} M\bar{r}_0^{m+1} \\ &= \left[\left(1+\beta_{(1,m+1)}^{in}\right) M^2 - \beta_{(1,m+1)}^{in} M\right]\bar{r}_0^{m+1}\end{aligned} \tag{27}$$

At the end of the inner loop, we set

$$x_{m+2} = \bar{x}_2^{m+1}, \tag{28}$$

therefore, we finally obtain

$$\begin{aligned}r_{m+2} =& \bar{r}_2^{m+1} \\ =& \left[\left(1+\beta_{(1,m+1)}^{in}\right) M^2 - \beta_{(1,m+1)}^{in} M\right]\bar{r}_0^{m+1} \\ =& \left[\left(1+\beta_{(1,m+1)}^{in}\right) M^2 - \beta_{(1,m+1)}^{in} M\right] \\ & \times \left[\left(1+\sum_{i=1}^{m}\beta_{(i,m+1)}^{out}\right) M r_{m+1} - \sum_{i=1}^{m}\beta_{(i,m+1)}^{out} M r_{m+1-i}\right].\end{aligned} \tag{29}$$

Similarly, one can get, for $t > 2$,

$$\begin{aligned} & r_{m+t} \\ &= \left[\left(1+\beta^{in}_{(1,m+t-1)}\right)M^2 - \beta^{in}_{(1,m+t-1)}M\right] \\ &\quad \times \left[\left(1+\sum_{i=1}^{m}\beta^{out}_{(i,m+t-1)}\right)Mr_{m+t-1} - \sum_{i=1}^{m}\beta^{out}_{(i,m+t-1)}Mr_{m+t-1-i}\right]. \end{aligned} \tag{30}$$

With derivations, we have the following lemma.

Proposition 2 *AA(m,AA(1)) iteration with initial guess* $\{x_0, x_1, \cdots, x_m\}$ *is a multi-Krylov method. The residual can be expressed as*

$$r_{k+1} = \sum_{j=0}^{m} p_{(k-m+1,j)}(M)r_j, \quad k \geq m, \tag{31}$$

where the $p_{(k-m+1,j)}(\lambda)$ *are polynomials of degree at most* $3(k-m+1)-2$ *satisfying the following relations:*

$$p_{(1,j)}(\lambda) = -\lambda^2\left[\left(1+\beta^{in}_{(1,m)}\right)\lambda - \beta^{in}_{(1,m)}\right]\beta^{out}_{(m-j,m)}, \quad j = 0, \cdots, m-1; \tag{32}$$

$$p_{(1,m)}(\lambda) = \lambda^2\left[\left(1+\beta^{in}_{(1,m)}\right)\lambda - \beta^{in}_{(1,m)}\right]\left(1+\sum_{i=1}^{m}\beta^{out}_{(i,m)}\right), \tag{33}$$

and for $k-m+1 > 1$, $j = 0, \cdots, m$,

$$\begin{aligned} p_{(k-m+1,j)}(\lambda) =& \lambda^2\left[\left(1+\beta^{in}_{(1,k-m)}\right)\lambda - \beta^{in}_{(1,k-m)}\right] \\ & \times \left[\left(1+\sum_{i=1}^{m}\beta^{out}_{(i,k-m)}\right)p_{(k-m,j)} - \sum_{i=1}^{m}\beta^{out}_{(i,k-m)}p_{(k-m-i,j)}\right], \end{aligned} \tag{34}$$

where for $i = 1, 2, \cdots, m$ *and* $j = 0, 1, \cdots, m$,

$$p_{(1-i,j)}(\lambda) = \begin{cases} 1 & if \quad i = j+1-m, \\ 0 & otherwise. \end{cases} \tag{35}$$

Compared with the results from De Sterck and He's work [9] for $AA(m)$, we obtain the highest possible degree of the residual polynomials for $AA(m, AA(1))$ in Table 3.

Moreover, using Proposition 2, we can easily show that there is a periodic pattern in the $AA(m, AA(1))$ polynomials.

Table 3 Highest order (at most) term of residual polynomials at $3(k-m+1)$-th function evalutations

Methods	Highest order (at most) term in residual polynomials
AA(m)	$M^{3(k-m+1)}$
AA(m,AA(1))	$M^{3(k-m+1)-2}$

Proposition 3 *The residuals generated by $AA(m, AA(1))$ iteration with general initial guess $\{x_0, x_1, \cdots, x_m\}$ applied to fixed-point iteration can be also written as:*

$$r_{s(m+1)+i} = M^{2s} \sum_{j=0}^{m} g_{(s(m+1)+i-m-s,j)}(M) r_j, \quad s \geq 1, \ j = 0, \cdots, m, \tag{36}$$

where the $g_{(s(m+1)+i-m-s,j)}(\lambda)$ are polynomials of degree at most $3[s(m+1)+i-m-s]-4$ satisfying $\lambda^{2s} g_{(s(m+1)+i-m-s,j)}(\lambda) = p_{(s(m+1)+i-m,j)}(\lambda)$ from (31).

Here, Eq. (36) indicates that the power of M in the right hand side increases by 2 for every $m+1$ outer loop iterations. We refer to this property as the $AA(m, AA(1))$ iterations possessing a periodic memory effect. However, for AA(m), the power of M in the right hand side increases by 1 for every $m+1$ iterations [9].

4 Influence of initial guess on convergence speed of AA(1,AA(1))

In this section, we study the influence of initial guesses on the r-linear convergence factor of error for $AA(1, AA(1))$. We first investigate a scaling invariance property of the initial guess x_0 for the $AA(1, AA(1))$ method with $x_1 = q(x_0)$ in the linear case. Then, we study the root-linear asymptotic convergence factor under scaling of the initial condition. Since solving $Ax = b$ is equivalent to solving the homogeneous system $Ay = 0$ with $y = x - A^{-1}b$, thus we consider solving $Ax = 0$ in the following discussion.

Proposition 4 *Consider solving homogeneous system $Ax = 0$. For any nonzero initial guess x_0, let $x_1 = q(x_0)$. Then, the $AA(1, AA(1))$ polynomials in (15) have the following scaling invariance properties: for any nonzero scalar α,*

$$\beta_k^{out}(x_0) = \beta_k^{out}(\alpha x_0), \quad \beta_k^{in}(x_0) = \beta_k^{in}(\alpha x_0) \tag{37}$$

and

$$p_k(\lambda, x_0) = p_k(\lambda, \alpha x_0). \tag{38}$$

Proof We prove this proposition by induction. To begin with, we have

$$r_0 = Ax_0, \quad r_1 = Mr_0 = MAx_0 \tag{39}$$

and

$$\begin{aligned}\beta_1^{out}(x_0) = \frac{-r_1^T(r_1 - r_0)}{(r_1 - r_0)^T(r_1 - r_0)} &= \frac{-(MAx_0)^T(MA - A)x_0}{((MA - A)x_0)^T(MA - A)x_0} \\ &= \frac{-x_0^T(MA)^T(MA - A)x_0}{x_0^T(MA - A)^T(MA - A)x_0}.\end{aligned} \tag{40}$$

Therefore, for $k = 1$ and any nonzero scalar α,

$$\begin{aligned}\beta_1^{out}(\alpha x_0) &= \frac{-\alpha^T x_0^T(MA)^T(MA - A)\alpha x_0}{\alpha^T x_0^T(MA - A)^T(MA - A)\alpha x_0} \\ &= \frac{-x_0^T(MA)^T(MA - A)x_0}{x_0^T(MA - A)^T(MA - A)x_0} \cdot \frac{\alpha^2}{\alpha^2} \\ &= \frac{-x_0^T(MA)^T(MA - A)x_0}{x_0^T(MA - A)^T(MA - A)x_0} \\ &= \beta_1^{out}(x_0).\end{aligned} \tag{41}$$

To show $\beta_1^{in}(\alpha x_0) = \beta_1^{in}(x_0)$, note the fact that

$$\begin{aligned}\beta_1^{in}(x_0) = \frac{-(\bar{r}_1^1)^T(\bar{r}_1^1 - \bar{r}_0^1)}{(\bar{r}_1^1 - \bar{r}_0^1)^T(\bar{r}_1^1 - \bar{r}_0^1)} &= \frac{-(M\bar{r}_0^1)^T(M\bar{r}_0^1 - \bar{r}_0^1)}{(M\bar{r}_0^1 - \bar{r}_0^1)^T(M\bar{r}_0^1 - \bar{r}_0^1)} \\ &= \frac{-(\bar{r}_0^1)^T M^T(M - I)\bar{r}_0^1}{(\bar{r}_0^1)^T(M - I)^T(M - I)\bar{r}_0^1}.\end{aligned} \tag{42}$$

From (4), we have

$$\begin{aligned}\bar{r}_0^1 = \left(1 + \beta_1^{out}\right)Mr_1 - \beta_1^{out}Mr_0 &= \left(1 + \beta_1^{out}\right)M^2 r_0 - \beta_1^{out}Mr_0 \\ &= \left[\left(1 + \beta_1^{out}\right)M^2 - \beta_1^{out}M\right]r_0 \\ &= \left[\left(1 + \beta_1^{out}\right)M^2 - \beta_1^{out}M\right]Ax_0.\end{aligned} \tag{43}$$

Use (42) and (43), we have

$$\beta_1^{in}(x_0) = \frac{-(\bar{r}_0^1)^T M^T(M - I)\bar{r}_0^1}{(\bar{r}_0^1)^T(M - I)^T(M - I)\bar{r}_0^1}. \tag{44}$$

Therefore, using the fact that $\beta_1^{out}(\alpha x_0) = \beta_1^{out}(x_0)$, a direct calculation gives

$$\beta_1^{in}(\alpha x_0) = \beta_1^{in}(x_0). \tag{45}$$

So, we have $\beta_1^{out}(\alpha x_0) = \beta_1^{out}(x_0)$ and $\beta_1^{in}(\alpha x_0) = \beta_1^{in}(x_0)$. Since $p_1(\lambda) = \lambda$, it is obvious that $p_1(\lambda, \alpha x_0) = p_1(\lambda, x_0)$.

Next, assume that for $k \leq n$, $p_k(\lambda, \alpha x_0) = p_k(\lambda, x_0)$. Note that for $k \leq n$, from (14),

$$\begin{aligned} \beta_k^{out}(\alpha x_0) &= \frac{-r_k^T (r_k - r_{k-1})}{(r_k - r_{k-1})^T (r_k - r_{k-1})} \\ &= \beta_k^{out}(x_0). \end{aligned} \tag{46}$$

Similarly, one can also derive that

$$\beta_k^{in}(\alpha x_0) = \beta_k^{in}(x_0). \tag{47}$$

For $k = n + 1$, using (15), (46) and (47) and the induction assumption, we then have

$$\begin{aligned} & p_{n+1}(\lambda, \alpha x_0) \\ =& \lambda^2 \left[\left(1 + \beta_k^{in}(\alpha x_0) \right) \lambda - \beta_k^{in}(\alpha x_0) \right] \\ & \times \left[\left(1 + \beta_k^{out}(\alpha x_0) \right) \cdot p_k(\lambda, \alpha x_0) - \beta_k^{out}(\alpha x_0) \cdot p_{k-1}(\lambda, \alpha x_0) \right] \\ =& \lambda^2 \left[\left(1 + \beta_k^{in}(x_0) \right) \lambda - \beta_k^{in}(x_0) \right] \\ & \times \left[\left(1 + \beta_k^{out}(x_0) \right) \cdot p_k(\lambda, x_0) - \beta_k^{out}(x_0) \cdot p_{k-1}(\lambda, x_0) \right] \\ =& p_{n+1}(\lambda, x_0), \end{aligned} \tag{48}$$

which completes the proof. □

Next, we give a result of the root-linear asymptotic convergence factor under the scaling of the initial guesses.

Proposition 5 *Consider solving a homogeneous system $Ax = 0$. For any nonzero guess x_0, let $x_1 = q(x_0)$. Then, the $AA(1, AA(1))$ residuals in (14) have the following property: for any nonzero scalar α,*

$$\lim_{k \to \infty} \|r_k(\alpha x_0)\|^{\frac{1}{k}} = \lim_{k \to \infty} \|r_k(x_0)\|^{\frac{1}{k}}, \tag{49}$$

where we explicitly indicate the dependence of r_k on the initial guess.

Proof For initial guess αx_0, we have

$$r_0(\alpha x_0) = A\alpha x_0 = \alpha A x_0 = \alpha r_0(x_0). \tag{50}$$

From Proposition 4, we have $p_k(\lambda, \alpha x_0) = p_k(\lambda, x_0)$. Therefore, we have

$$\begin{aligned}\lim_{k\to\infty} \|r_k(\alpha x_0)\|^{\frac{1}{k}} &= \lim_{k\to\infty} \|p_k(M, \alpha x_0) r_0(\alpha x_0)\|^{\frac{1}{k}} \\ &= \lim_{k\to\infty} \|p_k(M, x_0)\alpha r_0(x_0)\|^{\frac{1}{k}} \\ &= \lim_{k\to\infty} (\alpha)^{\frac{1}{k}} \|p_k(M, x_0) r_0(x_0)\|^{\frac{1}{k}} \\ &= \lim_{k\to\infty} \|p_k(M, x_0) r_0(x_0)\|^{\frac{1}{k}} \\ &= \lim_{k\to\infty} \|r_k(x_0)\|^{\frac{1}{k}}.\end{aligned} \tag{51}$$

□

To verify this result, in next section, we also numerically examine the influence of the initial guess on the asymptotic convergence factor of AA(m) and AA(m,AA(1)) for both linear and nonlinear problems, see more discussions in Section 7.

5 Local convergence results for linear problems

Let M be a linear operator with $\|M\| = c < 1$, we consider the following fixed point problem

$$x = q(x) = Mx + b.$$

The residual in this case is

$$f(x) = q(x) - x = b - (I - M)x. \tag{52}$$

Theorem 5.1 *Let $m_k = \min(m, k)$ and assume that $\|M\| = c < 1$ and $m > n$, then the composite Anderson iteration of $AA(m, AA(n))$ as in Algorithm 4*

$$x_{k+1/2} = \sum_{j=0}^{m_k} \beta_j^k q(x_{k-m_k+j}) \qquad (outer\ \ loop), \tag{53}$$

$$x_{k+1} = \sum_{i=0}^{n} \beta_i^{k+1/2} q(x_{k+1/2-n+j}) \quad (inner\ \ loop) \tag{54}$$

converges to $x^* = (I - M)^{-1}b$ and the residuals converge q-linearly to zero.

Proof Using the fact that $\sum_{j=0}^{m_k} \beta_j^k = 1$ and $\sum_{i=0}^{n} \beta_i^{k+1/2} = 1$ and the fact that if $\{\beta_j^k\}_{j=0}^{m_k}$ and $\{\beta_i^{k+1/2}\}_{i=0}^{n}$ are the solutions of the least-squares problem at outer iteration k and inner iteration $k + 1/2$, respectively, then by definiton, we have

$$\left\| \sum_{j=0}^{m_k} \beta_j^k f(x_{k-m_k+j}) \right\| \le \|f(x_k)\| \tag{55}$$

and

$$\left\| \sum_{i=0}^{n} \beta_i^{k+1/2} f(x_{k+1/2-n+i}) \right\| \leq \| f(x_{k+1/2}) \|. \tag{56}$$

Therefore, the new residual is

$$\begin{aligned}
\| f(x_{k+1}) \| &= \| b - (I - M) x_{k+1} \| \\
&= \left\| \sum_{i=0}^{n} \beta_i^{k+1/2} [b - (I - M)(b + M x_{k+1/2-n+i})] \right\| \\
&= \left\| \sum_{i=0}^{n} \beta_i^{k+1/2} M [b - (I - M) x_{k+1/2-n+i}] \right\| \\
&= \left\| M \sum_{i=0}^{n} \beta_i^{k+1/2} f(x_{k+1/2-n+i}) \right\| \\
&\leq c \left\| f(x_{k+1/2}) \right\| \\
&= c \left\| b - (I - M) x_{k+1/2} \right\| \\
&= c \left\| \sum_{j=0}^{m_k} \beta_j^{k} [b - (I - M)(b + M x_{k-m_k+j})] \right\| \\
&= c \left\| \sum_{j=0}^{m_k} \beta_j^{k} M [b - (I - M) x_{k-m_k+j}] \right\| \\
&= c \left\| M \sum_{j=0}^{m_k} \beta_j^{k} f(x_{k-m_k+j}) \right\| \\
&\leq c^2 \| f(x_k) \|,
\end{aligned} \tag{57}$$

where for the second equal sign, we use (54) and the fact that $\sum_{i=0}^{n} \beta_i^{k+1/2} = 1$; for the first inequality sign, we use $\|M\| = c < 1$ and (56). Thus, we have

$$\| f(x_{k+1}) \| \leq c^2 \| f(x_k) \| \quad (c < 1),$$

this prove that the residuals converge q-linearly to zero. □

Moreover, let the error

$$e = x - x^*,$$

from (52), we have

$$f(x^*) = q(x^*) - x^* = b - (I - M) x^* = 0,$$

so

$$b = (I - M) x^*.$$

Therefore,

$$\begin{aligned} f(x) &= q(x) - x \\ &= b - (I - M)x \\ &= (I - M)x^* - (I - M)x \\ &= -(I - M)e. \end{aligned} \tag{58}$$

Thus, the q-linear convergence of residuals implies that

$$(1 - c)\|e_{k+1}\| \leq \|f(x_{k+1})\| \leq c^{2(k+1)}\|f(x_0)\| \leq c^{2(k+1)}(1 + c)\|e_0\|.$$

So, we have the error

$$\|e_{k+1}\| \leq \left(\frac{1 + c}{1 - c}\right) c^{2(k+1)}\|e_0\|,$$

where $c = \|M\| < 1$.

6 Residual bounds estimation for nonlinear problems

In this section, we investigate the residual bounds of these composite Anderson acceleration methods. Here, we provide the residual bounds for $AA(m, AA(1))$ with inner loop iteration $iterN = 1$. Similarly, one can derive the rate of convergence to other non-stationary composite AA methods $AA(m, AA(n))$. The main assumptions and ideas of the proof are adopted from papers [13, 24, 31] with necessary modifications. We numerically study these methods in Section 7.

We first summarize the convergence result for Anderson acceleration with damping factors as in Algorithm 2. Then, we provide the residual bounds estimations for our proposed composite AA methods.

Theorem 6.1 *Assume that*

- $q : R^n \to R^n$ *has a fixed point* $x^* \in R^n$ *such that* $q(x^*) = x^*$.
- q *is uniformly Lipschitz continuously differentiable in the ball* $B(\xi) = \{x | \|x - x^*\|_2 \leq \xi\}$.
- *There exists* $\kappa \in (0, 1)$ *such that* $\|q(y) - q(x)\|_2 \leq \kappa\|y - x\|_2$, $\forall x, y \in R^n$.

Then, we have the following residual bound for $AA(m)$ *given in Algorithm 2:*

$$\begin{aligned} \|f(x_{k+1})\|_2 \leq & \theta_{k+1}\left[(1 - \gamma_k) + \kappa\gamma_k\right]\|f(x_k)\|_2 \\ & + O(\|f(x_k)\|\sum_{i=0}^{m}\|f(x_{k-m+i})\|), \end{aligned} \tag{59}$$

where γ_k *are damping factors with* $\gamma_k \in (0, 1]$ *and the average gain*

$$\theta_{k+1} = \frac{\|\sum_{i=0}^{m}\beta_i^k f(x_{k-m+i})\|_2}{\|f(x_k)\|_2}$$

with

$$x_k^{\beta} = \sum_{i=0}^{m_k} \beta_i^{(k)} x_{k-m_k+i}, \quad \tilde{x}_k^{\beta} = \sum_{i=0}^{m_k} \beta_i^{(k)} q(x_{k-m_k+i}).$$

Proof The proof of this theorem can be found in this paper [24] for general damping γ_k. The key ideas of the proof are relating the difference of consecutive iterates to residuals based on performing the inner-optimization and explicitly defining the gain in the optimization stage to be the ratio of improvement over a step of the unaccelerated fixed-point iteration. □

Now, we give the residual bounds estimations for the composite Anderson acceleration methods $AA(m, AA(1))$ with inner loop iteration $iterN = 1$, which means the window size for the outer loop is m and the window size for the inner loop iteration is 1.

Theorem 6.2 *Assume that $q : R^n \to R^n$ has a fixed point $x^* \in R^n$ such that $q(x^*) = x^*$ and satisfies all assumptions in* Theorem 6.1. *Then, we have the following convergence rate for $AA(m, AA(1))$ as in Algorithm 4 with $iterN = 1$ inner loop iterations:*

$$\begin{aligned} \|f(x_{k+1})\|_2 \leq & \bar{\theta}_1 \theta_{k+1} \kappa \left[(1-\gamma_k) + \kappa\gamma_k\right] \|f(x_k)\|_2 \\ & + O(\|f(x_k)\| \sum_{i=0}^{m} \|f(x_{k-m+i})\|), \end{aligned} \tag{60}$$

where

$$\theta_{k+1} = \frac{\|\sum_{i=0}^{m} \beta_i^k f(x_{k-m+i})\|_2}{\|f(x_k)\|_2}, \quad \bar{\theta}_1 = \frac{\|\bar{\beta}_0^1 f(\bar{x}_0) + \bar{\beta}_1^1 f(\bar{x}_1)\|_2}{\|f(\bar{x}_1)\|_2}$$

with

$$\bar{x}_0 = x_{k+1/2}, \quad \bar{x}_1 = q(\bar{x}_0).$$

Proof For the outer loop, according to the results in Theorem 6.1 with any damping factor $\gamma_k \in (0, 1]$, we have

$$\begin{aligned} \|f(x_{k+1/2})\|_2 \leq & \theta_{k+1} \kappa \left[(1-\gamma_k) + \kappa\gamma_k\right] \|f(x_k)\|_2 \\ & + O(\|f(x_k)\| \sum_{i=0}^{m} \|f(x_{k-m+i})\|), \end{aligned} \tag{61}$$

where

$$\theta_{k+1} = \frac{\|\sum_{i=0}^{m} \beta_i^k f(x_{k-m+i})\|_2}{\|f(x_k)\|_2}.$$

As β_i^k is the solution to the optimization problem in Algorithm 2 and the fact that $\beta_k^k = 1$, $\beta_j^k = 0$, $j \neq k$, is in the feasible set for the optimization problem, we

immediately have

$$0 \leq \theta_{k+1} \leq 1.$$

For the inner loop with $iterN = 1$, we have the initial guess $\bar{x}_0 = x_{k+1/2}$, then $\bar{x}_1 = q(\bar{x}_0)$ and

$$f(\bar{x}_0) = q(\bar{x}_0) - \bar{x}_0, \quad f(\bar{x}_1) = q(\bar{x}_1) - \bar{x}_1.$$

Let $\bar{\beta}_0$ and $\bar{\beta}_1$ be the solution to the inner loop optimization problem, then applying Theorem 6.1 with $m = 1$ without damping, we have

$$\|f(\bar{x}_2)\|_2 \leq \bar{\theta}_1 \kappa \|f(\bar{x}_1)\| + O(\|f(\bar{x}_1)\|_2^2) + O(\|f(\bar{x}_0)\|_2^2) \tag{62}$$

with

$$\bar{\theta}_1 = \frac{\|\bar{\beta}_0^1 f(\bar{x}_0) + \bar{\beta}_1^1 f(\bar{x}_1)\|_2}{\|f(\bar{x}_1)\|_2},$$

where $\bar{\beta}_0^1$ and $\bar{\beta}_1^1$ solves the optimization problem of $AA(1)$ in the inner loop iteration. Similarly, since $\bar{\beta}_0^1 = 0$ and $\bar{\beta}_1^1 = 1$, is in the feasible set for the related optimization problem, we get

$$0 \leq \bar{\theta}_1 \leq 1.$$

Using (61) and the fact that the inner loop use $\bar{x}_0 = x_{k+1/2}$ as an initial guess, we have $\bar{x}_1 = q(\bar{x}_0) = q(x_{k+1/2})$. Therefore,

$$\|f(\bar{x}_1)\|_2 \leq \theta_{k+1} \kappa \|f(x_k)\|_2 + O(\|f(x_k)\| \sum_{i=0}^{m} \|f(x_{k-m+i})\|). \tag{63}$$

Since $iterN = 1$, so after finishing the inner loop iteration, we will set $x_{k+1} = \bar{x}_2$. Thus, from (62) and (63), we finally obtain

$$\begin{aligned} \|f(x_{k+1})\|_2 = \|f(\bar{x}_2)\|_2 &\leq \bar{\theta}_1 \theta_{k+1} \kappa \left[(1 - \gamma_k) + \kappa \gamma_k\right] \|f(x_k)\|_2 \\ &+ O(\|f(x_k)\| \sum_{i=0}^{m} \|f(x_{k-m+i})\|), \end{aligned} \tag{64}$$

which completes the proof of this theorem. □

7 Numerical experiments

In this section, we seek to illustrate how the theoretical results of this paper help in understanding the improvement in asymptotic convergence speed resulting from the acceleration of fixed-point iteration by AA(m) and AA(m,AA(n)) with inner loop iteration $iterN = n$.

We first introduce some notions and terminologies of convergence [9, 10].

Definition 7.1 (r-linear convergence) Let $\{x_k\}$ be any sequence that converges to x^*. Set

$$\rho_{\{x_k\}} = \lim_{k\to\infty} \sup \|x_k - x^*\|^{\frac{1}{k}}.$$

Then, we say$\{x_k\}$ converges r-linearly with r-linear convergence factor $\rho_{\{x_k\}}$ if $\rho_{\{x_k\}} \in (0, 1)$ and r-superlinearly if $\rho_{\{x_k\}} = 0$. The "r-" prefix stands for "root".

We define the root-averaged error sequence of x_k converging to x^* as

$$\sigma_k = \|x_k - x^*\|^{\frac{1}{k}}. \tag{65}$$

Definition 7.2 (r-linear convergence of a fixed-point iteration) Consider the fixed-point iteration $x_{k+1} = q(x_k)$. Define the set of iteration sequences that converge to a given fixed point x^* as

$$C(q, x^*) = \left\{ \{x_k\}_{k=0}^{\infty} | x_{k+1} = q(x_k) \ for\ k = 0, 1, 2, \cdots, \ and \lim_{k\to\infty} x_k = x^* \right\}$$

and the worst-case r-linear convergence factor over $C(q, x^*)$ is defined as

$$\rho_{q,x^*} = \sup \left\{ \rho_{\{x_k\}} | \{x_k\} \in C(q, x^*) \right\}. \tag{66}$$

We say that the fixed-point iteration converges r-linearly to x^* with r-linearly convergence factor ρ_{q,x^*} if $\rho_{q,x^*} \in (0, 1)$.

If the iteration function $q(x)$ in fixed-point iteration is differentiable at x^*, then the following classical theorem [22] shows that the worst-case r-linear convergence factor ρ_{q,x^*} is determined by the spectral radius of the Jacobian $q'(x)$ evaluated at x^*:

Theorem 7.1 [Ostrowski Theorem [22]] *Suppose that $q : D \subset \mathbb{R}^n \to \mathbb{R}^n$ has a fixed point x^* that is an interior point of D and it is differentiable at x^*. If the spectral radius of $q'(x)$ satidfied $0 < \rho(q'(x^*)) < 1$, then the fixed-point iteration method converges r-linearly with $\rho_{q,x^*} = \rho(q'(x^*))$.*

In order to compare the behaviors of non-stationary AA(m,AA(1)) with AA(m), the numerical tests we consider in this section were mainly from De Sterck and He, [9, 10] where they identify and shed interesting light on AA(m) convergence patterns for those problems.

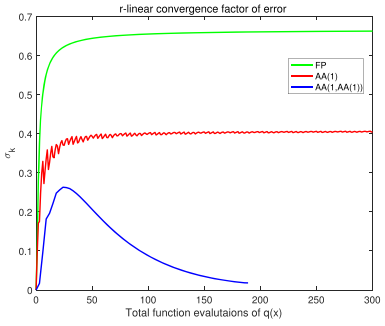
7.1 AA(m,AA(1)) for linear problems

Example 7.1 [A 2×2 linear system with $\rho(q'(x)) < 1$] [9, 10] Apply composite Anderson acceleration AA(1,AA(1)) with inner loop iteration $iterN = 1$ to accelerate the following linear fixed-point iteration function $q(x) = Mx$, i.e., $x_{k+1} = Mx_k$, where M is

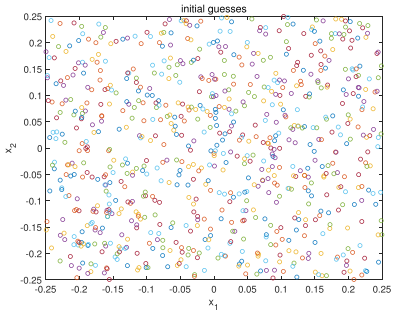
$$M_{2\times 2} = \begin{pmatrix} 2/3 & 1/4 \\ 0 & 1/3 \end{pmatrix}$$

with fixed point $x^* = (0, 0)^T$. It is easy to see that the eigenvalues of M are $\lambda_1 = 2/3$ and $\lambda_2 = 1/3$.

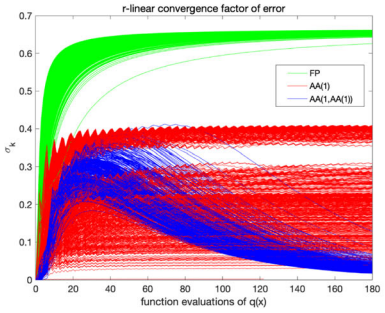
The results are shown in Figs. 1 and 2. Figure 1a shows convergence curves for the root-averaged error σ_k (defined in (65)) of the Fixed-point iteration, AA(1) acceleration and AA(1,AA(1)) acceleration, for initial condition $x_0 = (0.2, 0.1)^T$. Since AA(1,AA(1)) with inner loop iteration $iterN = 1$ will do two more function evaluations of $q(x)$, in order to compare the performance with AA(1), we use function evaluations of q(x) instead of iterations throughout this paper. Monte Carlo results with a large number of random initial guesses (see Fig. 1b) are shown in Fig. 1c. Firstly, for fixed-point iteration (FP), it is easy to see from Theorem 7.1 that the σ_k values of FP iterations must converge to $\lim_{k\to\infty} \sigma_k = \rho_{q,x^*} = \rho(q'(x)) = \rho(M) = \lambda_1 = 2/3$ for all initial conditions except when x_0 lies in the eigenvector direction of λ_2. Our numerical results in Fig. 1c are consistent with this. Secondly, the r-linear convergence factors $\rho_{\{x_k\}}$ of $AA(1)$ are smaller than $\lambda_1 = 2/3$ but strongly depend on the initial guess. De Sterck and He [9] for composite Anderson acceleration $AA(1, AA(1))$, we calculate σ_k once every three function evaluations. Although σ_k also depends on initial guess, those σ_k values are relatively smaller than that of AA(1), which indicates



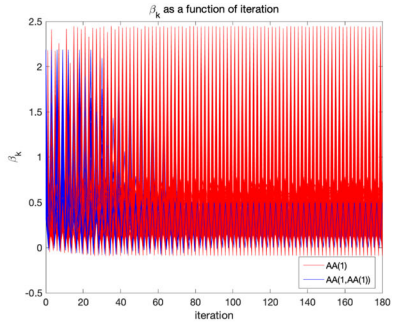
(a) $x_0 = [0.2, 0.1]^T$, convergence factor



(b) Random initial guesses

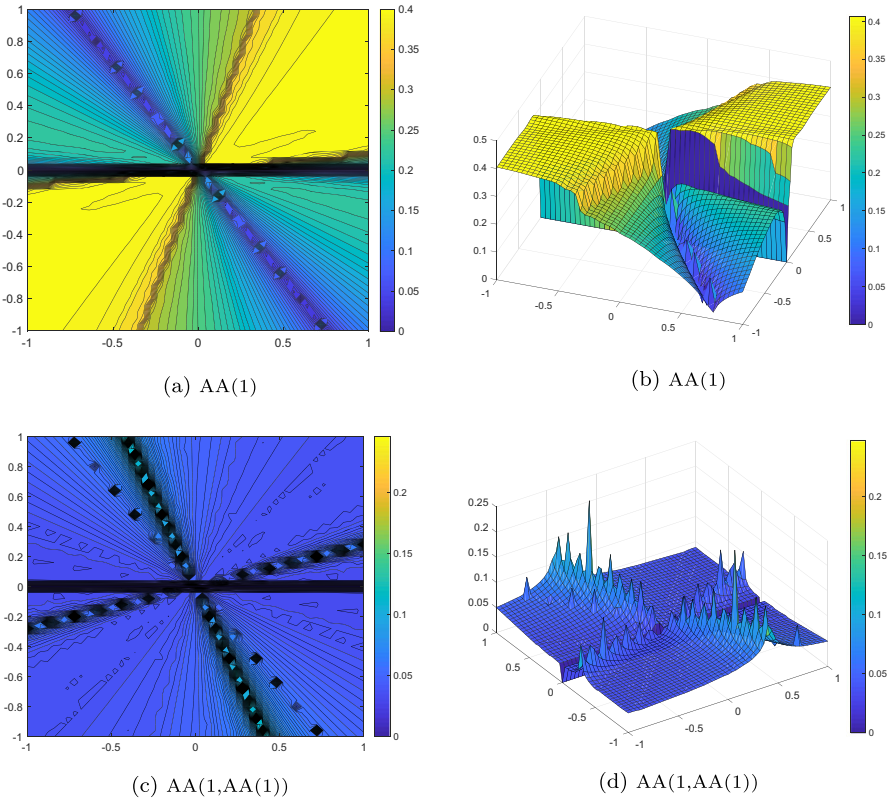


(c) Convergence factor



(d) Random initial guesses, β_k values

Fig. 1 Example 7.1



(a) AA(1) (b) AA(1)

(c) AA(1,AA(1)) (d) AA(1,AA(1))

Fig. 2 Asymptotic convergence factor as a function of initial condition for Example 7.1

a faster convergence rate. We further study this property in the next figure. Furthermore, compared with AA(1), Fig. 1d shows that the outer loop β_k values used in $AA(1, AA(1))$ are also oscillate but with smaller amplitude.

In Fig. 2, we show the convergence factors of AA(1) and AA(1,AA(1)) applied to the linear Example 7.1 for different initial guesses x_0 on a regular grid with 50 by 50 points and $x_1 = q(x_0)$. Figure 2a and b are the results for AA(1), also see De Sterck and He [9]. Compared with AA(1), similar patterns are observed in Fig. 2c and d. However, the convergence factors for AA(1,AA(1)) are much smaller than that of $AA(1)$, which is consistent with results in Fig. 1c. Besides, for this linear problem, a clear radial invariance pattern in Fig. 2c also validates our Propositions 4 and 5

7.2 AA(m,AA(n)) for non-linear problems

Example 7.2 [**A** 2×2 **nonlinear system with** $\rho(q'(x)) < 1$] [9, 10] Apply composite AA to solve the following nonlinear system:

$$\begin{aligned} x_1^2 - x_2 &= 0, \\ x_1 + (x_1 - 1)^2 + x_2^2 &= 1, \end{aligned}$$

with solution $x^* = (0, 0)^T$. Let $x = (x_1, x_2)^T$ and define the FP iteration function

$$q(x) = \begin{bmatrix} \frac{1}{2}(x_1 + x_1^2 + x_2^2) \\ \frac{1}{2}(x_1^2 + x_2) \end{bmatrix}.$$

So the fixed-point for $x = q(x)$, i.e.,

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(x_1 + x_1^2 + x_2^2) \\ \frac{1}{2}(x_1^2 + x_2) \end{bmatrix}$$

is also the solution of the nonlinear equations. The Jacobian matrix of $q(x)$ is

$$q'(x) = \begin{bmatrix} x_1 + \frac{1}{2} & x_2 \\ x_1 & \frac{1}{2} \end{bmatrix}.$$

We have

$$q'(x^*) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \text{ and } \rho(q'(x^*)) = \frac{1}{2} < 1.$$

Figure 3a shows convergence factors for the root-averaged error σ_k (defined in (65)) of the FP iteration, AA(1) and AA(1,AA(1)) with initial condition $x_0 = (0.2, 0.1)^T$. This result is consistent with the convergence rate result in Fig. 3b, where $AA(1, AA(1))$ converges very fast and it seems to converge superlinearly instead of linearly. Moreover, Monte Carlo results with a large number of random initial guesses (see Fig. 3c) for Example 7.2 are shown in Fig. 3d. Again, since $\rho(q'(x^*)) = 1/2 < 1$, the σ_k of FP iteration converges to $1/2$ for all initial guesses. The σ_k of AA(1) are smaller than 1/2 but strongly depend on initial guesses. Like in linear cases, $AA(1, AA(1))$ has σ_k values that are very close to zero.

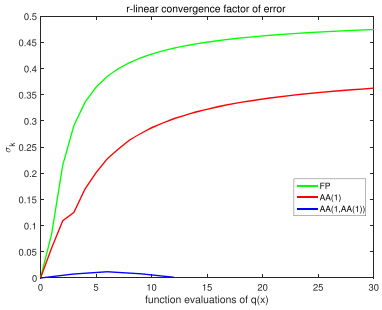
In Fig. 4, we show the convergence factor of AA(1,AA(1)) with $x_1 = q(x_0)$ applied to the nonlinear Example 7.2 for various initial guesses, where we take x_0 on a uniform grid with 101 by 101 points. Since this is a nonlinear problem, we do not have the scaling invariance property. However, for AA(1), there are also preferred directions with fast convergence for the initial condition near the solution $x^* = (0, 0)$, as in the linear case. Again, for composite Anderson acceleration $AA(1, AA(1))$, the convergence factors σ_k are smaller than that of $AA(1)$.

Example 7.3 [Trigonometric functions, k=10,50,100] [26] Apply composite AA to solve the following larger nonlinear systems. Let

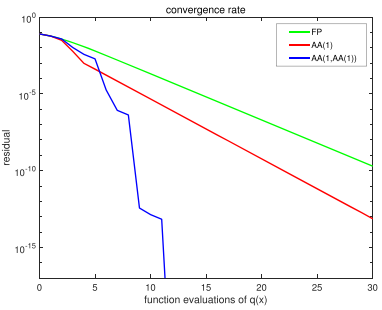
$$f_i(x) = k - \sum_{j=1}^{n} \cos(x_j) + i[1 - \cos(x_i)] - \sin(x_i), \ i = 1, 2, \cdots, k,$$

we create a manufacturing fixed-point $x = [\frac{\pi}{4}, \cdots, \frac{\pi}{4}]^T$ of the following system

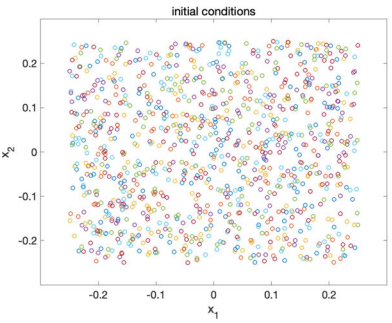
$$n - \sum_{j=1}^{k} \cos(x_j) + i[1 - \cos(x_i)] - \sin(x_i) = f_i(\frac{\pi}{4}), \ i = 1, 2, \cdots, k.$$



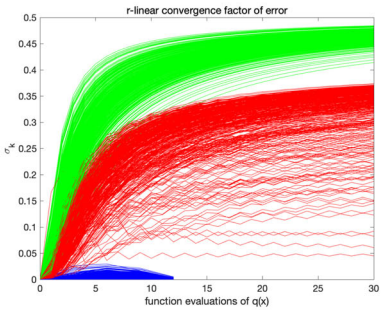
(a) $x_0 = [0.2, 0.1]^T$, convergence factor



(b) $x_0 = [0.2, 0.1]^T$, convergence rate



(c) Random initial guesses



(d) Convergence factor

Fig. 3 Example 7.2

Next, we define the FP iteration function

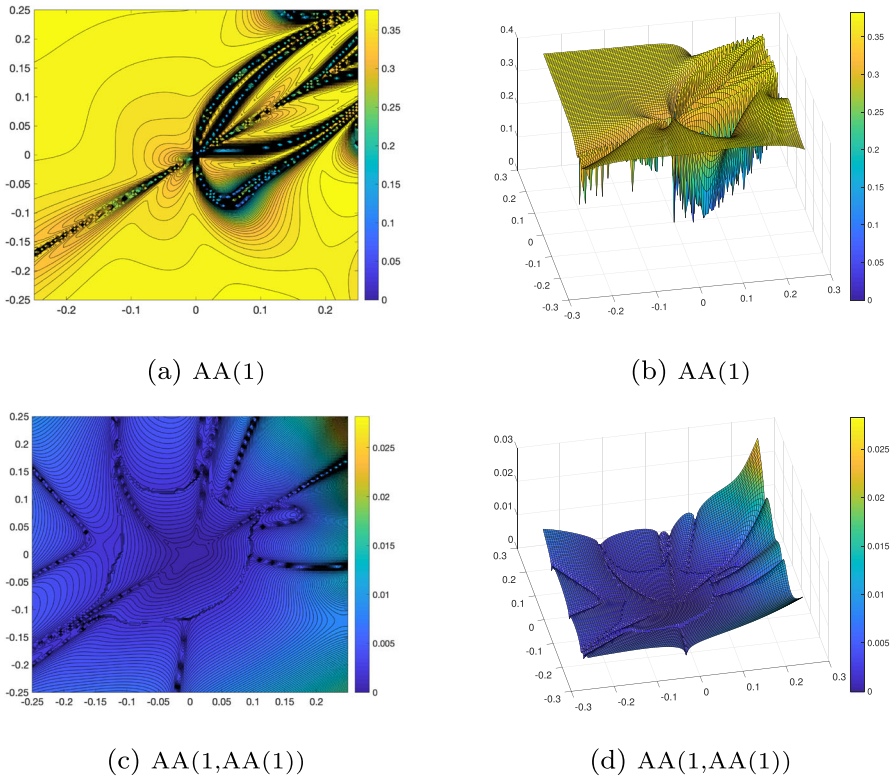
$$q_i(x) = x - \frac{1}{k}\left(f_i(x) - f_i(\frac{\pi}{4})\right),$$

so the fixed point of $x = q(x)$ is also the solution of the above nonlinear equations.

In this numerical test, we adopt an example from paper [26] such that a manufacturing solution is created. The initial guesses $x_0 \in \mathbb{R}^n$ used here are random numbers close to x^* such that $x_0 \in (\pi/4-0.05, \pi/4+0.05) \times \cdots \times (\pi/4-0.05, \pi/4+0.05)$. The results are shown in Fig. 5. Larger window sizes are used in Anderson acceleration when the dimension n increases. From Fig. 5, we observe that the composite AA converges faster than stationary AA for all cases $k = 10, 50, 100$.

7.3 Steady-state incompressible Navier–Stokes equation

The Navier–Stokes system is the basis for computational modeling of the flow of an incompressible Newtonian fluid. In this experiment, we focus on solving the steady incompressible Navier–Stokes equations. Let $\Omega \subset \mathbb{R}^2$ denote the region that enclosed



(a) AA(1) (b) AA(1)

(c) AA(1,AA(1)) (d) AA(1,AA(1))

Fig. 4 Asymptotic convergence factor as a function of initial condition for Example 7.2

the fluid. The steady-state NSE are given in a domain Ω by

$$-\nu\Delta u + (u\cdot\nabla)u + \nabla p = f \tag{67}$$

$$\nabla\cdot u = 0 \tag{68}$$

with appropriate boundary conditions. Here ν is the kinematic viscosity, f is a forcing, u and p represent velocity and pressure. We will seek the velocity and pressure (u, p) of the fluid motion in the spaces (X, Q) given by:

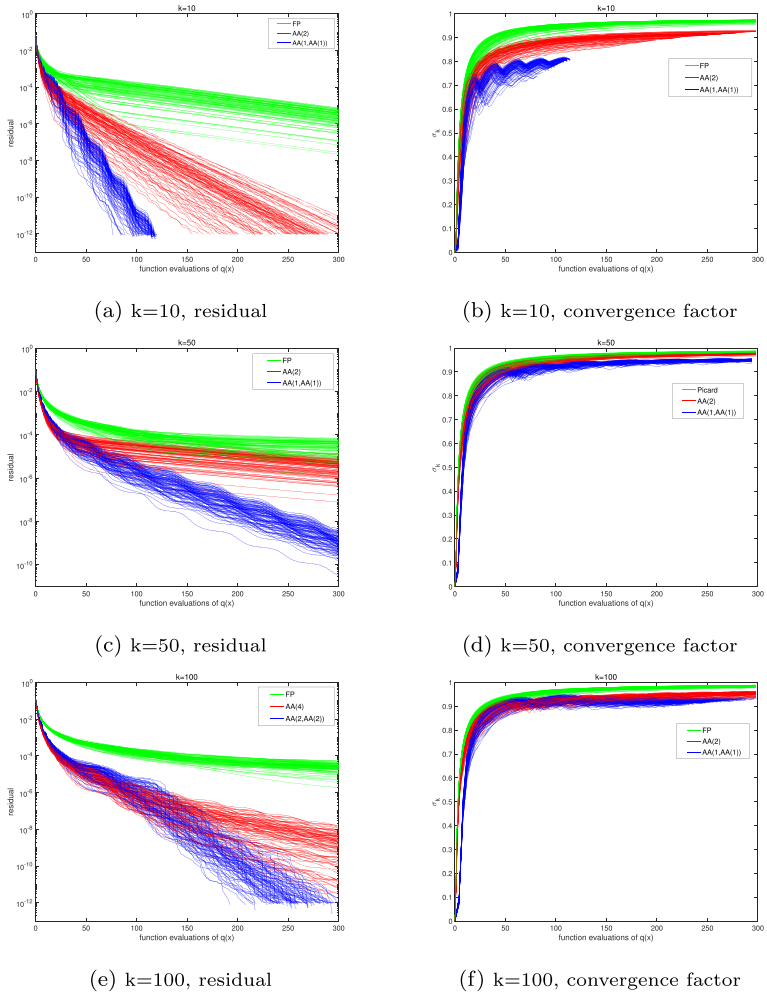
$$X := \left\{ v : \Omega \to \mathbb{R}^2 : v \in L^2(\Omega), \nabla v \in L^2(\Omega) \; and \; v = 0 \; on \; \partial\Omega \right\},$$

$$Q := \left\{ \omega : \Omega \to \mathbb{R}^2 : v \in L^2(\Omega) \; and \; \int_\Omega \omega = 0 \right\}.$$

To find the weak form, we multiply (67)-(68) by $v \in X$ and $\omega \in Q$ and integrating over Ω, so the solution (u, p))satisfies

$$(\nu\nabla u, \nabla v) + (u\cdot\nabla u, v) - (p, \nabla\cdot v) = (f, v), \tag{69}$$

$$(\omega, \nabla\cdot u) = 0. \tag{70}$$



(a) k=10, residual (b) k=10, convergence factor

(c) k=50, residual (d) k=50, convergence factor

(e) k=100, residual (f) k=100, convergence factor

Fig. 5 Example 7.3

for all $v \in X$ and $\omega \in Q$. To numerically solve this problem, we discretize the domain with Taylor-Hood finite elements $(X_h, Q_h) = ((P_2)^2, P_1)$ and define the trilinear form b^* by

$$b^*(u, v, w) := ((u \cdot \nabla)v, w) + \frac{1}{2}((\nabla \cdot u)v, w),$$

then the discrete steady incompressible Navier–Stokes problem (with skew-symmetrized) reads as follows: find $(u, p) \in (X_h, Q_h)$ satisfying for all $(v, \omega) \in (X_h, Q_h)$,

$$b^*(u, u, v) + \nu(\nabla u, \nabla v) - (p, \nabla \cdot v) = (f, v) \tag{71}$$

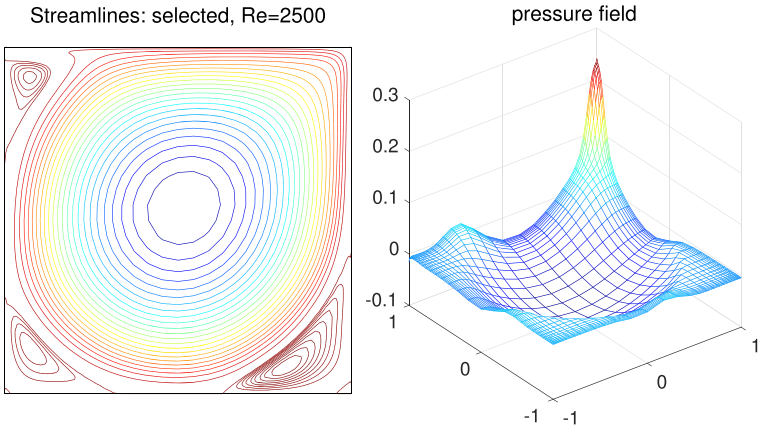
$$(\nabla \cdot u, \omega) = 0 \tag{72}$$

Finally, we get the Picard iterations for the steady NSE which is given in Algorithm 5.

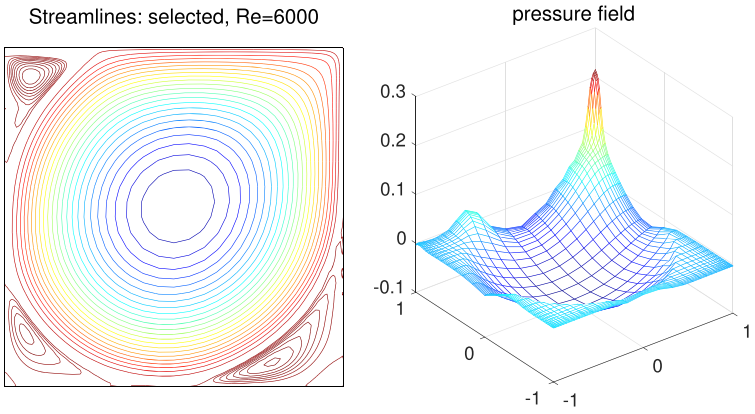
For sufficiently small data, the steady NSE and these iterations are well-posed [18]. Hence, we can consider the Picard iterations as fixed-point iterations $u_{k+1} = q(u_k)$, where g is a solution operator of (73)-(74) for the Picard iterations. In this way, we can apply both the Anderson Acceleration method and the composite Anderson Acceleration method to accelerate the Picard iterations.

Example 7.4 2D lid-driven cavity The 2D lid-driven cavity uses a domain $\Omega = (-1, 1)^2$ with no-slip boundary conditions on the sides and bottom and a "moving lid" on the top which is implemented by enforcing the Dirichlet boundary condition:

$$\{y = 1; -1 \leq x \leq 1 | u_x = 1 - x^4\} \ a \ regularized \ cavity.$$



(a)



(b)

Fig. 6 Lid-driven cavity: **a** $Re = 2500$, solution; **b** $Re = 6000$, solution

There is no forcing and the kinematic viscosity ν is set to be $\nu = \frac{1}{Re}$. We discretize with $(P2 - P1)$ Taylor-Hood elements on a 64×64 non-uniform mesh and use the corresponding discrete Stokes solution as initial guess.

We test it with Reynolds numbers of $Re = 2500$ and $Re = 6000$ as in the paper of Evans et al. [13], respectively. In this example, we use inner loop window size equals two. The results are shown in Figs. 6 and 7. In Fig. 6, we plot the streamlines of the velocity and the pressure field for different Reynolds numbers. The convergence results are shown in Fig. 7. Firstly, in both cases, we see that the Newton method does not converge. We also observe similar results that the composite AA methods

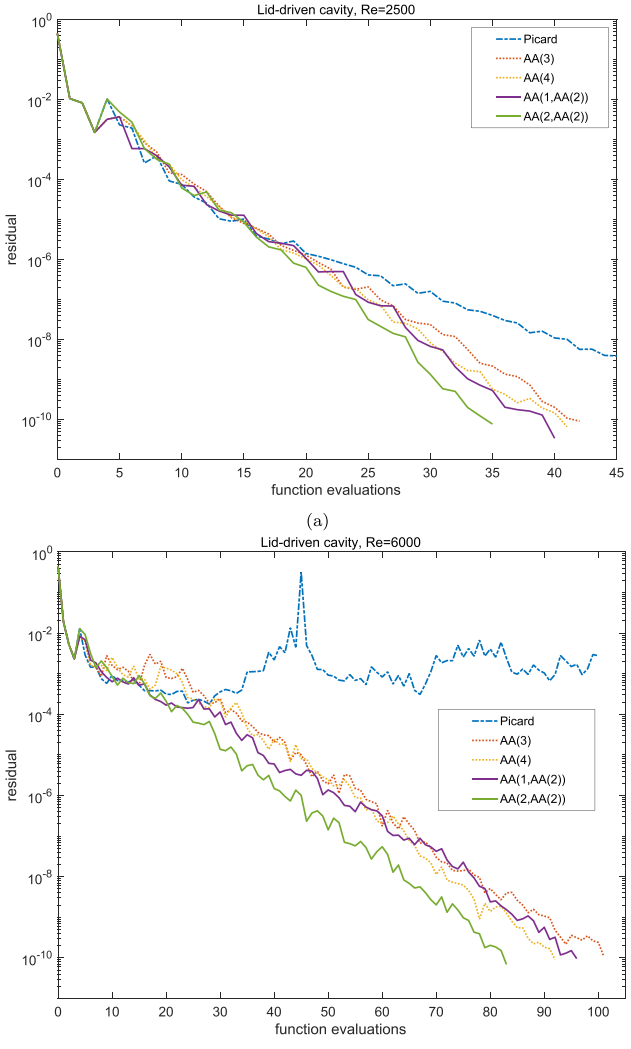


Fig. 7 Lid-driven cavity convergence results: **a** $Re = 2500$; **b** $Re = 6000$

Algorithm 5 Picard iteration for steady NSE.

Choose: $u_0 \in X_h$
Find $(u_k, p_k) \in (X_h, Q_h)$ satisfying for all $(v, \omega) \in (X_h, Q_h)$,

$$b^*(u_{k-1}, u_k, v) + \nu(\nabla u_k, \nabla v) - (p_k, \nabla \cdot v) = (f, v) \tag{73}$$
$$(\nabla \cdot u_k, \omega) = 0 \tag{74}$$

$AA(1, AA(2))$ performs better than stationary AA method $AA(3)$ and $AA(2, AA(2))$ performs better than $AA(4)$.

8 Conclusions

In the present work, we study the polynomial residual update formulas for non-stationary AA(m,AA(1)) and find that AA(m,AA(1)) with general initial guesses is a multi-Krylov method and it possesses a memory effect. For the linear problem, AA(m,AA(1)) potentially has higher order (at most) degree of polynomials and a stronger memory effect than that of AA(m) at k-th iteration. Moreover, we further study the influence of initial guess on the asymptotic convergence factor of $AA(1, AA(1))$. We show a scaling invariance property of the initial guess x_0 for AA(1,AA(1)) methods in the linear case. Lastly, we numerically examine the influence of the initial guess on the asymptotic convergence factor of AA(n) and AA(m,AA(n)) for both linear and nonlinear problems. In general, AA(m,AA(n)) performs relatively better than AA(m+n), especially in solving non-linear problems.

Funding This work was partially supported by the National Natural Science Foundation of China [grant number 12001287]; the Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology [grant number 2019r106].

Data Availability Data will be available on request.

References

1. Anderson, D.G.: Iterative procedures for nonlinear integral equations. J. Assoc. Comput. Mach. **12**, 547–560 (1965). https://doi.org/10.1145/321296.321305
2. Anderson, D.G.M.: Comments on Anderson acceleration, mixing and extrapolation. Numer. Algorithms **80**(1), 135–234 (2019). https://doi.org/10.1007/s11075-018-0549-4
3. Bian, W., Chen, X., Kelley, C.T.: Anderson acceleration for a class of nonsmooth fixed-point problems. SIAM J. Sci. Comput. **43**(5), S1–S20 (2021). https://doi.org/10.1137/20M132938X
4. Brown, J., Knepley, M.G., May, D.A., McInnes, L.C., Smith, B.: Composable linear solvers for multiphysics. In: 2012 11th International Symposium on Parallel and Distributed Computing, pp. 55–62. IEEE (2012). https://doi.org/10.1109/ISPDC.2012.16
5. Brune, P.R., Knepley, M.G., Smith, B.F., Tu, X.: Composing scalable nonlinear algebraic solvers. SIAM Rev. **57**(4), 535–565 (2015). https://doi.org/10.1137/130936725
6. Carlson, N.N., Miller, K.: Design and application of a gradient-weighted moving finite element code. I. In one dimension. SIAM J. Sci. Comput. **19**(3), 728–765 (1998). https://doi.org/10.1137/S106482759426955X

7. Chen, K., Vuik, C.: Composite Anderson acceleration method with two window sizes and optimized damping. Int. J. Numer. Methods Eng. (2022). https://doi.org/10.1002/nme.7096
8. Chen, K., Vuik, C.: Non-stationary Anderson acceleration with optimized damping. arXiv:2202.05295 (2022). https://doi.org/10.48550/arXiv.2202.05295
9. De Sterck, H., He, Y.: Anderson acceleration as a Krylov method with application to asymptotic convergence analysis (2021). https://doi.org/10.48550/arXiv.2109.14181
10. De Sterck, H., He, Y.: Linear asymptotic convergence of Anderson acceleration: fixed-point analysis (2021). https://doi.org/10.48550/arXiv.2109.14176
11. De Sterck, H., He, Y.: On the asymptotic linear convergence speed of Anderson acceleration, Nesterov acceleration, and nonlinear GMRES. SIAM J. Sci. Comput. **43**(5), S21–S46 (2021). https://doi.org/10.1137/20M1347139
12. Eirola, T., Nevanlinna, O.: Accelerating with rank-one updates. pp. 511–520 (1989). https://doi.org/10.1016/0024-3795(89)90719-2
13. Evans, C., Pollock, S., Rebholz, L.G., Xiao, M.: A proof that Anderson acceleration improves the convergence rate in linearly converging fixed-point methods (but not in those converging quadratically). SIAM J. Numer. Anal. **58**(1), 788–810 (2020). https://doi.org/10.1137/19M1245384
14. Eyert, V.: A comparative study on methods for convergence acceleration of iterative vector sequences. J. Comput. Phys. **124**(2), 271–285 (1996). https://doi.org/10.1006/jcph.1996.0059
15. Fang, H.r., Saad, Y.: Two classes of multisecant methods for nonlinear acceleration. Numer. Linear Algebra Appl. **16**(3), 197–221 (2009). https://doi.org/10.1002/nla.617
16. Haelterman, R., Degroote, J., Van Heule, D., Vierendeels, J.: On the similarities between the quasi-Newton inverse least squares method and GMRES. SIAM J. Numer. Anal. **47**(6), 4660–4679 (2010). https://doi.org/10.1137/090750354
17. Kirby, R.C., Mitchell, L.: Solver composition across the PDE/linear algebra barrier. SIAM J Sci Comput **40**(1), C76–C98 (2018). https://doi.org/10.1137/17M1133208
18. Layton, W.: Introduction to the numerical analysis of incompressible viscous flows. SIAM (2008). https://doi.org/10.1137/1.9780898718904
19. Lin, L., Yang, C.: Elliptic preconditioner for accelerating the self-consistent field iteration in Kohn-Sham density functional theory. SIAM J. Sci. Comput. **35**(5), S277–S298 (2013). https://doi.org/10.1137/120880604
20. Miller, K.: Nonlinear Krylov and moving nodes in the method of lines. J. Comput. Appl. Math. **183**(2), 275–287 (2005). https://doi.org/10.1016/j.cam.2004.12.032
21. Oosterlee, C.W., Washio, T.: Krylov subspace acceleration of nonlinear multigrid with application to recirculating flows. pp. 1670–1690 (2000). https://doi.org/10.1137/S1064827598338093
22. Ortega, J.M., Rheinboldt, W.C.: Iterative solution of nonlinear equations in several variables. SIAM (2000). https://doi.org/10.1137/1.9780898719468
23. Peng, Y., Deng, B., Zhang, J., Geng, F., Qin, W., Liu, L.: Anderson acceleration for geometry optimization and physics simulation. ACM Transactions on Graphics (TOG) **37**(4), 1–14 (2018). https://doi.org/10.1145/3197517.3201290
24. Pollock, S., Rebholz, L.G.: Anderson acceleration for contractive and noncontractive operators. IMA J. Numer. Anal. **41**(4), 2841–2872 (2021). https://doi.org/10.1093/imanum/draa095
25. Pollock, S., Rebholz, L.G., Xiao, M.: Anderson-accelerated convergence of Picard iterations for incompressible Navier-Stokes equations. SIAM J. Numer. Anal. **57**(2), 615–637 (2019). https://doi.org/10.1137/18M1206151
26. Pollock, S., Schwartz, H.: Benchmarking results for the Newton-Anderson method. Results in Applied Mathematics **8**, 100095 (2020). https://doi.org/10.1016/j.rinam.2020.100095
27. Pulay, P.: Convergence acceleration of iterative sequences the case of SCF iteration. Chem. Phys. Lett. **73**(2), 393–398 (1980). https://doi.org/10.1016/0009-2614(80)80396-4
28. Pulay, P.: Improved SCF convergence acceleration. J. Comput. Chem. **3**(4), 556–560 (1982). https://doi.org/10.1002/jcc.540030413
29. Shi, W., Song, S., Wu, H., Hsu, Y.C., Wu, C., Huang, G.: Regularized Anderson acceleration for off-policy deep reinforcement learning. arXiv:1909.03245 (2019)
30. Toth, A., Ellis, J.A., Evans, T., Hamilton, S., Kelley, C.T., Pawlowski, R., Slattery, S.: Local improvement results for Anderson acceleration with inaccurate function evaluations. SIAM J. Sci. Comput. **39**(5), S47–S65 (2017). https://doi.org/10.1137/16M1080677
31. Toth, A., Kelley, C.T.: Convergence analysis for Anderson acceleration. SIAM J. Numer. Anal. **53**(2), 805–819 (2015). https://doi.org/10.1137/130919398

32. van der Vorst, H.A., Vuik, C.: GMRESR: a family of nested GMRES methods. Numer. Linear Algebra Appl. **1**(4), 369–386 (1994). https://doi.org/10.1002/nla.1680010404
33. Vuik, C.: Solution of the discretized incompressible Navier-Stokes equations with the GMRES method. Int J Numer Methods Fluids **16**(6), 507–523 (1993). https://doi.org/10.1002/fld.1650160605
34. Walker, H.F., Ni, P.: Anderson acceleration for fixed-point iterations. SIAM J. Numer. Anal. **49**(4), 1715–1735 (2011). https://doi.org/10.1137/10078356X
35. Wang, D., He, Y., De Sterck, H.: On the asymptotic linear convergence speed of Anderson acceleration applied to ADMM. J. Sci. Comput. **88**(2), Paper No. 38, 35 (2021). https://doi.org/10.1007/s10915-021-01548-2
36. Washio, T., Oosterlee, C.W.: Krylov subspace acceleration for nonlinear multigrid schemes. pp. 271–290 (1997). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.147.3799
37. Yang, C., Meza, J.C., Lee, B., Wang, L.W.: KSSOLV—a MATLAB toolbox for solving the Kohn-Sham equations. ACM Trans. Math. Software **36**(2), Art. 10, 35 (2009). https://doi.org/10.1145/1499096.1499099
38. Yang, Y.: Anderson acceleration for seismic inversion. Geophysics **86**(1), R99–R108 (2021). https://doi.org/10.1190/geo2020-0462.1
39. Zhang, J., O'Donoghue, B., Boyd, S.: Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations. SIAM J. Optim. **30**(4), 3170–3197 (2020). https://doi.org/10.1137/18M1232772

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Kewang Chen[1,2] · Cornelis Vuik[3]

1 College of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing 210044, China

2 Jiangsu International Joint Laboratory on System Modeling and Data Analysis, Nanjing University of Information Science and Technology, Nanjing 210044, China

3 Delft Institute of Applied Mathematics, Delft University of Technology, Delft 2628CD, The Netherlands