

*Anderson Acceleration of the Alternating
Projections Method for Computing the Nearest
Correlation Matrix*

Higham, Nicholas J. and Strabić, Nataša

2016

MIMS EPrint: **2015.39**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: <http://eprints.maths.manchester.ac.uk/>

And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

Anderson acceleration of the alternating projections method for computing the nearest correlation matrix

Nicholas J. Higham¹ · Nataša Strabić¹

Received: 4 August 2015 / Accepted: 18 November 2015 / Published online: 21 December 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract In a wide range of applications it is required to compute the nearest correlation matrix in the Frobenius norm to a given symmetric but indefinite matrix. Of the available methods with guaranteed convergence to the unique solution of this problem the easiest to implement, and perhaps the most widely used, is the alternating projections method. However, the rate of convergence of this method is at best linear, and it can require a large number of iterations to converge to within a given tolerance. We show that Anderson acceleration, a technique for accelerating the convergence of fixed-point iterations, can be applied to the alternating projections method and that in practice it brings a significant reduction in both the number of iterations and the computation time. We also show that Anderson acceleration remains effective, and indeed can provide even greater improvements, when it is applied to the variants of the nearest correlation matrix problem in which specified elements are fixed or a lower bound is imposed on the smallest eigenvalue. Alternating projections is a general method for finding a point in the intersection of several sets and ours appears to be the first demonstration that this class of methods can benefit from Anderson acceleration.

This work was supported by European Research Council Advanced Grant MATFUN (267526). The first author was also supported by Engineering and Physical Sciences Research Council grant EP/I01912X/1.

✉ Nicholas J. Higham
nick.higham@manchester.ac.uk
<http://www.maths.manchester.ac.uk/~higham>

Nataša Strabić
natasa.strabic@manchester.ac.uk
<http://www.maths.manchester.ac.uk/~nstrabic>

¹ School of Mathematics, University of Manchester, Manchester, M13 9PL, England

Keywords Nearest correlation matrix · Indefinite matrix · Positive semidefinite matrix · Anderson acceleration · Alternating projections method · Dykstra's correction

Mathematics Subject Classification (2010) 15A57 · 65F30

1 Introduction

In many data analysis applications one must construct a correlation matrix from empirical or experimental data. A correlation matrix is symmetric, has unit diagonal, and is positive semidefinite. Frequently, asynchronous or missing observations lead to the obtained matrix being indefinite. Furthermore, in some applications, such as stress testing and risk aggregation in finance [2, 18, 40] or large-scale resource assessment [6], individual elements of a valid correlation matrix are set or modified by expert judgment, which can again result in an indefinite matrix.

A standard way to correct an invalid correlation matrix, by which we mean a real, symmetric indefinite matrix with unit diagonal, is to replace it by the nearest correlation matrix in the Frobenius norm, that is, by the solution of the problem

$$\min\{ \|A - X\|_F : X \text{ is a correlation matrix} \},$$

where $\|A\|_F^2 = \sum_{i,j} a_{ij}^2$. Other norms could be used, but almost all existing research treats the Frobenius norm (possibly weighted), for which the problem has a unique solution.

The first method for computing the nearest correlation matrix with guaranteed convergence was the alternating projections method proposed by Higham [24]. Although a faster Newton algorithm was subsequently developed by Qi and Sun [39], and practical improvements to it were made by Borsdorf and Higham [8], the alternating projections method remains widely used in applications. Major reasons for its popularity are its ease of coding and the availability of implementations in MATLAB, Python, R, and SAS [25]. Recent examples of applications in which the alternating projections method is being used include probabilistic forecasts of streamflows [51], prediction of electricity peak-demand during the winter in England and Wales [34], analysis of wind farms [19], modeling public health [13], simulation of wireless links in vehicular networks [52], the analysis of carbon dioxide storage resources in the US [6], and a modeling framework that combines different sources of variability in biological systems [45]. As well as being easy to understand and easy to implement, the alternating projections method has the attractive feature that it is easily modified to incorporate additional constraints on the matrix, in particular to fix certain elements or to compute a strictly positive definite solution with a lower bound on the smallest eigenvalue. Both of these problem variants commonly appear in practice.

Since each iteration of the alternating projections method requires a full eigenvalue decomposition and the rate of convergence is at best linear, the method can potentially be very slow. The aim of this work is to reduce the number of iterations required. We attempt to accelerate the alternating projections method by employing

Anderson acceleration [1, 33, sec. 1.1.4], also known as Anderson mixing, which is designed for fixed-point problems. While fixed-point iteration uses only the current, k th, iterate to define the next one, Anderson acceleration uses the additional information from the m_k previous iterations and computes the new iterate as a specific linear combination of these $m_k + 1$ quantities. The selected history length m_k is usually small. A discussion that puts Anderson acceleration in context with other acceleration methods can be found in [50].

In quantum chemistry Anderson acceleration is known as Pulay mixing or direct inversion in the iterative subspace (DIIS) [38] and it has been widely used in electronic structure computations; see [43] and the references therein. Anderson acceleration is related to multiseccant methods (extensions of quasi-Newton methods involving multiple secant conditions); in fact, Eyert [16] proves that it is equivalent to the so-called “bad” Broyden’s method [11, 28], and a similar analysis is done by Fang and Saad [17] and Rohwedder and Schneider [43]. For linear systems, if $m_k = k$ for each k then Anderson acceleration is essentially equivalent to the generalized minimal residual (GMRES) method [44], as shown by Potra and Engler [36], Rohwedder and Schneider [43], and Walker and Ni [50]. For nonlinear problems Rohwedder and Schneider [43] show that Anderson acceleration is locally linearly convergent under certain conditions. Adding to the above convergence analysis is the recent work by Toth and Kelley [46] concerning Anderson acceleration with $m_k = \min(m, k)$, for a fixed m , applied to contractive mappings.

Even though there are no general guarantees of its convergence, Anderson acceleration has a successful record of use in electronic structure computations. Furthermore, it significantly improved the performance of several domain decomposition methods presented in [50] and has proved to be very efficient on various examples in the above references. Hence Anderson acceleration has great potential for enhancing the convergence of the alternating projections method for the nearest correlation matrix.

Recently, López and Raydan [29] have proposed a geometrically-based acceleration scheme for the alternating projections method that builds a new sequence from the original one by taking linear combinations of successive pairs of iterates. The new sequence is tested for convergence and the original iteration remains unchanged. We compare this method with Anderson acceleration in Section 4 (Experiment 9).

The paper is organized as follows. We present the Anderson acceleration scheme in Section 2. In Section 3 we recall the necessary results on the alternating projections method with Dykstra’s correction for computing the nearest correlation matrix and the problem variants in which some elements remain fixed or the smallest eigenvalue of the solution must be above a given threshold, and we explain how to apply Anderson acceleration to these problems. Numerical experiments presented in Section 4 show that Anderson acceleration at least halves the number of iterations required by the alternating projections method for the nearest correlation matrix problem, which results in a significant reduction in computation time for large problems. The experiments also show that even greater improvements can be achieved for the problem variants, which is especially important for the fixed elements constraint since in this case there is no available Newton method. Concluding remarks are given in Section 5.

2 Anderson acceleration for fixed-point iteration

A basic method for the solution of the fixed-point problem $g(x) = x$ for $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is fixed-point iteration, also known as the (nonlinear) Richardson iteration, Picard iteration, or the method of successive substitution. It has the form

$$x_{k+1} = g(x_k), \quad k \geq 1, \quad x_0 \in \mathbb{R}^n \text{ given.} \tag{1}$$

To guarantee convergence of (1) assumptions must be made on the function g and the starting vector x_0 , and in general convergence is at a linear rate [26, Chap. 4.2]. A method that attempts to encourage or accelerate convergence is Anderson acceleration, which redefines x_{k+1} to make use of the information from the m_k previous steps. We first briefly outline the original method derived by Anderson [1].

Algorithm 1 (Original Anderson acceleration) Given $x_0 \in \mathbb{R}^n$ and an integer $m \geq 1$ this algorithm produces a sequence x_k of iterates intended to converge to a fixed point of the function $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$

- 1 $x_1 = g(x_0)$
 - 2 for $k = 1, 2, \dots$ until convergence
 - 3 $m_k = \min(m, k)$
 - 4 Determine $\theta^{(k)} = (\theta_1^{(k)}, \dots, \theta_{m_k}^{(k)})^T \in \mathbb{R}^{m_k}$ that minimizes $\|u_k - v_k\|_2^2$,
 where $u_k = x_k + \sum_{j=1}^{m_k} \theta_j (x_{k-j} - x_k)$, $v_k = g(x_k) + \sum_{j=1}^{m_k} \theta_j (g(x_{k-j}) - g(x_k))$.
 - 5 Set $x_{k+1} = v_k$ using the parameters from $\theta^{(k)}$.
 - 6 end
-

In [1] the final step is $x_{k+1} = u_k + \beta_k(v_k - u_k)$, where u_k and v_k are defined from the computed $\theta^{(k)}$, and $\beta_k > 0$ is empirically determined. The usual choice in the literature is $\beta_k \equiv 1$, which we use here. We have also taken the history length parameter m_k to be fixed, at m , once the first m iterations have been taken.

We can give some insight into Algorithm 1 by writing

$$u_k = \left(1 - \sum_{j=1}^{m_k} \theta_j^{(k)} \right) x_k + \sum_{j=1}^{m_k} \theta_j^{(k)} x_{k-j} = \sum_{j=0}^{m_k} w_j x_{k-j},$$

$$v_k = \left(1 - \sum_{j=1}^{m_k} \theta_j^{(k)} \right) g(x_k) + \sum_{j=1}^{m_k} \theta_j^{(k)} g(x_{k-j}) = \sum_{j=0}^{m_k} w_j g(x_{k-j}),$$

where $\sum_{j=0}^{m_k} w_j = 1$. Algorithm 1 minimizes $\|u_k - v_k\|_2^2$ subject to $\sum_{j=0}^{m_k} w_j = 1$. If g is linear then the objective function is $\|u_k - g(u_k)\|_2^2$ and so $v_k = g(u_k)$ is the

vector from the affine subspace spanned by the current iterate and the previous m_k iterates that minimizes the residual of the fixed-point equation.

We will use an equivalent form of the method that stores in two matrices the differences of the successive iterates and their function values. These matrices are related by simple update formulae that can be exploited for an efficient implementation. This variant is given by Fang and Saad [17], Plasse [35], Walker [49], and Walker and Ni [50]. Here, Anderson acceleration is applied to the equivalent problem $f(x) = 0$, where $f(x) = g(x) - x$, instead of the fixed-point problem $g(x) = x$.

Algorithm 2 (Anderson acceleration) Given $x_0 \in \mathbb{R}^n$ and an integer $m \geq 1$ this algorithm produces a sequence x_k of iterates intended to converge to a zero of the function $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$. The following notation is used: $m_k = \min(m, k)$, $\Delta x_i = x_{i+1} - x_i$, $\mathcal{X}_k = [\Delta x_{k-m_k} \ \dots \ \Delta x_{k-1}]$, $f_i = f(x_i)$, $\Delta f_i = f_{i+1} - f_i$, and $\mathcal{F}_k = [\Delta f_{k-m_k} \ \dots \ \Delta f_{k-1}]$

```

1   $x_1 = x_0 + f(x_0)$ 
2  for  $k = 1, 2, \dots$  until convergence
3     $m_k = \min(m, k)$ 
4    Compute  $\gamma^{(k)} = (\gamma_{k-m_k}^{(k)}, \dots, \gamma_{k-1}^{(k)})^T \in \mathbb{R}^{m_k}$  that solves
       $\min_{\gamma \in \mathbb{R}^{m_k}} \|f_k - \mathcal{F}_k \gamma\|_2^2$ .
5     $\bar{x}_k = x_k - \sum_{i=k-m_k}^{k-1} \gamma_i^{(k)} \Delta x_i = x_k - \mathcal{X}_k \gamma^{(k)}$ 
6     $\bar{f}_k = f_k - \sum_{i=k-m_k}^{k-1} \gamma_i^{(k)} \Delta f_i = f_k - \mathcal{F}_k \gamma^{(k)}$ 
7     $x_{k+1} = \bar{x}_k + \bar{f}_k$ 
8  end

```

Line 4 of Algorithm 2 consists of the following major computations. We assume that \mathcal{F}_k has full rank and that the least squares problem is solved using a QR factorization of \mathcal{F}_k .

1. Compute $f_k = f(x_k)$.
2. Obtain a QR factorization of \mathcal{F}_k from that of \mathcal{F}_{k-1} . Since \mathcal{F}_k is just \mathcal{F}_{k-1} with the first column removed (for $k \geq m$) and a new last column added this is a QR factorization updating problem.
3. Solve the least squares problem using the QR factorization.

Assume that $k \geq m$. Since \mathcal{F}_{k-1} is $n \times m$ and its first column is removed in passing to \mathcal{F}_k , to update the R factor we need $m^2/2$ flops and to update Q an additional $6mn$ flops [21, p. 28]. Updating the QR factors after the last column has been added to the matrix costs $4mn + 3n$ flops [21, sec. 2.5.1]. Hence the total cost for step 2 above is at most $m^2/2 + 10mn + 3n$ flops. The cost of step 3 (which forms and solves by back substitution a triangular system involving R) is $2mn + m^2$ flops. Therefore, Anderson acceleration takes an additional $3m^2/2 + 12mn + 3n$ flops per step compared with the unaccelerated iteration.

More details of the updating scheme, as well as a strategy that removes more than one leading column of \mathcal{F}_k , if necessary, in order to ensure that it does not become too ill-conditioned are given in [49, 50, sec. 4]. A MATLAB implementation of Algorithm 2, which we use in our numerical experiments, is given in [49].

3 Accelerating the alternating projections method for the nearest correlation matrix

We now summarize the method to which we wish to apply Anderson acceleration: the alternating projections method for computing the nearest correlation matrix in the Frobenius norm. In its basic form the alternating projections method attempts to find a point in the intersection of two closed subspaces that is nearest to a given point by iteratively projecting onto each subspace. This simple idea is motivated by the fact that it is often easier to compute the individual projections onto the given subspaces than the projection onto their intersection. A detailed exposition of the origins and generalizations of alternating projections methods is given by Escalante and Raydan [15].

Let A be a given symmetric matrix of order n and define the sets

$$S_n = \{ X \in \mathbb{R}^{n \times n} : X \text{ is symmetric positive semidefinite} \}, \quad (2)$$

$$\mathcal{U}_n = \{ X = X^T \in \mathbb{R}^{n \times n} : x_{ii} = 1, i = 1 : n \}. \quad (3)$$

For the nearest correlation matrix problem, we are looking for the closest matrix to A that lies in the intersection of S_n and \mathcal{U}_n . Since these are convex sets rather than subspaces the alternating projections method has to be used in a modified form proposed by Dykstra [14], in which each projection incorporates a correction; each correction can be interpreted as a normal vector to the corresponding convex set. This correction is not needed for a translate of a subspace [9], so is only required for the projection onto S_n .

Denote the projections of a symmetric matrix X onto S_n and \mathcal{U}_n by $\mathcal{P}_{S_n}(X)$ and $\mathcal{P}_{\mathcal{U}_n}(X)$, respectively. The projection $\mathcal{P}_{S_n}(X)$ is computed from an eigenvalue decomposition of X by setting all the negative eigenvalues to zero and leaving the eigenvectors unchanged [23] (see also Theorem 3.4 below), while $\mathcal{P}_{\mathcal{U}_n}(X)$ is obtained by setting the diagonal elements of X to 1.

The use of alternating projections for computing the nearest correlation matrix was proposed by Higham [24, Algorithm 3.3] in the following form.

Algorithm 3 Given a symmetric matrix $A \in \mathbb{R}^{n \times n}$ this algorithm computes the nearest correlation matrix Y to A by alternating projections. It requires a convergence tolerance tol

```

1   $\Delta S_0 = 0, Y_0 = A$ 
2  for  $k = 1, 2, \dots$ 
3     $R_k = Y_{k-1} - \Delta S_{k-1}$ 
4     $X_k = \mathcal{P}_{S_n}(R_k)$  % Project onto  $S_n$ .
5     $\Delta S_k = X_k - R_k$  % Dykstra's correction.
6     $Y_k = \mathcal{P}_{\mathcal{U}_n}(X_k)$  % Project onto  $\mathcal{U}_n$ .
7    if  $\|Y_k - X_k\|_F \leq \text{tol}\|Y_k\|_F, Y = Y_k, \text{quit, end}$ 
8  end
    
```

It is known that X_k and Y_k both converge to the nearest correlation matrix as $k \rightarrow \infty$, with a convergence rate that is linear at best [24]. The termination criterion on line 7 is a simplification of the criterion

$$\max \left\{ \frac{\|X_k - X_{k-1}\|_F}{\|X_k\|_F}, \frac{\|Y_k - Y_{k-1}\|_F}{\|Y_k\|_F}, \frac{\|Y_k - X_k\|_F}{\|Y_k\|_F} \right\} \leq \text{tol} \tag{4}$$

used by Higham [24], who notes that the three terms inside the max are usually of similar size. We use only the last term, since the test on line 7 is equivalent to the robust stopping criterion for Dykstra’s algorithm proposed by Birgin and Raydan [5] and this choice works well in all our experiments.

Aitken extrapolation (see, for example, [10]) cannot be used to accelerate Algorithm 3 because it requires the underlying sequence to be linearly convergent, which is not guaranteed here. We therefore turn to Anderson acceleration. To use it we must recast Algorithm 3 as a fixed-point method, that is, define the function g for the iteration (1). We do this as follows, noting that two matrices are recurred: Y_k and ΔS_k , while X_k is only used for the convergence test.

Algorithm 4 (Fixed-point form of Algorithm 3) Given a symmetric matrix $A \in \mathbb{R}^{n \times n}$ this algorithm computes the nearest correlation matrix Y to A . It requires a convergence tolerance tol

```

1   $\Delta S_0 = 0, Y_0 = A$ 
2  for  $k = 1, 2, \dots$ 
3     $[X_k, Y_k, \Delta S_k] = g(Y_{k-1}, \Delta S_{k-1})$ 
4    if  $\|Y_k - X_k\|_F \leq \text{tol}\|Y_k\|_F, Y = Y_k, \text{quit, end}$ 
5  end
    
```

where the computation of $[X_k, Y_k, \Delta S_k] = g(Y_{k-1}, \Delta S_{k-1})$ is effected by

```

5   $R_k = Y_{k-1} - \Delta S_{k-1}$ 
6   $X_k = \mathcal{P}_{S_n}(R_k)$ 
7   $\Delta S_k = X_k - R_k$ 
8   $Y_k = \mathcal{P}_{\mathcal{U}_n}(X_k)$ 
    
```


To apply Anderson acceleration (Algorithm 2) we write the matrices in terms of vectors via the vec operator, which stacks the columns of a matrix one on top of the other. We denote by unvec the inverse operation to vec . The complete algorithm is then as follows.

Algorithm 5 Given a symmetric matrix $A \in \mathbb{R}^{n \times n}$ this algorithm attempts to compute the nearest correlation matrix Y to A by alternating projections with Anderson acceleration. It requires a convergence tolerance tol

- 1 Run Algorithm 2 on the function $f : \mathbb{R}^{2n^2} \rightarrow \mathbb{R}^{2n^2}$ given by $f(z) = \text{vec}(\tilde{g}(Z) - Z)$, where $z_k = \text{vec}(Z_k)$, $Z_k = (Y_k, \Delta S_k) \in \mathbb{R}^{n \times 2n}$ and $[X_k, \tilde{g}(Z_k)] = g(Z_k)$ for the function g defined by Algorithm 4. Terminate the iteration when $\|Y_k - X_k\|_2 / \|Y_k\|_2 \leq \text{tol}$. Denote the result by x_* .
 - 2 $Y = \text{unvec}(x_*)$
-

Note that the convergence criterion in Algorithm 5 is equivalent to that in Algorithm 4. Note also that, unlike Algorithms 3 and 4, Algorithm 5 is not guaranteed to converge, since there are no suitable convergence results for Anderson acceleration. Whether convergence can be proved under reasonable assumptions is an open question.

The cost per step of the standard alternating projections method (Algorithm 3) is dominated by the cost of computing $\mathcal{P}_{S_n}(R_k)$, which is $10n^3$ flops if we compute a full eigendecomposition, or $14n^3/3$ flops if we use tridiagonalization followed by bisection and inverse iteration (computing just the eigenpairs corresponding to the positive eigenvalues or the negative ones, depending which are fewer in number). One step of Anderson acceleration applied to the alternating projections method in the fixed-point form (Algorithm 4) uses $2n^2$ -sized vectors, so the method takes at most an additional $3m^2/2 + 24mn^2 + 6n^2$ flops per step. Since we find experimentally (see Section 4) that taking $m \leq 5$ (say) is sufficient, the additional cost of Anderson acceleration is $O(n^2)$ flops, which is negligible for large n . Anderson acceleration also incurs an increase in storage of $2n^2m$ elements.

We next consider two modifications of the alternating projections method for computing the nearest correlation matrix. The first is the problem variant in which specified elements of A have to remain fixed and the second requires the correlation matrix to have smallest eigenvalue bounded below by a positive tolerance δ .

3.1 Fixing elements

The nearest correlation matrix problem with fixed elements was previously investigated by Borsdorf [7, Chap. 7] and Lucas [30]. We first give some motivation for fixing elements.

In statistical applications the data from ℓ observations of n random variables is collected in an $\ell \times n$ matrix and it is often the case that some of the observations are missing. We may assume that the missing entries do not occur in the first n_1 columns, since we can permute the columns if necessary. One way to form correlations is via the pairwise deletion method [30, sec. 2.2]. It calculates the correlation coefficient

between a pair of vectors by using only the components available in both vectors simultaneously and the result is a unit diagonal symmetric matrix C of the form

$$C = \begin{matrix} & \begin{matrix} n_1 & n_2 \end{matrix} \\ \begin{matrix} n_1 \\ n_2 \end{matrix} & \begin{bmatrix} A & Y \\ Y^T & B \end{bmatrix} \end{matrix} \in \mathbb{R}^{n \times n}.$$

The leading block A is positive semidefinite (hence, a correlation matrix) because it is constructed from the columns of X that have no missing values, but there is no guarantee that the matrix C is positive semidefinite as well. Since the correlations in A are considered exact we wish to compute the nearest correlation matrix to C with this block unchanged.

In correlation stress testing [18], [40] the assets are split into two groups. Their correlation matrix can then be block-partitioned as

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{12}^T & C_{22} \end{bmatrix} \in \mathbb{R}^{n \times n},$$

where the inter-group correlations correspond to the diagonal blocks C_{11} and C_{22} and the off-diagonal block C_{12} carries the cross-group correlations. To see the effect of pushing risk parameters toward extreme levels the block C_{22} is replaced with a new correlation matrix \widehat{C}_{22} that reflects the changes to the second group of assets. If this results in an indefinite modified matrix we can again compute its nearest correlation matrix, but the C_{11} block should remain unchanged since the first group of assets was not affected.

In risk aggregation [2], [27] and large scale resource assessment [6] we have a generalization of the above constraint. Here, due to the methodology, we have a large correlation matrix with block structure, where the diagonal blocks represent individual groups, for example markets or geographical regions. Correlations in each group can be updated with more refined analysis locally and then the old diagonal blocks are replaced with the new ones. This might destroy the definiteness of the global matrix which we must then restore while keeping the new diagonal blocks unchanged.

In all of the above applications we are looking for the closest matrix in the Frobenius norm to a matrix A that lies in the intersection of the set S_n from (2) and

$$\mathcal{E}_n = \{ X = X^T \in \mathbb{R}^{n \times n} : x_{ii} = 1, i = 1, \dots, n \text{ and } x_{ij} = a_{ij} \text{ for } (i, j) \in \mathcal{N} \},$$

where \mathcal{N} denotes the index set of the fixed off-diagonal elements. Clearly, for $(i, j) \in \mathcal{N}$ we have $(j, i) \in \mathcal{N}$. The intersection $S_n \cap \mathcal{E}_n$ is nonempty, which is equivalent to the problem having a unique solution, if \mathcal{N} is chosen such that there exists a correlation matrix with the prescribed fixed elements. This need not be true for every \mathcal{N} , as the following simple example shows. Take

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{5}$$

and $\mathcal{N} = \{(2, 3), (3, 2), (2, 4), (4, 2), (3, 4), (4, 3)\}$. We cannot replace A with a valid correlation matrix while keeping the elements prescribed by \mathcal{N} fixed, since they correspond to the trailing 3×3 block of A , which is indefinite.

The Newton algorithm of [39] which solves the original nearest correlation matrix problem does not generalize to the fixed elements variant. According to Qi and Sun [41, p. 509], the Newton method that solves the so-called H -weighted nearest correlation matrix problem

$$\min\{ \|H \circ (A - X)\|_F^2 : X \text{ is a correlation matrix } \}, \tag{6}$$

where \circ is the Hadamard (elementwise) matrix product and H a symmetric elementwise nonnegative matrix, could be extended to fix elements but the details are not provided. Moreover, the documentation for the NAG [32] code `g02aj/nag_nearest_correlation_h_weight`, which solves (6), notes that the algorithm might not converge if the weights vary by several orders of magnitude.

The alternating projections method trivially generalizes to incorporate the fixed elements constraint: we simply need to replace the projection $\mathcal{P}_{\mathcal{U}_n}$ by the projection $\mathcal{P}_{\mathcal{E}_n}$ onto the set \mathcal{E}_n . For a symmetric matrix X this projection is given by

$$\mathcal{P}_{\mathcal{E}_n}(X)_{ij} = \begin{cases} 1, & i = j, \\ a_{ij}, & (i, j) \in \mathcal{N}, \\ x_{ij} & \text{otherwise.} \end{cases}$$

Since we have assumed that \mathcal{N} does not contain any indices corresponding to diagonal elements, $\mathcal{P}_{\mathcal{E}_n}$ remains well-defined even if A does not have unit diagonal. Algorithm 3 can now be used to solve this problem with a trivial modification of step 6, where $\mathcal{P}_{\mathcal{U}_n}$ is replaced with $\mathcal{P}_{\mathcal{E}_n}$. The extensive numerical experiments in [7, sec. 7] show that having the additional constraint can result in a significant increase in the number of iterations compared with solving the original problem, so using an acceleration method becomes even more appealing. The details of applying Anderson acceleration are the same as in the original problem.

The possible non-existence of a solution of this variant of the nearest correlation matrix problem must be reflected in the convergence test. For the matrix (5) it is easy to see that X_k and Y_k are both constant for $k \geq 1$, so the first two terms in (4) are zero. The last term of (4) is, however, of order 1 for all k . The convergence test on line 7 of Algorithm 3 is hence suitable both for the original problem and for variants that may not have a solution.

3.2 Imposing a lower bound on the smallest eigenvalue

For an invalid correlation matrix A with t nonpositive eigenvalues, from [24, Cor. 3.5] it follows that the nearest correlation matrix to A will have at least t zero eigenvalues. Singularity is an issue in applications where the inverse of a correlation matrix is needed, for example in multivariate data analysis [42] or regression [22, 37]. Hence, a common requirement in practice is to compute the nearest correlation matrix X to A with $\lambda_{\min}(X) \geq \delta$, where $\lambda_{\min}(X)$ denotes the smallest eigenvalue of X and δ is a given positive tolerance. Since for a correlation matrix $\text{trace}(X) = \sum_i \lambda_i(X) = n$, it follows that we must take $\delta \leq 1$.

The Newton algorithm [39] for the original nearest correlation matrix problem can be used to compute the solution to the problem with the constraint on λ_{\min} . We discuss this modification of the alternating projections method because it further

demonstrates the flexibility of the method, which can easily incorporate both the fixed elements constraint and the eigenvalue constraint, unlike the existing Newton methods.

For a given $0 \leq \delta \leq 1$ we define the set

$$S_n^\delta = \{ X = X^T \in \mathbb{R}^{n \times n} : \lambda_{\min}(X) \geq \delta \}. \tag{7}$$

Clearly, S_n^0 is the original S_n from (2). We are looking for the nearest matrix in the Frobenius norm to A from the intersection of S_n^δ and \mathcal{U}_n , where \mathcal{U}_n is defined in (3). The set S_n^δ is closed and convex for each δ and since $I_n \in S_n^\delta$ for every $0 \leq \delta \leq 1$ the closed convex set $S_n^\delta \cap \mathcal{U}_n$ is nonempty, which implies that this modification of the nearest correlation matrix problem has a unique solution. A formula for the projection $\mathcal{P}_{S_n^\delta}$ of a symmetric matrix onto the set S_n^δ is given by the following result of Cheng and Higham [12, Thm. 3.1].

Theorem 3.4 *Let the symmetric matrix $X \in \mathbb{R}^{n \times n}$ have the spectral decomposition $X = Q \text{diag}(\lambda_i) Q^T$ and let $\delta \geq 0$. Then for the Frobenius norm the unique matrix nearest to X with the smallest eigenvalue at least δ is given by*

$$\mathcal{P}_{S_n^\delta}(X) = Q \text{diag}(\tau_i) Q^T, \quad \tau_i = \begin{cases} \lambda_i, & \lambda_i \geq \delta \\ \delta, & \lambda_i < \delta. \end{cases}$$

Hence, to solve this version of the nearest correlation matrix problem we simply replace the projection \mathcal{P}_{S_n} in Algorithm 3 with $\mathcal{P}_{S_n^\delta}$. If, in addition, some elements of A must remain fixed, we replace $\mathcal{P}_{\mathcal{U}_n}$ with $\mathcal{P}_{\mathcal{E}_n}$ as well. However, note that the latter problem variant does not have a solution for all possible sets \mathcal{N} of fixed positions.

Finally, we briefly discuss how the use of the $\lambda_{\min}(X) \geq \delta$ constraint can address a subtle issue concerning methods for computing the nearest correlation matrix. The resulting matrix is expected to be a positive semidefinite matrix with unit diagonal closest to A . The Newton algorithm of [8] computes a positive semidefinite solution, but to guarantee a unit diagonal the computed matrix is diagonally scaled, which slightly increases the optimal distance to A . In the alternating projections method (Algorithm 3) the diagonal elements of the returned matrix are exactly 1 but this computed matrix might be indefinite since it is obtained by modifying the diagonal (as well as any other fixed elements) of the positive semidefinite projection. We could swap the order of the projections so that the result is a positive semidefinite matrix, up to roundoff, but then this matrix will not have an exactly unit diagonal. Probably the best solution to these problems is to impose a lower bound on λ_{\min} sufficiently large that changes of order the convergence tolerance, tol , will not affect the definiteness. For example, if $\text{tol} \approx 10^{-16}$ then $\delta \approx 10^{-8}$ would be adequate.

4 Numerical experiments

Now we present experiments that explore the effectiveness of Anderson acceleration at reducing the number of iterations, and the overall execution time, of the alternating projections method for computing the nearest correlation matrix.

Our experiments were carried out in MATLAB R2014a on a machine with an Intel Core i7-4910MQ 2.90GHz processor and 16GB RAM. We use the following algorithms.

1. `nearcorr`: the alternating projections method for the nearest correlation matrix, Algorithm 3, modified to incorporate both the fixed elements constraint and the lower bound δ on the smallest eigenvalue by replacing $\mathcal{P}_{\mathcal{U}_n}$ with $\mathcal{P}_{\mathcal{E}_n}$ and $\mathcal{P}_{\mathcal{S}_n}$ with $\mathcal{P}_{\mathcal{S}_n^\delta}$, as described in Sections 3.1 and 3.2. The number of iterations for `nearcorr` is denoted by `it`. Our code is based on that from [25].
2. `nearcorr_AA`: Algorithm 5 applied to `nearcorr`. We use the Anderson acceleration implementation from [49], which employs QR factorization with updating, as described in Section 2. The number of iterations is denoted by `itAA`.

The convergence tolerance `tol` is set to nu , where n is the order of the matrix and $u \approx 1.1 \times 10^{-16}$ is the unit roundoff.

Convergence is guaranteed for the alternating projections algorithm assuming there are no fixed off-diagonal elements, but could potentially be destroyed by Anderson acceleration, for which we have no convergence guarantees. However, in every test Anderson acceleration and the corresponding unaccelerated algorithm produced computed matrices X with values of $\|A - X\|_F$ agreeing to within a small multiple of the convergence tolerance.

In the first three experiments, we have no fixed elements and set $\delta = 0$, that is, we are solving the standard nearest correlation matrix problem.

Experiment 1 We first compare the number of iterations for `nearcorr` and `nearcorr_AA` as we vary the parameter m on four small examples of invalid correlation matrices found in the literature. They are a matrix of order 4 from Turkey, Epperlein, and Christofides [47], a matrix of order 5 from Bhansali and Wise [4], a matrix of order 6 constructed from foreign exchange trading data supplied by the Royal Bank of Scotland [31], and a matrix of order 7 from Finger [18], all of which are listed in the appendix. The results are given in Table 1.

Clearly, using Anderson acceleration leads to a significant decrease in the number of iterations, even for $m = 1$, with a 25-fold decrease achieved for the $n = 6$ matrix with $m = 6$. The number of iterations begins to stagnate as m grows, which is consistent with the reported behaviour of Anderson acceleration in the literature.

Table 1 Iteration counts for four small examples for `nearcorr` and `nearcorr_AA`, for varying m (Experiment 1)

| n | it | itAA | | | | | |
|-----|-----|---------|---------|---------|---------|---------|---------|
| | | $m = 1$ | $m = 2$ | $m = 3$ | $m = 4$ | $m = 5$ | $m = 6$ |
| 4 | 39 | 15 | 10 | 9 | 9 | 9 | 9 |
| 5 | 27 | 17 | 14 | 12 | 11 | 10 | 10 |
| 6 | 801 | 305 | 212 | 117 | 126 | 40 | 31 |
| 7 | 33 | 15 | 10 | 10 | 10 | 9 | 9 |

Table 2 Iteration counts and computation times in seconds for `nearcorr` and `nearcorr_AA` with $m = 2$ for six `RiskMetrics` matrices of order 387 (Experiment 2)

| Matrix | nearcorr | | nearcorr_AA | | | |
|--------|----------|------|-------------|------|-------|------|
| | it | t | itAA | t | t_apm | t_AA |
| 1 | 26 | 0.46 | 15 | 0.45 | 0.26 | 0.12 |
| 2 | 50 | 0.83 | 24 | 0.73 | 0.41 | 0.19 |
| 3 | 24 | 0.43 | 13 | 0.38 | 0.23 | 0.09 |
| 4 | 47 | 0.88 | 22 | 0.68 | 0.40 | 0.17 |
| 5 | 34 | 0.56 | 18 | 0.53 | 0.30 | 0.14 |
| 6 | 20 | 0.33 | 12 | 0.35 | 0.20 | 0.09 |

Experiment 2 Now we compare the iteration count and the computation time for `nearcorr` and `nearcorr_AA` with $m = 2$ for six matrices from the `RiskMetrics` database, as used in [8]. The documentation says that the underlying data sets “contain consistently calculated volatilities and correlation forecasts for use in estimating market risk. The asset classes covered are government bonds, money markets, swaps, foreign exchange and equity indices (where applicable) for 31 currencies, and commodities.” Each matrix has dimension 387.

In Table 2 we report the number of iterations along with t , the total run time in seconds for each algorithm, and t_{apm} and t_{AA} for `nearcorr_AA`, which are the total time taken in calls to the function g from Algorithm 4 and in computing the quantities for the convergence test, and the time taken to solve the least-squares problems, respectively. Anderson acceleration roughly halves the number of iterations and the total computation time for `nearcorr_AA` is a little less than for `nearcorr` in the first 5 examples.

The missing time $t - t_{apm} - t_{AA}$ for `nearcorr_AA` represents MATLAB overheads, such as in the `vec` and `unvec` conversions of Algorithm 5. Computing the eigenvalue decomposition, which is the dominant cost for the alternating projections method, remains the main contributing factor to the computation time of `nearcorr_AA`, with the least-squares update and solve taking less than half as much time.

Experiment 3 In the previous experiments our test matrices were small and the total computation time was not an issue. In order to illustrate the dramatic improvement Anderson acceleration can bring to `nearcorr` we next compare `nearcorr` and `nearcorr_AA` with $m = 2$ on two large invalid correlation matrices of stock data provided by a fund management company. The first of order 1399 is highly rank-deficient and the second of order 3120 is of full rank. The results are presented in Table 3. We again see a very sharp drop in the number of iterations, with `nearcorr_AA` taking less than a third of the iterations for `nearcorr`. This results in a significant reduction in the computation time, with a speedup of as much as 2.9. Comparing the times for the alternating projections part and the least-squares part of `nearcorr_AA` we see that the former heavily dominates the latter.

Table 3 Iteration counts and computation times for `nearcorr` and `nearcorr_AA` with $m = 2$ for `cor1399` and `cor3120` (Experiment 3)

| n | <code>nearcorr</code> | | <code>nearcorr_AA</code> | | | | speedup |
|------|-----------------------|--------|--------------------------|-------|-------|-------|---------|
| | it | t | itAA | t | t_apm | t_AA | |
| 1399 | 476 | 219.0 | 124 | 75.0 | 49.6 | 16.0 | 2.9 |
| 3120 | 559 | 2746.4 | 174 | 999.5 | 778.5 | 137.7 | 2.7 |

We next focus on the nearest correlation matrix problem variant with some fixed off-diagonal elements ($\delta = 0$).

Experiment 4 We compare the performance of the methods on the following three examples. The first is the matrix of order 7 that we have used in our first experiment. The original requirement in [18] was to compute the nearest correlation matrix having the same leading principal 3×3 submatrix. The second example is a symmetric indefinite block 9×9 matrix with each block of order 10, provided by a finance company. We need to compute the nearest positive semidefinite matrix to it while preserving the (positive definite) (1,1) block, the (positive) diagonal, and the diagonals in each of the remaining blocks in the first block-row and block-column. The large matrix does not have a unit diagonal but this makes no difference to the methods since these elements are fixed. In our third example, we have an invalid correlation matrix of order 94 for carbon dioxide storage assessment units for the Rocky Mountains region of the United States that was generated during the national assessment of carbon dioxide storage resources [48]. Due to the aggregation methodology construction, the matrix has a natural block structure. Its twelve diagonal blocks, with respective sizes 12, 5, 1, 14, 12, 1, 10, 4, 5, 9, 13, and 8, correspond to individual basins in the region and are all positive definite. We wish to compute the nearest correlation matrix to the large matrix while keeping all diagonal blocks unchanged.

Table 4 reports the number of iterations for `nearcorr` with no fixed elements (`it`), the number of iterations for `nearcorr` with the required elements fixed (`it_fe`) and the number of iterations for Anderson acceleration applied to the latter (`itAA_fe`) with m varying from 1 to 5 for our three examples. Table 5 presents the computation time in seconds, `time_fe` and `time_fe_AA`, for the latter two algorithms. We include `nearcorr` with no fixed elements only to demonstrate the effect

Table 4 Iteration counts for `nearcorr`, `nearcorr` with fixed elements, and Anderson acceleration of the latter with varying m (Experiment 4)

| n | it | it_fe | itAA_fe | | | | |
|-----|----|-------|---------|---------|---------|---------|---------|
| | | | $m = 1$ | $m = 2$ | $m = 3$ | $m = 4$ | $m = 5$ |
| 7 | 33 | 34 | 14 | 11 | 10 | 9 | 9 |
| 90 | 29 | 169 | 93 | 70 | 55 | 45 | 39 |
| 94 | 18 | 40 | 15 | 14 | 12 | 12 | 12 |

Table 5 Computation times for `nearcorr` with fixed elements and Anderson acceleration applied to it, with varying m (Experiment 4)

| n | time_fe | time_fe_AA | | | | |
|-----|---------|------------|---------|---------|---------|---------|
| | | $m = 1$ | $m = 2$ | $m = 3$ | $m = 4$ | $m = 5$ |
| 7 | 3.40e-3 | 2.51e-3 | 2.20e-3 | 2.11e-3 | 1.20e-3 | 1.14e-3 |
| 90 | 1.71e-1 | 1.33e-1 | 1.14e-1 | 9.06e-2 | 7.93e-2 | 8.02e-2 |
| 94 | 5.21e-2 | 2.06e-2 | 1.98e-2 | 1.87e-2 | 2.54e-2 | 2.19e-2 |

on the number of iterations of including this constraint, and as this method does not solve our problem we do not run Anderson acceleration on it. The second and third examples show that the constraint of having fixed elements can significantly increase the number of iterations for the alternating projections method compared with the standard nearest correlation matrix problem. From the number of iterations for `nearcorr` with fixed elements and the accelerated algorithm we see that using Anderson acceleration reduces the number of iterations by a similar factor as in the experiments for accelerating the original `nearcorr`. Hence while the additional constraint makes the problem harder to solve by alternating projections it does not affect the speedup of the Anderson acceleration scheme.

Experiment 5 In our second experiment with fixed elements we generate random invalid correlation matrices of order n , with n equal to 200, 400, 600, and 800 and compare the computation time of `nearcorr` and `nearcorr_AA` for varying m , where for each matrix a leading block of size $n/2$ is kept fixed in computing the nearest correlation matrix. We generate the leading block by the MATLAB function call `gallery('randcorr', n/2)` and embed it into an indefinite unit diagonal matrix of size n where the off-diagonal elements are taken from the uniform distribution on $[-1, 1]$. The results reported in Table 6 show that the time decreases for m up to 2, but for $m = 4$ or 5 we have an increase in the computation time, which further confirms the merit of keeping m very small. In each example Anderson acceleration achieves a significant reduction in computation time.

Our third set of experiments concerns the nearest correlation matrix problem with a lower bound on the smallest eigenvalue and no fixed elements.

Table 6 Computation times for `nearcorr` and `nearcorr_AA` with varying m for four examples where the leading $n/2 \times n/2$ block of a random matrix of size n remains fixed (Experiment 5)

| n | time_fe | time_fe_AA | | | | |
|-----|---------|------------|---------|---------|---------|---------|
| | | $m = 1$ | $m = 2$ | $m = 3$ | $m = 4$ | $m = 5$ |
| 200 | 6.41 | 4.42 | 2.77 | 2.67 | 2.29 | 2.45 |
| 400 | 18.53 | 13.44 | 9.35 | 8.10 | 6.91 | 7.35 |
| 600 | 59.47 | 47.51 | 28.15 | 32.04 | 26.25 | 31.50 |
| 800 | 136.12 | 82.23 | 53.76 | 63.77 | 47.61 | 51.35 |

Table 7 Iteration counts for four small examples for `nearcorr` and `nearcorr_AA`, for varying m and two values of δ . (Experiment 6)

| | | $\delta = 10^{-8}$ | | | | | |
|-----|-----|--------------------|---------|---------|---------|---------|---------|
| n | it | itAA | | | | | |
| | | $m = 1$ | $m = 2$ | $m = 3$ | $m = 4$ | $m = 5$ | $m = 6$ |
| 4 | 39 | 15 | 10 | 9 | 9 | 9 | 10 |
| 5 | 27 | 17 | 14 | 12 | 11 | 10 | 10 |
| 6 | 802 | 280 | 177 | 114 | 58 | 39 | 30 |
| 7 | 33 | 15 | 10 | 10 | 10 | 9 | 9 |
| | | $\delta = 0.1$ | | | | | |
| 4 | 66 | 31 | 19 | 16 | 13 | 14 | 13 |
| 5 | 34 | 23 | 15 | 14 | 12 | 12 | 12 |
| 6 | 895 | 269 | 216 | 127 | 59 | 48 | 41 |
| 7 | 54 | 31 | 24 | 15 | 15 | 14 | 14 |

Experiment 6 We first run `nearcorr` on the four small test matrices already used in Table 1 for $\delta = 10^{-8}$ and $\delta = 0.1$. The results, reported in Table 7, show that for the smaller value of $\delta = 10^{-8}$ the number of iterations is almost identical to the data in Table 1, but here the positive definiteness of the solution is guaranteed. For the larger value $\delta = 0.1$, the number of iterations is increased compared with $\delta = 0$. As with the fixed elements constraint, we see that Anderson acceleration again reduces the iteration number by a similar factor as in the unconstrained case, that is, its performance is not affected by including the bound on the smallest eigenvalue.

Experiment 7 The benefits of Anderson acceleration in the positive definite case are even more evident if we reproduce Experiment 2, now using `nearcorr` with $\delta = 0.1$ and compare the results in Table 8 with those in Table 2. Computing the positive definite solution takes between 30 and 90 times more iterations than computing the semidefinite nearest correlation matrix but Anderson acceleration now reduces the number of iterations by a factor between 3.6 and 4.6, compared with halving the iterations in the original experiment, which shows that Anderson acceleration can be even more effective for constrained nearest correlation matrix problems than for the original problem. We also see that `nearcorr_AA` requires approximately half the time of `nearcorr`.

We now combine the constraints of keeping elements fixed and of positive definiteness.

Experiment 8 We take the three matrices from Experiment 4 with fixed elements and run `nearcorr` and `nearcorr_AA` with $\delta = 0.1$, with varying m . Note that in this case we have no guarantee of the existence of a feasible point and in fact for the

Table 8 Iteration counts and computation times in seconds for `nearcorr` with $\delta = 0.1$ and `nearcorr_AA` with $m = 2$ for six RiskMetrics matrices of order 387 (Experiment 7)

| Matrix | nearcorr | | nearcorr_AA | | | |
|--------|----------|-------|-------------|-------|-------|------|
| | it | t | itAA | t | t_apm | t_AA |
| 1 | 1410 | 20.50 | 383 | 10.77 | 5.70 | 3.12 |
| 2 | 2100 | 33.93 | 513 | 15.83 | 8.52 | 4.56 |
| 3 | 1900 | 31.14 | 414 | 11.58 | 5.97 | 3.54 |
| 4 | 1586 | 29.06 | 369 | 12.83 | 7.09 | 3.54 |
| 5 | 1812 | 31.30 | 400 | 12.99 | 7.16 | 3.62 |
| 6 | 1794 | 29.08 | 393 | 11.63 | 6.20 | 3.40 |

second matrix ($n = 90$) the algorithms do not converge within 100,000 iterations for the default tolerance and hence we exclude this example and present in Table 9 only the results for the test matrices of order $n = 7$ and $n = 94$. We note the increase in the number of iterations compared with the data in Table 4 where we only fixed the elements. Anderson acceleration (with $m = 5$) reduces the iterations by a factor of 3.6 for the smaller matrix and 6.7 for the larger, while in the original experiment the factors were 3.8 and 3.3.

Experiment 9 As a final experiment we use the four matrices from Experiment 1 to compare Anderson acceleration with the acceleration scheme from [29]. Table 10 shows the number of iterations, `it_2`, for that scheme, in which we set its safeguard parameter ε to 10^{-14} and use the same convergence tolerance as in all our experiments. The number of iterations for the acceleration scheme is the same as for the unaccelerated method in each case except for the matrix with $n = 6$, and in that case we see a reduction in the number of iterations by a factor 1.1 versus 3.8 for Anderson acceleration. In all test cases, after a few initial iterations the mixing parameter α_k needed for the scheme [29] could not be computed because the safeguard was triggered. We conclude that the acceleration scheme of [29] is not

Table 9 Iteration counts and computation times in seconds for `nearcorr` and `nearcorr_AA` with $\delta = 0.1$ and varying m for two examples with fixed elements (Experiment 8)

| nearcorr | | nearcorr_AA | | | | | | | | | | |
|----------|---------|-------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----|
| | | $m = 1$ | | $m = 2$ | | $m = 3$ | | $m = 4$ | | $m = 5$ | | |
| n | t | it | t | it | t | it | t | it | t | it | t | it |
| 7 | 2.98e-3 | 54 | 4.95e-3 | 31 | 4.57e-3 | 25 | 2.59e-3 | 16 | 2.74e-3 | 15 | 2.75e-3 | 15 |
| 94 | 1.25e-1 | 128 | 5.24e-2 | 36 | 4.10e-2 | 25 | 4.32e-2 | 24 | 3.91e-2 | 20 | 3.93e-2 | 19 |

Table 10 Iteration counts for four small examples for `nearcorr`, `nearcorr_AA` with $m = 2$, and the acceleration scheme from [29] (Experiment 9)

| n | it | itAA | it_2 |
|-----|-----|------|------|
| 4 | 39 | 10 | 39 |
| 5 | 27 | 14 | 27 |
| 6 | 801 | 212 | 725 |
| 7 | 33 | 10 | 33 |

competitive with Anderson acceleration on this class of problems because it displays the “orthogonality property” discussed in [29, Rem. 1].

To summarize, in these experiments we have found that Anderson acceleration of the alternating projections method for the nearest correlation matrix, with an appropriate choice of $m \in [1, 6]$, results in a reduction in the number of iterations by a factor of at least 2 for the standard algorithm and a factor at least 3 when additional constraints are included. The factors can be much larger than these worst-cases, especially in the experiments with additional constraints, where we saw a reduction in the number of iterations by a factor 21.8 in Table 7. Acceleration therefore tends to produce the greatest benefits on the problems that alternating projections finds the hardest. Moreover, the reduction in the number of iterations is generally reflected in the run times, modulo MATLAB overheads.

5 Conclusions

Although Anderson acceleration is well established in quantum chemistry applications and has recently started to attract the attention of numerical analysts, the method is still not well known in the numerical analysis community. Indeed it has not, to our knowledge, previously been applied to alternating projections methods. The main contribution of this work is to show that Anderson acceleration with history length m equal to 2 or 3 works remarkably well in conjunction with the widely used alternating projections method of Higham [24] for computing the nearest correlation matrix, both in its original form and in the forms that allow elements of the matrix to be fixed or a lower bound to be imposed on the smallest eigenvalue. This is particularly significant for the nearest correlation matrix problem with fixed elements because no Newton method is available for it. Our recommendation for m is based on the balance between the reduction in both the number of iterations and the computation time: even though larger values of m in some examples lead to a further decrease in the number of iterations the computation time sometimes increases for m larger than 2 or 3. MATLAB implementations of the algorithms can be found at <https://github.com/higham/anderson-accel-ncm>.

The success of Anderson acceleration in the nearest correlation matrix context suggests the possibility of using it in conjunction with other projection algorithms,

such as those for feasibility problems, that is, finding a point (not necessarily the nearest one) in the intersection of several convex sets. Such algorithms include the (uncorrected) alternating projections method and the Douglas–Rachford method [3]. Gould [20, p. 10] states that an efficient acceleration scheme is needed for projection methods if they are to be successfully applied to real-life convex feasibility problems. Our work suggests that Anderson acceleration could make projection methods competitive in this context, and in future work we intend to investigate this possibility.

Acknowledgments We are grateful to Dr Madalyn Blondes of the U.S. Geological Survey and Dr George Mylnikov of Fischer Francis Trees & Watts, Inc. for providing us with test matrices. We also thank Professor Marcos Raydan for a helpful discussion about the acceleration scheme in [29].

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix A

We list the four invalid correlation matrices used in Experiment 1.

1. Turkay, Epperlein, and Christofides [47]:

$$A = \begin{bmatrix} 1 & -0.55 & -0.15 & -0.10 \\ -0.55 & 1 & 0.90 & 0.90 \\ -0.15 & 0.90 & 1 & 0.90 \\ -0.10 & 0.90 & 0.90 & 1 \end{bmatrix}.$$

2. Bhansali and Wise [4]:

$$A = \begin{bmatrix} 1 & -0.50 & -0.30 & -0.25 & -0.70 \\ -0.50 & 1 & 0.90 & 0.30 & 0.70 \\ -0.30 & 0.90 & 1 & 0.25 & 0.20 \\ -0.25 & 0.30 & 0.25 & 1 & 0.75 \\ -0.70 & 0.70 & 0.20 & 0.75 & 1 \end{bmatrix}.$$

3. Minabutdinov, Manaev, and Bouev [31]: $D^{-1/2}AD^{-1/2}$, where $D = \text{diag}(A)$ with

$$A = \begin{bmatrix} 0.010712 & 0.000654 & 0.002391 & 0.010059 & -0.008321 & 0.001738 \\ 0.000654 & 0.000004 & 0.002917 & 0.000650 & 0.002263 & 0.002913 \\ 0.002391 & 0.002917 & 0.013225 & -0.000525 & 0.010834 & 0.010309 \\ 0.010059 & 0.000650 & -0.000525 & 0.009409 & -0.010584 & -0.001175 \\ -0.008321 & 0.002263 & 0.010834 & -0.010584 & 0.019155 & 0.008571 \\ 0.001738 & 0.002913 & 0.010309 & -0.001175 & 0.008571 & 0.007396 \end{bmatrix}.$$

4. Finger [18]:

$$A = \begin{bmatrix} 1 & 0.18 & -0.13 & -0.26 & 0.19 & -0.25 & -0.12 \\ 0.18 & 1 & 0.22 & -0.14 & 0.31 & 0.16 & 0.09 \\ -0.13 & 0.22 & 1 & 0.06 & -0.08 & 0.04 & 0.04 \\ -0.26 & -0.14 & 0.06 & 1 & 0.85 & 0.85 & 0.85 \\ 0.19 & 0.31 & -0.08 & 0.85 & 1 & 0.85 & 0.85 \\ -0.25 & 0.16 & 0.04 & 0.85 & 0.85 & 1 & 0.85 \\ -0.12 & 0.09 & 0.04 & 0.85 & 0.85 & 0.85 & 1 \end{bmatrix}.$$

References

1. Anderson, D.G.: Iterative procedures for nonlinear integral equations. *J. Assoc. Comput. Mach.* **12**(4), 547–560 (1965). doi:[10.1145/321296.321305](https://doi.org/10.1145/321296.321305)
2. Anderson, G., Goldberg, L., Kercheval, A.N., Miller, G., Sorge, K.: On the aggregation of local risk models for global risk management. *J. Risk* **8**(1), 25–40 (2005)
3. Aragón Artacho, F.J., Borwein, J.M., Tam, M.K.: Douglas–Rachford feasibility methods for matrix completion problems. *The ANZIAM Journal* **55**, 299–326 (2014). doi:[10.1017/S1446181114000145](https://doi.org/10.1017/S1446181114000145)
4. Bhansali, V., Wise, M.B.: Forecasting portfolio risk in normal and stressed markets. *J. Risk* **4**(1), 91–106 (2001)
5. Birgin, E.G., Raydan, M.: Robust stopping criteria for Dykstra’s algorithm. *SIAM J. Sci. Comput.* **26**(4), 1405–1414 (2005). doi:[10.1137/03060062X](https://doi.org/10.1137/03060062X)
6. Blondes, M.S., Schuenemeyer, J.H., Olea, R.A., Drew, L.J.: Aggregation of carbon dioxide sequestration storage assessment units. *Stoch. Environ. Res. Risk Assess.* **27**(8), 1839–1859 (2013). doi:[10.1007/s00477-013-0718-x](https://doi.org/10.1007/s00477-013-0718-x)
7. Borsdorf, R.: A Newton algorithm for the nearest correlation matrix. M.Sc. Thesis, The University of Manchester, Manchester, UK. MIMS EPrint 2008.49, Manchester Institute for Mathematical Sciences, The University of Manchester (2007)
8. Borsdorf, R., Higham, N.J.: A preconditioned Newton algorithm for the nearest correlation matrix. *IMA J. Numer. Anal.* **30**(1), 94–107 (2010). doi:[10.1093/imanum/drn085](https://doi.org/10.1093/imanum/drn085)
9. Boyle, J.P., Dykstra, R.L.: A method for finding projections onto the intersection of convex sets in Hilbert spaces. In: Dykstra, R., Robertson, T., Wright, F. (eds.) *Advances in Order Restricted Statistical Inference, Lecture Notes in Statistics*, vol. 37, pp. 28–47. Springer, New York (1986). doi:[10.1007/978-1-4613-9940-7_3](https://doi.org/10.1007/978-1-4613-9940-7_3)
10. Brezinski, C., Redivo-Zaglia, M.: *Extrapolation Methods: Theory and Practice. Studies in Computational Mathematics*, vol. 2. North-Holland, Amsterdam (1991)
11. Broyden, C.G.: A class of methods for solving nonlinear simultaneous equations. *Math. Comp.* **19**, 577–593 (1965). doi:[10.2307/2003941](https://doi.org/10.2307/2003941)
12. Cheng, S.H., Higham, N.J.: A modified Cholesky algorithm based on a symmetric indefinite factorization. *SIAM J. Matrix Anal. Appl.* **19**(4), 1097–1110 (1998). doi:[10.1137/S0895479896302898](https://doi.org/10.1137/S0895479896302898)
13. Demirtas, H., Hedeker, D., Mermelstein, R.J.: Simulation of massive public health data by power polynomials. *Statist. Med.* **31**(27), 3337–3346 (2012). doi:[10.1002/sim.5362](https://doi.org/10.1002/sim.5362)
14. Dykstra, R.L.: An algorithm for restricted least squares regression. *J. Amer. Statist. Assoc.* **78**, 837–842 (1983). doi:[10.1080/01621459.1983.10477029](https://doi.org/10.1080/01621459.1983.10477029)
15. Escalante, R., Raydan, M.: *Alternating Projection Methods. Society for Industrial and Applied Mathematics, Philadelphia, PA* (2011)
16. Eyert, V.: A comparative study on methods for convergence acceleration of iterative vector sequences. *J. Comput. Phys.* **124**(2), 271–285 (1996). doi:[10.1006/jcph.1996.0059](https://doi.org/10.1006/jcph.1996.0059)
17. Fang, H.R., Saad, Y.: Two classes of multisection methods for nonlinear acceleration. *Numer. Linear Algebra Appl.* **16**(3), 197–221 (2009). doi:[10.1002/nla.617](https://doi.org/10.1002/nla.617)
18. Finger, C.C.: A methodology to stress correlations. *RiskMetrics Monitor Fourth Quarter*, pp. 3–11 (1997)

19. Fripp, M.: Greenhouse gas emissions from operating reserves used to backup large-scale wind power. *Environ. Sci. Technol.* **45**(21), 9405–9412 (2011). doi:[10.1021/es200417b](https://doi.org/10.1021/es200417b)
20. Gould, N.I.M.: How good are projection methods for convex feasibility problems? *Comput. Optim. Appl.* **40**, 1–12 (2008). doi:[10.1007/s10589-007-9073-5](https://doi.org/10.1007/s10589-007-9073-5)
21. Hammarling, S., Lucas, C.: Updating the QR factorization and the least squares problem. MIMS EPrint 2008.111, Manchester Institute for Mathematical Sciences, The University of Manchester, UK (2008)
22. Hawkins, D.M., Eplett, W.J.R.: The Cholesky factorization of the inverse correlation or covariance matrix in multiple regression. *Technometrics* **24**(3), 191–198 (1982). <http://www.jstor.org/stable/1268678>
23. Higham, N.J.: Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra Appl.* **103**, 103–118 (1988). doi:[10.1016/0024-3795\(88\)90223-6](https://doi.org/10.1016/0024-3795(88)90223-6)
24. Higham, N.J.: Computing the nearest correlation matrix—A problem from finance. *IMA J. Numer. Anal.* **22**(3), 329–343 (2002). doi:[10.1093/imanum/22.3.329](https://doi.org/10.1093/imanum/22.3.329)
25. Higham, N.J.: The nearest correlation matrix. <https://nickhigham.wordpress.com/2013/02/13/the-nearest-correlation-matrix> (2013)
26. Kelley, C.T.: *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA (1995). http://www.siam.org/books/textbooks/fr16_book.pdf
27. Kercheval, A.N.: Optimal covariances in risk model aggregation. In: *Proceedings of the 3rd IASTED International Conference on Financial Engineering and Applications*, pp. 30–35. ACTA Press, Calgary (2006)
28. Kvaalen, E.: A faster Broyden method. *BIT* **31**(2), 369–372 (1991). doi:[10.1007/BF01931297](https://doi.org/10.1007/BF01931297)
29. López, W., Raydan, M.: An acceleration scheme for Dykstra’s algorithm. *Comput. Optim. Appl.* doi:[10.1007/s10589-015-9768-y](https://doi.org/10.1007/s10589-015-9768-y) (2015)
30. Lucas, C.: *Computing nearest covariance and correlation matrices*. M.Sc. Thesis, University of Manchester, Manchester, England (2001)
31. Minabutdinov, A., Manaev, I., Bouev, M.: Finding the nearest valid covariance matrix: A FX market case. Working paper Ec-07/13, Department of Economics, European University at St. Petersburg, St. Petersburg, Russia. Revised June 2014 (2013)
32. NAG Library. NAG Ltd., Oxford, UK. <http://www.nag.co.uk>
33. Olshanskii, M., Tyrtysnikov, E.: *Iterative Methods for Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA (2014)
34. Pezzulli, S., Frederic, P., Majithia, S., Sabbagh, S., Black, E., Sutton, R., Stephenson, D.: The seasonal forecast of electricity demand: a hierarchical Bayesian model with climatological weather generator. *Appl. Stochastic Models Bus. Ind.* **22**(2), 113–125 (2006). doi:[10.1002/asmb.622](https://doi.org/10.1002/asmb.622)
35. Plasse, J.H.: The EM algorithm in multivariate Gaussian mixture models using Anderson acceleration. M.Sc. Thesis, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA. <http://www.wpi.edu/Pubs/ETD/Available/etd-042513-091152/> (2013)
36. Potra, F.A., Engler, H.: A characterization of the behavior of the Anderson acceleration on linear problems. *Linear Algebra Appl.* **438**(3), 1002–1011 (2013). doi:[10.1016/j.laa.2012.09.008](https://doi.org/10.1016/j.laa.2012.09.008)
37. Pourahmadi, M.: Joint mean-covariance models with applications to longitudinal data: unconstrained parameterisation. *Biometrika* **86**(3), 677–690 (1999). <http://www.jstor.org/stable/2673662>
38. Pulay, P.: Convergence acceleration of iterative sequences. The case of SCF iteration. *Chem. Phys. Lett.* **73**(2), 393–398 (1980). doi:[10.1016/0009-2614\(80\)80396-4](https://doi.org/10.1016/0009-2614(80)80396-4)
39. Qi, H., Sun, D.: A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM J. Matrix Anal. Appl.* **28**(2), 360–385 (2006). doi:[10.1137/050624509](https://doi.org/10.1137/050624509)
40. Qi, H., Sun, D.: Correlation stress testing for value-at-risk: an unconstrained convex optimization approach. *Comput. Optim. Appl.* **45**(2), 427–462 (2010). doi:[10.1007/s10589-008-9231-4](https://doi.org/10.1007/s10589-008-9231-4)
41. Qi, H., Sun, D.: An augmented Lagrangian dual approach for the H -weighted nearest correlation matrix problem. *IMA J. Numer. Anal.* **31**, 491–511 (2011). doi:[10.1093/imanum/drp031](https://doi.org/10.1093/imanum/drp031)
42. Raveh, A.: On the use of the inverse of the correlation matrix in multivariate data analysis. *Am. Stat.* **39**(1), 39–42 (1985). doi:[10.2307/2683904](https://doi.org/10.2307/2683904)
43. Rohwedder, T., Schneider, R.: An analysis for the DIIS acceleration method used in quantum chemistry calculations. *J. Math. Chem.* **49**(9), 1889–1914 (2011). doi:[10.1007/s10910-011-9863-y](https://doi.org/10.1007/s10910-011-9863-y)
44. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.* **7**(3), 856–869 (1986). doi:[10.1137/0907058](https://doi.org/10.1137/0907058)

45. Toni, T., Tidor, B.: Combined model of intrinsic and extrinsic variability for computational network design with application to synthetic biology. *PLoS Comput. Biol.* **9**(3), e1002960 (2013). doi:[10.1371/journal.pcbi.1002960](https://doi.org/10.1371/journal.pcbi.1002960)
46. Toth, A., Kelley, C.T.: Convergence analysis for Anderson acceleration. *SIAM J. Numer. Anal.* **53**(2), 805–819 (2015). doi:[10.1137/130919398](https://doi.org/10.1137/130919398)
47. Turkay, S., Epperlein, E., Christofides, N.: Correlation stress testing for value-at-risk. *J. Risk* **5**(4), 75–89 (2003)
48. U.S. Geological Survey Geologic Carbon Dioxide Storage Resources Assessment Team: National Assessment of Geologic Carbon Dioxide Storage Resources—Results (Ver. 1.1, September, 2013). <http://pubs.usgs.gov/circ/1386> (2013)
49. Walker, H.F.: Anderson acceleration: Algorithms and implementations. Tech. Rep. MS-6-15-50, Mathematical Sciences Department, Worcester Polytechnic Institute, Worcester, MA 01609, USA (2011)
50. Walker, H.F., Ni, P.: Anderson acceleration for fixed-point iterations. *SIAM J. Numer. Anal.* **49**(4), 1715–1735 (2011). doi:[10.1137/10078356X](https://doi.org/10.1137/10078356X)
51. Wang, Q.J., Robertson, D.E., Chiew, F.H.S.: A Bayesian joint probability modeling approach for seasonal forecasting of streamflows at multiple sites. *Water Resour. Res.* **45**(5) (2009). doi:[10.1029/2008WR007355](https://doi.org/10.1029/2008WR007355)
52. Wang, X., Anderson, E., Steenkiste, P., Bai, F.: Simulating spatial cross-correlation in vehicular networks. In: *IEEE Vehicular Networking Conference (VNC)*, pp. 207–214 (2014). doi:[10.1109/VNC.2014.7013350](https://doi.org/10.1109/VNC.2014.7013350)