Semi–definite programming for the nearest circulant semi–definite matrix problem

SULIMAN AL-HOMIDAN

ABSTRACT. Positive semi–definite circulant matrices arise in many important applications. The problem arises in various applications where the data collected in a matrix do not maintain the specified structure as is expected in the original system. The task is to retrieve useful information while maintaining the underlying physical feasibility often necessitates search for a good structured approximation of the data matrix. This paper construct structured circulant positive semi–definite matrix that is nearest to a given data matrix. The problem is converted into a semi–definite programming problem as well as a problem comprising a semi–defined program and second-order cone problem. The duality and optimality conditions are obtained and the primal-dual algorithm is outlined. Some of the numerical issues involved will be addressed including unsymmetrical of the problem. Computational results are presented.

1. INTRODUCTION

We are interested in finding the nearest approximation of a given data matrix by a circulant positive semi–definite matrix. Circulant matrices are not symmetric, hence, difficulty arise in solving the problem since most methods for solving problems involving semi–definite matrices required the matrix to be symmetric. The circulant matrix is a matrix where each row is cyclic shift to the right in one place of the previous one. The study of circulant matrices and their use in realizing lists of complex numbers have produced many useful results [5].

Circulant matrices appear naturally in a variety of applications in mathematics and engineering. The significant applications of circulant matrices have attracted attention of researchers working in different disciplines. Several problems in digital signal processing, tensor decompositions and structural coupling require the computation of a positive definite circulant matrix that closely approximates a given matrix. Circulant matrices play an important role in investigation of tensor decompositions [15].

A general forms of coupling uses circulant matrices through a base substructure. These structural arise naturally in certain types of rotating machines. Circulant matrices arise also in stationary structures such as satellite antennae. The nature of rotationally periodic systems imposes a cyclic structure, which are circulant matrices [11].

Solving many systems requires approximating a data matrix to the nearest circulant positive semi–definite matrix. This data matrix supposes to be circulant but because of rounding or truncation errors incurred while evaluating the data matrix, it does not satisfy one or all conditions. Sometime because the data matrix is estimated from autocorrelation coefficients is often losses some or all its properties.

The problem was studied by Al-Homidan [1]. He solves the problem using sequential quadratic programming method. He also solves the problem using projection method

that was successfully used in solving similar approximation problems for distance matrices [3] and Hankel approximation [4]. The theory and properties of circulant matrix are discussed and presented in [7]. A semi–definite programming approach for the nearest correlation matrix problem has been studied by [6].

In the present paper we address the problem of finding the best approximation of a given data matrix by a positive semi–definite circulant matrix. The problem is solved using methods specialized for nonsymmetric matrices, it is formulated as an optimization problem with quadratic objective function and semi–definite programming constraints, several algorithms developed to solve this problem. More precisely, we have discussed the solution of the following problem:

Given an arbitrary data matrix $D \in \mathbb{R}^{n \times n}$, find the nearest positive semi–definite circulant matrix C to D, i. e.,

$$\text{minimize} \quad \phi = \frac{1}{2}\|C - D\|_F^2$$

(1.1) $$\text{subject to } C \in \mathbb{P} \cap \mathbb{C},$$

where $\mathbb{P}$ is the set of all $n \times n$ real symmetric or nonsymmetric positive semi–definite matrices, $\mathbb{P} = \{A : A \succeq 0, A \in \mathcal{M}^n\}$, and $A \succeq 0$ means positive semi–definite matrix. $\mathbb{C}$ is the set of all circulant matrices and $\mathcal{M}^n$ is the space of all $n \times n$ matrices. Here, $\|.\|_F$ denotes the Frobenius norm given by $\|A\|_F^2 = \langle A, A \rangle$ and $A^\dagger$ means conjugate transpose of A. The inner product $\langle A, B \rangle = \text{trace } A^T B = \sum_{ij} a_{ij} b_{ij}$.

The paper is organized as follows: Section 2 contains some characterization of circulant matrices and notations needed in the paper. In Section 3, we solve (1.1) using the von Neumann projection method. In Section 4, the problem is formulated as a semi–defined programming problem (SDP), then as a mixed SDP and second-order cone problem (SOC). The duality and optimality conditions for quadratic programs are presented in Section 5. The primal-dual algorithm is outlined in Section 6 with crossover criteria. Computational results are presented in Section 7.

2. CIRCULANT MATRIX CHARACTERIZATION AND NOTATION

Circulant matrices are diagonalized by the Fourier matrix F; i.e., $C = F^\dagger \Lambda F$, where the entries of F are given by

$$f_{jk} = \frac{1}{\sqrt{n}} e^{2\pi ijk/n}, \qquad 0 \leq j, k \leq n-1$$

and Λ is a diagonal matrix holding the eigenvalues of C; see, for instance, Davis [7]. The diagonal entries of Λ are given by

$$\lambda_k = \sum_{j=0}^{n-1} c_j e^{2\pi ijk/n}, \qquad k = 0, \ldots n-1.$$

To solve problem (1.1) we introduce a preliminary result that allows the matrix D be replaced by a certain circulant matrix C. To find the nearest matrix $C \in \mathbb{C}$ to D, we proceed as follows, if $D \in \mathbb{C}$ then $C = D$. Otherwise the solution is the near point in $\mathbb{C}$ to D, which is unique by convexity of $\mathbb{C}$. Now let $D = [d_{ij}]$ be an arbitrary $n \times n$ data matrix and $C = [c_{ij}]$ be an $n \times n$ circulant matrix. Then

(2.2) $$\langle D, C \rangle = c_1 \sum_{j=1}^{n} d_{jj} + \sum_{k=2}^{n-1} c_k \sum_{j=k}^{n} d_{(j-k+1)j} + \sum_{k=2}^{n-1} c_{n-k+1} \sum_{j=k}^{n} d_{j(j-k+1)}$$

Let B be the average of the diagonals of D in circulant form, that is,

$$(2.3) \qquad b_1 = \frac{1}{n} \sum_{j=1}^{n} d_{jj}, \quad b_k = \frac{1}{n} \{ \sum_{j=1}^{k-1} d_{(n-k+j)j} + \sum_{j=k}^{n} d_{(j-k+1)j} \}, \qquad k = 2, \ldots, n.$$

Then it is clear that B is circulant matrix and $\langle D, C \rangle = \langle B, C \rangle$ for all circulant matrices C. Define $B = P_{\mathbb{C}}(D)$ to be the orthogonal projection of D onto the subspace of circulant matrices $\mathbb{C}$. These observations prove the following lemma.

Lemma 2.1. *Let D be a given $n \times n$ data matrix and define B by (2.3). For all circulant matrices* $min\{\|D - C\| : C \text{ is circulant}\} = \|D - P_{\mathbb{C}}(D)\| = \|D - B\|$.

Several linear transformations between vector spaces and notations needed are defined. For a general rectangular matrix $M = [m_1 \ m_2 \ \ldots \ m_n] \in \mathcal{M}^n$, $v = \text{vec}(M)$ forms a vector from columns of M. The inverse mapping, vec^{-1}, and the adjoint mapping, vec^*, are given by $\text{Mat} = \text{vec}^{-1} = \text{vec}^*$, the adjoint formula following from $\langle \text{vec}(M), u \rangle = \langle M, \text{vec}^*(u) \rangle$. Mat and vec is an isometry.

If A is symmetric, then define $\text{svec}(A) = \mathbf{a}$, be the vector obtained columnwise from the upper triangular part of A, where the strictly upper triangular part is multiplied by $\sqrt{2}$.

We also need the operator $\mathfrak{C} : \mathbb{R}^n \longrightarrow \mathbb{C}$

$$\mathfrak{C}(\mathbf{c}) = \begin{bmatrix} c_1 & c_n & \cdots & c_2 \\ c_2 & c_1 & \cdots & c_3 \\ \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n-1} & \cdots & c_1 \end{bmatrix},$$

where $\mathbf{c} = \sqrt{n}[c_1, \ldots, c_n]^T$. Also, define the isometry operator $\text{cvec} : \mathbb{C} \longrightarrow \mathbb{R}^n$ as

$$\text{cvec}(\mathfrak{C}(\mathbf{c})) = \sqrt{n}[c_1, c_2, \cdots, c_n]^T = \mathbf{c}$$

for any $\mathbf{c} \in \mathbb{R}^n$. Now, for any $U, V \in \mathbb{C}$, there exist $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, such that $\mathfrak{C}(\mathbf{u}) = U$ and $\mathfrak{C}(\mathbf{v}) = V$. Note that $\mathbf{u}$ is equal to the first column of U multiplied by $\sqrt{n}$. Therefore, cvec is a linear operator satisfying the following: For any $U, V \in \mathbb{C}$

$$\langle U, V \rangle = \langle \mathfrak{C}(\mathbf{u}), \mathfrak{C}(\mathbf{v}) \rangle = \text{trace } \mathfrak{C}(\mathbf{u})^T \mathfrak{C}(\mathbf{v}) = n \sum_{i=1}^{n} u_i v_i$$

$$(2.4) \qquad = \text{cvec}(\mathfrak{C}(\mathbf{u}))^T \text{cvec}(\mathfrak{C}(\mathbf{v})) = \mathbf{u}^T \mathbf{v}$$

where $\mathbf{u} = \sqrt{n}[u_1, \cdots, u_n]^T$ and $\mathbf{v} = \sqrt{n}[v_1, \cdots, v_n]^T$. Also,

$$\|\mathfrak{C}(\mathbf{u}) - \mathfrak{C}(\mathbf{v})\|_F^2 = (\text{cvec}(\mathfrak{C}(\mathbf{u})) - \text{cvec}(\mathfrak{C}(\mathbf{v}))^T (\text{cvec}(\mathfrak{C}(\mathbf{u})) - \text{cvec } \mathfrak{C}(\mathbf{v}))$$

$$(2.5) \qquad = (\mathbf{u} - \mathbf{v})^T (\mathbf{u} - \mathbf{v}) = n \sum_{i=1}^{n} (u_i - v_i)^2.$$

Now $\mathfrak{C} = \text{cvec}^{-1}$ denote the inverse mapping into $\mathbb{C}$, i.e., the one-one mapping between $\mathbb{R}^n$ and $\mathbb{C}$. The adjoint operator $\mathfrak{C}^* = \text{cvec}$, since for any $U, V \in \mathbb{C}$

$$\langle \mathfrak{C}(\mathbf{u}), V \rangle = \text{trace } \mathfrak{C}(\mathbf{u})^T V = \mathbf{u}^T \mathbf{v} \qquad by \quad (2.4)$$

$$= \langle \mathbf{u}, \mathfrak{C}^*(V) \rangle = \langle \mathbf{u}, \text{cvec}(V) \rangle.$$

It is well known that circulant matrices commute under multiplication. The inverse of a non-singular circulant matrix is circulant. The product of two circulant matrices is

circulant. Moreover, the product of $\mathfrak{C}(\mathbf{a})$ and $\mathfrak{C}(\mathbf{b})$ is $\mathfrak{C}(\mathbf{c})$ where

$$c_j = \sum_{k=1}^{n} a_k b_{n+j-k}, \quad j = 1, \ldots, n,$$

where $b_{n+g} = b_g$, for $g \geq 0$. Let $\mathbf{e}$ denote a vector of unities, then $B\mathbf{e} = t\mathbf{e}$, where t is the row sum of B (the same for every row); and when B^{-1} exists, its row sum is $1/t$, [13]. These properties are very useful when constructing matrix derivative in Section 5 and 6.

3. THE PROJECTION ALGORITHM

Now, a projection method for solving problem (1.1) will be explained. This method is obtained from an oscillating projection method given by Dykstra [9] for finding the nearest distance from a fixed point to a convex sets intersection.

Defined in the previous section the two convex sets $\mathbb{P}$ is a convex cone with dimension n^2 and $\mathbb{C}$ is a subspace with dimension n.

When applying projection method to the circulant matrices, it is important to use the Frobenius norm, as expressed (1.1). To apply the algorithm, we need the projection maps $P_{\mathbb{P}}(.)$ and $P_{\mathbb{C}}(.)$. These are the maps from $\mathcal{M}^n$ on to $\mathbb{P}$ and $\mathbb{C}$.

To take the advantage of the isometry operator $\mathfrak{C}(.)$, we need D to be circulant. If we project D onto $\mathbb{C}$, we get $P_{\mathbb{C}}(D)$. The following lemma shows that the nearest circulant positive semi–definite matrix to D is exactly equal to the nearest circulant positive semi–definite matrix to $P_{\mathbb{C}}(D)$.

Lemma 3.2. *Let $\mathfrak{C}(\mathbf{x})$ be the nearest circulant positive semi–definite matrix to $P_{\mathbb{C}}(D)$, then $\mathfrak{C}(\mathbf{x})$ is the nearest circulant positive semi–definite matrix to D.*

Proof. If $P_{\mathbb{C}}(D)$ is positive semi–definite, then we are done. If not, then for any $\mathfrak{C}(\mathbf{x}) \in \mathbb{C}$, we have

$$\langle (\mathfrak{C}(x) - P_{\mathbb{C}}(D)), (P_{\mathbb{C}}(D) - D) \rangle = 0$$

since $P_{\mathbb{C}}(D)$ is the orthogonal projection of D. Thus,

$$\|\mathfrak{C}(\mathbf{x}) - D\|_F^2 = \|\mathfrak{C}(x) - P_{\mathbb{C}}(D)\|_F^2 + \|P_{\mathbb{C}}(D) - D\|_F^2.$$

This complete the proof since the second part of the above equation is constant. $\qquad\square$

The matrix D is not symmetric, hence, to find the projection of D onto $\mathbb{P}$, first, we find the symmetric and skew–symmetric parts of D, they are $B = \frac{D+D^T}{2}$; $F = \frac{D-D^T}{2}$, then, since B is symmetric, there exists a decomposition $B = U\Lambda U^T$, let $\Lambda^+ = \begin{pmatrix} \Lambda_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$, and $\Lambda_r = \text{diag}\,[\lambda_1, \lambda_2, \ldots, \lambda_r]$ is a diagonal matrix consist from the non–negative eigenvalues of Λ. Then, the projection maps $P_{\mathbb{P}}(D)$ and $P_{\mathbb{C}}(D)$ are

$$(3.6) \qquad\qquad P_{\mathbb{P}}(D) = U\Lambda^+ U^T + F \quad \text{and} \quad P_{\mathbb{C}}(D) = \mathfrak{C}(\mathbf{c})$$

where c_i are given in (2.3), and for more details see [10, 1].

The projection maps $P_{\mathbb{P}}(D)$ and $P_{\mathbb{C}}(D)$ given by (3.6) can now be carried out to apply the method. For $D \in \mathbb{R}^{n \times n}$, the algorithm start by $D^{(0)} = D$ and then the formula for the iterations is as follow

$$(3.7) \qquad\qquad D^{(k+1)} = D^{(k)} + (P_{\mathbb{C}}(P_{\mathbb{P}}(D^{(k)}))) - P_{\mathbb{P}}(D^{(k)}).$$

Both sequences $\{P_{\mathbb{P}}(D^{(k)})\}$ and $\{P_{\mathbb{C}}(P_{\mathbb{P}}(D^{(k)}))\}$ generated by (3.7) converged globally to the solution C^* of (1.1), [9].

4. MIXED-CONE FORMULATION

A brief linear direct approach for solving (1.1) will be introduced. This approach is obtained by formulating (1.1) first as an SDP problem then as a mixed SDP and second-order cone problem.

In view of Lemma 3.2, (1.1) is equivalent to the problem

$$p^* = \quad \text{minimize} \quad \frac{1}{2}\|\mathfrak{C}(\mathbf{x}) - P_{\mathbb{C}}(D)\|_F^2$$

(4.8)
$$\text{subject to} \quad \mathfrak{C}(x) \succeq 0.$$

Now, we have the following equivalences (for $\alpha \geq 0 \in \mathbb{R}$):

$$\|\mathfrak{C}(\mathbf{x}) - P_{\mathbb{C}}(D)\|_F^2 \leq \alpha \qquad \Leftrightarrow \qquad (\mathbf{x} - \mathbf{d})^T(\mathbf{x} - \mathbf{d}) \leq \alpha \quad \text{by} \quad (2.5)$$

$$\Leftrightarrow \quad \alpha - (\mathbf{x} - \mathbf{d})^T I(\mathbf{x} - \mathbf{d}) \geq 0$$

$$\Leftrightarrow \quad \begin{bmatrix} I & (\mathbf{x} - \mathbf{d}) \\ (\mathbf{x} - \mathbf{d}))^T & \alpha \end{bmatrix} \succeq 0$$

where $\mathfrak{C}(\mathbf{d}) = P_{\mathbb{C}}(D)$, the last inequality hold by Schur Complement. Since SDP requires the matrix constrains to be symmetric and since $\mathfrak{C}(\mathbf{x})$ is positive semi–definite if and only if $\frac{\mathfrak{C}(\mathbf{x})^T + \mathfrak{C}(\mathbf{x})}{2}$ is positive semi–definite. Hence, we have the following SDP problem:

$$\text{minimize} \quad \alpha$$

(4.9)
$$\text{subject to} \quad \begin{bmatrix} \frac{\mathfrak{C}(\mathbf{x})^T + \mathfrak{C}(\mathbf{x})}{2} & 0 & 0 \\ 0 & I & \mathbf{x} - \mathbf{d} \\ 0 & (\mathbf{x} - \mathbf{d})^T & \alpha \end{bmatrix} \succeq 0.$$

This SDP problem has dimensions $n + 1$ and $2n + 1$. The original problem (1.1) can also be formulated as a mixed SDP and second-order (or Lorentz) cone problem as follows: Since $\|\mathfrak{C}(\mathbf{x}) - P_{\mathbb{C}}(D)\|_F^2 = \|\mathbf{x} - \mathbf{d}\|_2^2$ we have the following equivalent problem:

$$\text{minimize} \quad \alpha$$

$$\text{subject to} \quad \frac{\mathfrak{C}(\mathbf{x})^T + \mathfrak{C}(\mathbf{x})}{2} \succeq 0$$

(4.10)
$$\begin{bmatrix} \alpha \\ \mathbf{x} - \mathbf{d} \end{bmatrix} \geq_Q 0.$$

The dimension of SOC is $n + 1$. The constraint $\geq_Q$ is the second-order cone constraint. For more details see [2].

These two equivalent problem will be solved in computational results section using public domain software packages. The main work per iteration for solving this problem is to form and solve the usually normal equations for the Newton search direction. There are many complications when forming and solving this system, since it is usually ill-conditioned at the solution.

5. DUALITY AND OPTIMALITY CONDITIONS

An equivalent problem to (4.8) and hence to (1.1) is:

(5.11)
$$p^* = \quad \text{minimize} \quad \frac{1}{2}\|\mathbf{x} - \mathbf{d}\|_2^2$$
$$\text{subject to} \quad \mathfrak{C}(\mathbf{x}) \succeq 0,$$

To obtain optimality conditions, we use a dual problem. Slater's condition (strict feasibility) holds for (5.11). This implies that we have strong duality with the Lagrangian dual

(5.12)
$$d^* = \max_{\mathfrak{C}(\mathbf{c}) \succeq 0} \min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{d}\|_2^2 - \text{trace}\, \mathfrak{C}(\mathbf{c})^T \mathfrak{C}(\mathbf{x}),$$

here, note that both $\mathfrak{C}(\mathbf{c})$ and $\mathfrak{C}(\mathbf{x})$ are not symmetric. Let $\phi = \frac{1}{2}\|\mathbf{x} - \mathbf{d}\|_2^2$, then

(5.13) $$\Delta_{\mathbf{x}}\phi = \sqrt{n}(\mathbf{x} - \mathbf{d}).$$

Similarly

(5.14) $$\Delta_{\mathbf{x}}(\operatorname{trace}\mathfrak{C}(\mathbf{c})^T\mathfrak{C}(\mathbf{x})) = \Delta_{\mathbf{x}}(\mathbf{c}^T\mathbf{x}) = \sqrt{n}\,\mathbf{c}.$$

Now, we change (5.12) to the Wolfe dual by noting that the inner problem is a convex unconstrained problem. Using (5.13) and (5.14) the optimal solution for (5.12) is characterized by stationary: $\mathbf{c} = \mathbf{x} - \mathbf{d}$. Hence, the objective of the dual can be expressed as:

$$\frac{1}{2}\|\mathbf{x} - \mathbf{d}\|_2^2 - \mathbf{c}^T\mathbf{x} \;=\; \frac{1}{2}(\mathbf{d}^T\mathbf{d} - \mathbf{x}^T\mathbf{x}).$$

Thus we obtain the equivalent dual problem:

(5.15) $$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|\mathbf{c} + \mathbf{d}\|_2^2 \\ \text{subject to} & \mathfrak{C}(\mathbf{c}) \succeq 0. \end{array}$$

Slater's condition is satisfied for both primal and dual programs. Therefore, we get the following optimality conditions.

Theorem 5.1. *The optimal values $p^* = d^*$ and the primal-dual pair $\mathbf{x}, \mathbf{c}$ are optimal for (5.11) and (5.15) if and only if*

$$\begin{array}{rcll} \mathfrak{C}(\mathbf{x}) & \succeq & 0 & \textit{(primal feasibility)} \\ \mathbf{c} & = & \mathbf{x} - \mathbf{d}, \ \ \mathfrak{C}(\mathbf{c}) \succeq 0 & \textit{(dual feasibility)} \\ \mathfrak{C}(\mathbf{x})^T\mathfrak{C}(\mathbf{c}) & = & 0 & \textit{(complementary slackness)} \end{array}$$

Proof. By convexity of the problem there is no duality gap, then the Slater's condition hold. $\qquad\square$

For our primal-dual interior-point algorithm, we use $\mathfrak{C}(\mathbf{x})^T\mathfrak{C}(\mathbf{c}) = \mu I$ perturbed complementary slackness. By substituting the primal and dual feasibility equations into the perturbed complementary slackness equation and obtain a *single bilinear equation* in $\mathbf{x}$ that characterizes optimality for the perturbed barrier problem:

(5.16) $$\begin{array}{rcl} F_\mu(\mathbf{x}) : \mathbb{R}^n &\to& \mathbb{R}^{n^2}, \\[4pt] F_\mu(\mathbf{x}) : &=& \operatorname{vec}\left[\{\mathfrak{C}(\mathbf{x}) - \mathfrak{C}(\mathbf{d})\}\mathfrak{C}(\mathbf{x})\right] - \mu\operatorname{vec} I = 0. \end{array}$$

Note that the original problem has n variables and the semi–definiteness constraint on $\mathfrak{C}(\mathbf{x})$. Therefore, the dual problem has n variables. Hence, dual based algorithms do not reduce the size of the problem and standard primal-dual based algorithms have $2n$ variables.

We solve (5.16) using an inexact Gauss-Newton method. Let linearize (5.16) then, we obtain a linear system for the search direction Δx:

(5.17) $$\begin{array}{rcl} F_\mu(\mathbf{x} + \Delta\mathbf{x}) &=& F_\mu(\mathbf{x}) + F_\mu'(\mathbf{x})(\Delta\mathbf{x}) + o(\|\Delta\mathbf{x}\|) \\ &=& F_\mu(\mathbf{x}) + \operatorname{vec}\{\mathfrak{C}(\mathbf{c})^T\mathfrak{C}(\Delta\mathbf{x}) + \mathfrak{C}(\Delta\mathbf{x})^T\mathfrak{C}(\mathbf{x})\} + o(\|\Delta\mathbf{x}\|). \end{array}$$

Hence,

$$F_\mu'(\mathbf{x})(\Delta\mathbf{x}) = \operatorname{vec}\{\mathfrak{C}(\mathbf{c})^T\mathfrak{C}(\Delta\mathbf{x}) + \mathfrak{C}(\Delta\mathbf{x})^T\mathfrak{C}(\mathbf{x})\}$$

Note that the adjoint of $\mathfrak{C}(\mathbf{c})B$ is $\frac{1}{2}(\mathfrak{C}(\mathbf{c})^T B + B^T\mathfrak{C}(\mathbf{c}))$; while the adjoint of $B\mathfrak{C}(\mathbf{x})$ is $\frac{1}{2}(\mathfrak{C}(\mathbf{x})B^T + B\mathfrak{C}(\mathbf{x})^T)$. Therefore, the adjoint $(F_\mu')^*$ for $\mathbf{w} \in \mathbb{R}^{n^2}$ is

$$(F_\mu')^*(\mathbf{w}) = \frac{1}{2}\operatorname{svec}\{[\mathfrak{C}(\mathbf{c})^T\operatorname{Mat}(\mathbf{w}) + \operatorname{Mat}(\mathbf{w})^T\mathfrak{C}(\mathbf{c})] + [\mathfrak{C}(\mathbf{x})\operatorname{Mat}(\mathbf{w})^T + \operatorname{Mat}(\mathbf{w})\mathfrak{C}(\mathbf{x})^T]\}.$$

Let simplify notation by defining the transformations with their adjoints,

$$\mathcal{X}(\cdot) = \text{vec}\left\{\mathfrak{C}(\cdot)^T\mathfrak{C}(\mathbf{x})\right\}; \quad \mathcal{X}^*(\cdot) = \frac{1}{2}\text{svec}\left\{\mathfrak{C}(\mathbf{x})\text{Mat}(\cdot)^T + \text{Mat}(.)\mathfrak{C}(\mathbf{x})^T\right\};$$

$$\mathcal{C}(\cdot) = \text{vec}\left\{\mathfrak{C}(\mathbf{c})^T\text{sMat}(\cdot)\right\}; \quad \mathcal{C}^*(\cdot) = \frac{1}{2}\text{svec}\left\{\mathfrak{C}(\mathbf{c})^T\text{Mat}(\cdot) + \text{Mat}(\cdot)^T\mathfrak{C}(\mathbf{c})\right\}.$$

Therefore,

(5.18) $$F'_\mu(\Delta\mathbf{x}) = \mathcal{C}(\Delta\mathbf{x}) + \mathcal{X}(\Delta\mathbf{x}), \quad \Delta\mathbf{x} \in \mathbb{R}^n;$$

(5.19) $$(F'_\mu)^*(\mathbf{w}) = \mathcal{C}^*(\mathbf{w}) + \mathcal{X}^*(\mathbf{w}), \quad \mathbf{w} \in \mathbb{R}^{n^2}.$$

$F'_\mu(\mathbf{x})(\Delta\mathbf{x})$ is a linear, full rank, overdetermined system. We use its least squares solution as the search direction $\Delta\mathbf{x}$ in our algorithm.

6. THE ALGORITHM

A primal-dual interior-exterior-point algorithm is developed using equation (5.16), i.e., we linearize to find the search direction, assuming that we start from a feasible point, using a linear least squares problem. First, we explain crossover criteria.

We work with an overdetermined nonlinear system; thus we use inexact Gauss–Newton approach, which has many advantages including full rank of the Jacobian at each iteration, optimality and a zero residual. A centering parameter σ is included in the perturbed complementary slackness equation, i.e. we replace μI by $\sigma\mu I$. Hence, once we are close enough to the optimum we set centering parameter σ to zero (crossover step) and use step lengths of one without backtracking to maintain positive definiteness. That is to say, there is a local neighborhood of quadratic convergence around each point on the central path and this neighborhood is not restricted to $\mathfrak{C}(\mathbf{c}), \mathfrak{C}(\mathbf{x}) \succ 0$. This allows fast quadratic convergence. Standard convergence results, e.g. [8], show that the Gauss-Newton method applied to $F_\mu(\mathbf{x}) = 0$ is locally quadratically convergent, since the Jacobian at the optimum is full column rank. We follow [4] to discuss several constants used to determine the region of quadratic convergence. The centering parameter is set to zero since we have a zero residual. Now,

(6.20)
$$\begin{aligned}
\|F'_\mu(\Delta\mathbf{x})\|_F &= \|\{\mathfrak{C}(\mathbf{c})^T\mathfrak{C}(\Delta\mathbf{x}) + \mathfrak{C}(\Delta\mathbf{x})^T\mathfrak{C}(\mathbf{x})\|_F \\
&\leq \|\mathfrak{C}(\mathbf{c})\|_F\|\mathfrak{C}(\Delta\mathbf{x}))\|_F + \|\mathfrak{C}(\Delta\mathbf{x})\|_F\|\mathfrak{C}(\mathbf{x})\|_F \\
&= \|\Delta\mathbf{x}\|_2(\|\mathfrak{C}(\mathbf{c})\|_F + \|\mathfrak{C}(\mathbf{x})\|_F) \\
&\leq \|\Delta\mathbf{x}\|_2\sqrt{\|\mathbf{c}\|_2^2 + \|\mathbf{x}\|_2^2} \\
&\quad \text{(by Cauchy} - \text{Schwartz inequality)},
\end{aligned}$$

the norm of the Jacobian bound is $\alpha = \sqrt{\|\mathbf{c}\|_2^2 + \|\mathbf{x}\|_2^2}$, and

$$\begin{aligned}
\|F'_\mu(\mathbf{x} - \bar{\mathbf{x}})(\Delta\mathbf{x})\|_F &= \|(\mathfrak{C}(\mathbf{x}) - \mathfrak{C}(\bar{\mathbf{x}}))\mathfrak{C}(\Delta\mathbf{x}) + \mathfrak{C}(\Delta\mathbf{x})(\mathfrak{C}(\mathbf{c}) - \mathfrak{C}(\bar{\mathbf{c}}))\|_F \\
&\leq \|\Delta\mathbf{x}\|_2(\|\mathbf{x} - \bar{\mathbf{x}}\|_2 + \|\mathbf{c} - \bar{\mathbf{c}}\|_2).
\end{aligned}$$

Hence, the Lipschitz constant is $\gamma = 1$.

If we assume nondegeneracy of optimum, i.e. $\mathbf{x}^*$ is unique and the smallest singular value satisfies $\sigma_{\min}(F'_\mu(\mathbf{x})) \geq \sqrt{K}$, for all $\mathbf{x}$ in an ϵ_1 neighborhood of $\mathbf{x}^*$, for some constant $K > 0$. Following [8, Page 223], define

$$\epsilon = \min\left\{\epsilon_1; \frac{K}{\alpha\gamma}\right\} = \min\left\{\epsilon_1; \frac{K}{\sqrt{\|\mathbf{x}^*\|_2^2 + \|\mathbf{c}^*\|_2^2}}\right\}.$$

Then quadratic convergence is guaranteed once the current iterate is in this ϵ neighborhood of the optimum. One possible heuristic for this is to start the crossover if

$$\rho\sigma_{\min}(F'_\mu(\mathbf{x})) \geq \|\Delta\mathbf{x}\|\sqrt{\|\mathbf{x}\|_2^2 + \|\mathbf{c}\|_2^2}, \quad \rho \in (0,1).$$

This bound is overly restrictive since it does not take into account the direction of the step. In our tests we started the crossover when the relative duality gap

$$\frac{\mathbf{x}^T\mathbf{c}}{\frac{1}{2}\|\mathfrak{C}(x) - D\|_F^2 + 1} < 0.5.$$

This simpler heuristic never failed to converge, and quadratic convergence was observed.

Equation (5.16) and the linearization (5.17) is used to develop the primal-dual interior-point Algorithm 6.2.

Algorithm 6.2. *(Primal-Dual Gauss-Newton via Conjugate Gradients)*
Input: Objective: data matrix D
Input: Tolerances: δ_1 (gap), δ_2 (crossover)
Initialization:

$$\mathfrak{C}(\mathbf{x}^0) \succ 0 \text{ and } \mathfrak{C}(\mathbf{c}^0) = P_\mathbb{C}(D) - \mathfrak{C}(\mathbf{x}^0) \succ 0$$

$$g = \text{trace } \mathfrak{C}(\mathbf{c}^0)\mathfrak{C}(\mathbf{x}^0); \ \mu = g/n; \ \sigma = 1; \ \phi = \frac{1}{2}\|\mathfrak{C}(\mathbf{x}) - P_\mathbb{C}(D)\|_F^2.$$

while $\min\{\frac{g}{\phi+1}, \phi\} > \delta_1$

>   **if** $\min\{\frac{g}{\phi+1}, \phi\} < \delta_2$ **then**
> $\sigma = 0$
>   **else**
>   *update σ*
>   **end if**
>   *Solve $F'_{\sigma\mu}(\mathbf{x})(\Delta\mathbf{x}) = -F_{\sigma\mu}(\mathbf{x})$ for $\Delta\mathbf{x}$*
>   *update $\mathfrak{C}(\mathbf{x}) = \mathfrak{C}(\mathbf{x}) + \alpha\,\mathfrak{C}(\Delta\mathbf{x}), \alpha > 0$,*
> $\mathfrak{C}(\mathbf{c}) = \mathfrak{C}(\mathbf{x}) - P_\mathbb{C}(D), (\mathfrak{C}(\mathbf{x}), \mathfrak{C}(\mathbf{c}) \succ 0)$
> $g = \text{trace } \mathfrak{C}(\mathbf{x})\mathfrak{C}(\mathbf{c}); \ \mu = g/n;$
> $\phi = \frac{1}{2}\|\mathfrak{C}(x) - P_\mathbb{C}(D)\|_F^2$

endwhile

This approach eliminate the primal and dual linear feasibility equations and work with the overdetermined nonlinear system. For each iteration, we find a new iterate by taking a step in the inexact Gauss-Newton search direction $\Delta\mathbf{x}$. The search direction is found using a conjugate gradient method, LSQR [12]. The cost of each conjugate gradients iteration is a matrix multiplication. We ensure that before crossover, the new iterate $\mathbf{x} + \alpha\Delta\mathbf{x}$ results in both $\mathfrak{C}(\mathbf{x})$, $\mathfrak{C}(\mathbf{c})$ sufficiently positive definite; then, after crossover we take $\alpha = 1$. By our construction, the iterates maintain both primal and dual feasibility.

7. Computational results

Numerical testing was performed to illustrate some features of our algorithm. The crossover using a step length of one has a significant effect on both the number of iterations and linear system condition. This allow the algorithm to run without maintain interiority and to use affine scaling.

The tests were done using MATLAB 7. We solve a set of problems using four approaches: (i) Algorithm 6.2, (ii) the SDP formulation (4.9), (iii) the mixed SDP and second order formulation (4.10) and (iv) projection method. We use the SDPT3-3.0 code [14] for

approaches (ii) and (iii). We applied all approaches to problems starting from small dense problems with $n = 10$ up to a large problems with $n = 500$.

We obtained results by applying the methods as follows: A positive definite circulant matrix C was formed randomly, then this matrix was perturbed by adding random noise matrix N to C, where elements of C vary between -0.50 and 0.50. The problem is to recover the matrix before the noise was added. In all these cases, we found the optimum to high accuracy, at least ten decimals, except for the projection method we stop with seven decimals of accuracy. Table 1, shows how close, in Frobenius norm for selected test problem, the optimal solution of each method, C^*, to the data matrix D.

TABLE 1. Performance comparison (norm $\|C^* - D\|_F$) among the four methods.

Methods\n	50	100	150	200	300	400	500
Algorithm 6.2	287.3321	843.3121	1214.1162	1743.7537	2843.1630	3638.2813	5134.4367
SDP	287.3321	843.3123	1214.1165	1743.7539	2843.1633	3638.2821	5134.4378
SDP & SOC	287.3322	843.3131	1214.1170	1743.7542	2843.1642	3638.2834	5134.4404
Pro.	287.3321	843.3121	1214.1161	1743.7537	2843.1629	3638.2812	5134.4366

The projection method is clearly the slowest. The results appear in Figure 1. We can see from the figure the correlation between the CPU time and size of the matrix $D = N + C$. The number of iterations required by Algorithm 6.2 remains essentially and efficiently constant from 9 to 17 iterations for all the test problems. The number of LSQR iterations required for small problems by one iteration of Algorithm 6.2 ranges from 7 to 18 iterations. However, this number increases for the mid-size problems from 4 to 33 iterations. Then, in the larger problems, the number of iterations ranges from 23 to 42 iterations. Note in Figure 1 the CPU time become very large for the projection method, the SDP and the mixed SDP when n is very large, therefore, we stop reporting

The CPU time to make the graph with in the figure frame. Figure 1 shows the total number of iterations by LSQR against the size of the matrix D.
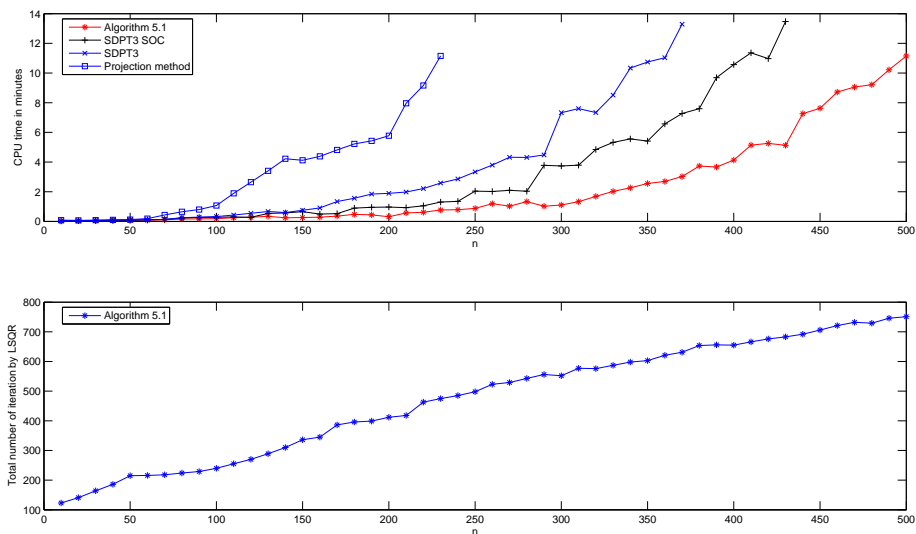

FIGURE 1. Above: Comparing all three approaches. Below: Total number of iterations by LSQR

We will explain now our strategy for choosing μ and σ with different values of s the step length. If the step is less than 0.001 then μ will be multiplied by 2.1 and $\sigma = 1$. If the step is less than 0.25 then we will multiply μ by 1.5 and $\sigma = 1$. If $s < 1$ then μ will be multiplied by 1.2 and $\sigma = 1 - 0.3\min(s, 1)$. However, if the step is larger than or equal to 1 then μ will be multiplied by 0.7 and $\sigma = 1 - 0.3s$. We get quadratic convergence after crossover, centering σ is set to 0 and steplength α is set to one. This step allows for warm starts. Then, long steps can be taken beyond the bounds of positivity. This will improves the convergence rates in all the numerical problems; the crossover starts in most tests at the $9 - 10$th iteration and converges in $2 - 4$ iterations.

The data matrix D is always dense. Even if some of the elements are zeros, these elements will disappear when we project to the circulant space $\mathbb{C}$. However, if any of anti-diagonals elements of D is all zeros, then these anti-diagonals elements can be eliminated and the dimension of the problem will be reduced by the number of zero anti-diagonals.

References

[1] Al-Homidan, S., *Low-Rank Approximation of Circulant Matrix to a Noisy Matrix* Submitted.

[2] Al-Homidan, S., *Mixed Semidefinite And Second-Order Cone Optimization Approach For The Circulant Matrix Approximation Problem* Submitted.

[3] Al-Homidan, S., *Structure method for solving the nearest Euclidean distance matrix problem*, J. Inequal. Appl., 2014, 2014:491, 7 pp.

[4] Al-Homidan, S., *Solving Hankel matrix approximation problem using semidefinite programming*, J. Comput. Appl. Math., **202** (2007), No. 2, 304–314

[5] Andrade, E., Arrieta, L. and Manzaneda, C. and Robbiano, M., *On the spectra of some g-circulant matrices and applications to nonnegative inverse eigenvalue problem*, Linear Algebra Appl., **590** (2020), 1–21

[6] Anjos, M., Higham, N., Takouda, P. and Wolkowicz, W., *A semdefinie programming approach for the nearest correlation matrix problem.* Technical report, Department of Combinatorics and Optimization, University of Waterloo, 2003

[7] Davis, P., *Circulant matrices*, John Wiley & Sons, New York-Chichester-Brisbane, 1979

[8] Dennis, J. E. and Schnabel, R. B, *Numerical methods for unconstrained optimization and nonlinear equations*, Corrected reprint of the 1983 original. Classics in Applied Mathematics, 16. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996

[9] Dykstra, R., *An algorithm for restricted least squares regression*, J. Amer. Stat. Assoc., **78** (1983), 839–842

[10] Higham, N., *Computing a nearest symmetric positive semi–definite matrix*, Linear Alg. Appl., **103** (1988), 103–118

[11] Olson, B., Shaw, S., Shi, C., Pierre, C. and Parker, R., *Circulant Matrices and Their Application to Vibration Analysis*, Appl. Mech. Rev., **66** (2014), No. 4, 41 pp.

[12] Paige, C. and Saunders, M., *LSQR: an algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, **8** (1982), No. 1, 43–71

[13] Searle, S., *On inverting circulant matrices*, Linear Algebra Appl., **25** (1979), 77–89

[14] Todd, M. J., Toh, K. C. and Tütüncü, R. H., *On the Nesterov-Todd direction in semidefinite programming*, SIAM J. Optim., **8** (1998), No. 3, 769–796

[15] Xu, J. J, Yang, Z. X. and Jiang, Y. L., *The best rank-one approximation of a symmetric tensor based on the block circulant matrix*, Oper. Res. Trans., **23** (2019), No. 1, 53–60

Department of Mathematics and Statistics
King Fahd University of Petroleum & Minerals
P. O. Box 119 Dhahran 31261, Saudi Arabia
Email address: `homidan@kfupm.edu.sa`