Mixed semidefinite and second-order cone optimization approach for the Hankel matrix approximation problem

Mohammed M. Alshahrani[1] Suliman S. Al-Homidan[2]

Abstract

Approximating the nearest positive semidefinite Hankel matrix in the Frobenius norm to an arbitrary data covariance matrix is useful in many areas of engineering, including signal processing and control theory. In this paper, interior point primal-dual path-following method will be used to solve our problem after reformulating it into different forms, first as a semidefinite programming problem, then into the form of a mixed semidefintie and second-order cone optimization problem. Numerical results, comparing the performance of these methods against the modified alternating projection method will be reported.

[1]Department of Mathematics, Dammam Teachers' College, P.O. Box 14262, Dammam 31424, SAUDI ARABIA **mmogib@awalnet.net.sa**

[2]Department of Mathematical Sciences, King Fahad University of Petroleum and Minerals, KFUPM Box 119, Dhahran 31261, SAUDI ARABIA **homidan@kfupm.edu.sa**

1 Introduction

In some application areas, such as digital signal processing and control theory, it is required to compute the closest, in some sense, positive semidefinite Hankel matrix, with no restriction on its rank, to a given data covariance matrix, computed from a data sequence. This problem was studied by Macinnes [18]. Similar problems involving structured covariance estimation were discussed in [15, 12, 24]. Related problems occur in many engineering and statistics applications [9].

The problem was formulated as a nonlinear minimization problem with positive semidefinite Hankel matrix as constraints in [2] and then was solved by l_2 Sequential Quadratic Programming (l_2 SQP) method. Another approach to deal with this problem was to solve it as a smooth unconstrained minimization problem [1]. Other methods to solve this problem or similar problems can be found in [18, 12, 15].

Our work is mainly casting the problem: first as a semidefinite programming problem and second as a mixed semidefinite and second-order cone optimization problem. A semidefinite programming (SDP) problem is to minimize a linear objective function subject to constraints over the cone of positive semidefinite matrices. It is a relatively new field of mathematical programming, and most of the papers on SDP were written in 1990s, although its roots can be traced back to a few decades earlier (see Bellman and Fan [7]). SDP problems are of great interest due to many reasons , e.g., SDP contains important classes of problems as special cases, such as linear and quadratic programming. Applications of SDP exist in combinatorial optimization, approximation theory, system and control theory, and mechanical and electrical engineering. SDP problems can be solved very efficiently in polynomial time by interior point algorithms [29, 31, 10, 5, 20].

The constraints in a mixed semidefinite and second-order cone optimization problem are constraints over the positive semidefinite and the second-order cones. Although the second-order cone constraints can be seen as positive semidefinite constraints, recent research has shown that it is more effecient to deal with mixed problems rather than the semidefinite programming problem. Nesterov et. al. [20] can be considered as the first paper to deal with mixed semidefinite and second-order cone optimization problems. However, the area was really brought to life by Alizadeh et al. [4] with the introduction of SDPPack, a software package for solving optimization problems from this class. The practical importance of second-order programming was demonstrated by Lobo et al. [17] and many subsequent papers. In [22] Sturm presented implementational issues of interior point methods for mixed SDP and SOCP problems in a unified framework. One class of these interior point methods is the primal-dual path-following methods. These methods are considered the most successful interior point algorithms for linear programming. Their extension from linear to semidefinite and then mixed problems has followed the same trends. One of the successful implementation of primal-dual path-following methods is in the software SDPT3 by Toh et al. [28, 25].

Similar problems, such as the problem of minimizing the spectral norm of a matrix was first formulated as a semidefinite programming problem in [29, 26]. Then, these problems and some others were formulated as a mixed semidefinite and second-order cone optimization problems [17, 3, 23]. None of these formulations exploited the special structure our problem has. For the purpose of exploiting the Hankel structure of the variable in this problem we will introduce an isometry operator, **hvec**, taking $n \times n$ Hankel matrices into $2n-1$ vectors. We will see later that using this operator gives our formulations an advantage over the others.

Before we go any further, we should introduce some notations. Throughout this paper, we will denote the set of all $n \times n$ real symmetric matrices by $\mathcal{S}_n$, the cone of the $n \times n$ real symmetric positive semidefinite matrices by $\mathcal{S}_n^+$ and the second-order cone of dimension k by $\mathcal{Q}_k$, and is defined as

$$\mathcal{Q}_k = \{\mathbf{x} \in I\!\!R^k : \|\mathbf{x}_{2:k}\|_2 \leq x_1\},$$

(also called Lorentz cone, ice cream cone or quadratic cone), where $\|.\|_2$ stands for the Euclidean distance norm defined as $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$, $\forall \mathbf{x} \in I\!\!R^n$. The set of all $n \times n$ real Hankel matrices will be denoted by $\mathcal{H}_n$. An $n \times n$ real Hankel matrix $H(\mathbf{h})$ has the following structure:

$$H(\mathbf{h}) = \begin{bmatrix} h_1 & h_2 & \cdots & h_n \\ h_2 & h_3 & \cdots & h_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_n & h_{n+1} & \cdots & h_{2n-1} \end{bmatrix}, \mathbf{h} \in I\!\!R^{2n-1}.$$

It is clear that $\mathcal{H}_n \subset \mathcal{S}_n$. The Frobenius norm is defined on $\mathcal{S}_n$ as follows:

$$\|U\|_F = \sqrt{U \bullet U} = \|\mathbf{vec}^T(U)\mathbf{vec}(U)\|_2, \ \forall\, U \in \mathcal{S}_n \tag{1.1}$$

Here $U \bullet U = \mathrm{trace}(U \cdot U) = \sum_{i,j}^n U_{i,j}^2$ and $\mathbf{vec}(U)$ stands for the vectorization operator found by stacking the columns of U together. The symbols $\succeq$ and $\geq_Q$ will be used to denote the partial orders induced by $\mathcal{S}_n^+$ and $\mathcal{Q}_k$ on $\mathcal{S}_n$ and $I\!\!R^k$, respectively. That is,

$$U \succeq V \quad \Leftrightarrow \quad U - V \in \mathcal{S}_n^+, \ \ \forall\, U,\ V \in \mathcal{S}_n$$

and

$$\mathbf{u} \geq_Q \mathbf{v} \Leftrightarrow \ \ \mathbf{u} - \mathbf{v} \in \ \mathcal{Q}_k, \ \ \forall\, \mathbf{u},\ \mathbf{v}\ \in I\!\!R^k.$$

The statement $\mathbf{x} \geq 0$ for a vector $\mathbf{x} \in I\!\!R^n$ means that each component of $\mathbf{x}$ is nonnegative. We use I and $\mathbf{0}$ for the identity and zero matrices. The dimensions of these matrices can be discerned from the context.

Our problem in mathematical notation can, now, be formulated as follows: Given a data matrix $F \in \mathbb{R}^{n\times n}$, find the nearest positive semidefinite Hankel matrix $H(\mathbf{h})$ to F such that $\|F - H(\mathbf{h})\|_F$ is minimal. Thus, we have the following optimization problem:

$$\begin{aligned} \text{minimize} \;\; & \|F - H(\mathbf{h})\|_F \\ \text{subject to} \;\; & H(\mathbf{h}) \in \mathcal{H}_n, \\ & H(\mathbf{h}) \succeq 0. \end{aligned} \tag{1.2}$$

It is worth describing the alternating projection method briefly; since this method is the most accurate, and converges to the optimal solution globally. However, the rate of convergence is slow. That makes it a good tool to provide us with accurate solutions against which we can compare the results obtained by the interior point methods. For these reason we devote Section 2 to the projection method. A brief description of semidefinite and second-order cone optimization problems along with reformulations of problem (1.2) in the form of the respective class will be given in Sections 3 and 4, respectively. Numerical results, showing the performance of the projection method against the primal-dual path-following method acting on our formulations, will be reported in Section 5.

2 The projection Method

The method of successive cyclic projections onto closed subspaces C_i's was first proposed by von Neumann [21] and independently by Wiener [30]. They showed that if, for example, C_1 and C_2 are subspaces and D is a given point, then the nearest point to D in $C_1 \cap C_2$ could be obtained by the following algorithm:

Alternating Projection Algorithm
Let $X_1 = D$
for $k = 1, 2, 3, \ldots$
$\quad X_{k+1} = P_1(P_2(X_k))$.

X_k converges to the near point to D in $C_1 \cap C_2$, where P_1 and P_2 are the orthogonal projections on C_1 and C_2, respectively. Dykstra [11] and Boyle and Dykstra [8] modified von Neumann's algorithm to handle the situation when C_1 and C_2 are replaced by convex sets. Other proofs and connections to duality along with applications were given in Han [16]. These modifications were applied in [14] to find the nearest Euclidean distance matrix to a given data matrix. The modified Neumann's algorithm when applied to (1.2) yields the following algorithm, called the Modified Alternating Projection Algorithm: Given a data matrix F, we have:

Let $F_1 = F$
for $j = 1, 2, 3, \ldots$
$\quad F_{j+1} = F_j + [P_S(P_H(F_j)) - P_H(F_j)]$

Then $\{P_H(F_j)\}$ and $P_S(P_H(F_j))$ converge in Frobenius norm to the solution. Here, $P_H(F)$ is the orthogonal projection onto the subspace of Hankel matrices $\mathcal{H}_n$. It is simply setting each antidiagonal to be the average of the corresponding antidiagonal of F. $P_S(F)$ is the projection of F onto the convex cone of positive semidefinite symmetric matrices. One finds $P_S(F)$ by finding a spectral decomposition of F and setting the negative eigenvalues to zero.

3 Semidefinite Programming Approach

The semidefinite programming (SDP) problem in primal standard form is:

$$\begin{aligned} (P) \quad & \min_X \ C \bullet X \\ & \text{s. t.} \ \ A_i \bullet X = b_i, \quad i = 1, \cdots, m \\ & \qquad\ X \succeq 0, \end{aligned} \tag{3.3}$$

where all $A_i, C \in \mathcal{S}_n, b \in I\!\!R^m$ are given, and $X \in \mathcal{S}_n$ is the variable. This optimization problem (3.3) is a convex optimization problem since its objective and constraint are convex. The dual problem of (3.3) is

$$\begin{aligned} (D) \quad & \max_{\mathbf{y}} \ \mathbf{b}^T \mathbf{y} \\ & \text{s. t.} \\ & \qquad \sum_{i=1}^m y_i A_i \preceq C, \end{aligned} \tag{3.4}$$

where $\mathbf{y} \in I\!\!R^m$ is the variable. Although (3.3) and (3.4) seem to be quite specialized, it includes, as we said before, many important problems as special cases. It also appears in many applications. One of these applications is problem (1.2) as we will show now.

For this purpose, we should introduce the following theorem:

THEOREM 3.1 (SCHUR COMPLEMENT)

If

$$M = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix},$$

where $A \in \mathcal{S}_n^+$ *and* $C \in \mathcal{S}_n$, *then the matrix* M *is positive (semi)definite if and only if the matrix* $C - B^T A^{-1} B$ *is positive (semi)definite.* □

This matrix $C - B^T A^{-1} B$ is called the schur complement of A in M.

Letting $\|F - H(\mathbf{h})\|_F^2 \le t$, t is a nonnegative real scalar and noting that:

$$\|F - H(\mathbf{h})\|_F^2 = \mathbf{vec}^T(F - H(\mathbf{h}))\mathbf{vec}(F - H(\mathbf{h})),$$

we have:

$$\begin{aligned} & \mathbf{vec}^T(F - H(\mathbf{h}))\mathbf{vec}(F - H(\mathbf{h})) \le t \\ \Leftrightarrow\ & t - \mathbf{vec}^T(F - H(\mathbf{h})) I \mathbf{vec}(F - H(\mathbf{h})) \ge 0 \\ \Leftrightarrow\ & \begin{bmatrix} I & \mathbf{vec}(F - H(\mathbf{h})) \\ \mathbf{vec}^T(F - H(\mathbf{h})) & t \end{bmatrix} \succeq 0. \end{aligned}$$

The last equivalence is a direct application of Theorem 3.1. Thus, problem (1.2) can be rewritten as

$$
\begin{aligned}
(SDV) \quad & \min \ t \\
& \text{s.t.} \\
& \begin{bmatrix} t & 0 & 0 \\ 0 & H(\mathbf{h}) & 0 \\ 0 & 0 & V \end{bmatrix} \succeq 0,
\end{aligned}
\tag{3.5}
$$

where

$$
V = \begin{bmatrix} I & \mathbf{vec}(F - H(\mathbf{h})) \\ \mathbf{vec}^T(F - H(\mathbf{h})) & t \end{bmatrix}
$$

which is an SDP problem in the dual form (3.4) with dimensions $2n$ and $n^2 + n + 2$, SDP problem (3.5) is very large even for a small data matrix F. For example, a 50×50 matrix F will give rise to a problem with dimensions 100 and 2552, hence solving (1.2) using formulation (3.5) is not efficeint. Furthermore, we do not exploit the structure of $H(\mathbf{h})$ being Hankel. This discussion leads us to think of another way of formulation that produces an SDP problem with reasonable dimensions and exploits the Hankel structure of $H(\mathbf{h})$. This can be done by means of the following isometry operator:

Definition 3.1

Let $\mathbf{hvec} : \mathcal{H}_n \longrightarrow I\!R^{2n-1}$ *be defined as*

$$\mathbf{hvec}(U) = [u_{1,1} \ \sqrt{2}u_{1,2} \ \cdots \ \sqrt{n-1}u_{1,n-1} \ \sqrt{n}u_{1,n} \ \sqrt{n-1}u_{2,n} \ \cdots \ \sqrt{2}u_{n-1,n} \ u_{n,n}]^T$$

for any $U \in \mathcal{H}_n$.

One can easily show that **hvec** is a linear operator from the set of all $n \times n$ real Hankel matrices to $I\!R^{2n-1}$. The following theorem gives us some characterizations of **hvec**.

THEOREM 3.2

For the operator **hvec** , *defined in (3.1), the following conditions hold: For any* $U, V \in \mathcal{H}_n$

1. $U \bullet U = \boldsymbol{hvec}^T(U)\boldsymbol{hvec}(U)$.
2. $\|U - V\|_F^2 = \boldsymbol{hvec}^T(U - V)\boldsymbol{hvec}(U - V)$. $\square$

Proof:

Part 1 is clear from the definition of the **hvec** operator. Part 2 is a consequece of part 1. $\triangle$

Part 1 implies that **hvec** is an isometry. We cannot take any advantage of this theorem unless F is Hankel. Of course, we can think of projecting F onto $\mathcal{H}_n$ using the orthogonal projection in Section 2 to get a Hankel matrix, say $\hat{F}$. But, is the nearest Hankel positive semidefinite matrix to $\hat{F}$, the nearest to F? The following proposition gives the answer:

PROPOSITION 3.1 Let $\hat{F}$ be the orthogonal projection of F onto $\mathcal{H}_n$ and let $H(\mathbf{h})$ be the nearest Hankel positive semidefinite matrix to $\hat{F}$, then $H(\mathbf{h})$ is so for F. $\square$

Proof:

If $\hat{F}$ is positive semidefinite, then we are done. If not, then for any $T \in \mathcal{H}_n$, we have

$$(F - \hat{F})^T \bullet (\hat{F} - T) = 0$$

since $\hat{F}$ is the orthogonal projection of F. Thus,

$$\|F - T\|_F^2 = \|F - \hat{F}\|_F^2 + \|\hat{F} - T\|_F^2.$$

As a consequence of this proposition, we have the following problem equivalent to (1.2):

$$\begin{array}{rl} \text{minimize} & \|\hat{F} - H(\mathbf{h})\|_F \\ \text{subject to} & H(\mathbf{h}) \in \mathcal{H}_n, \\ & H(\mathbf{h}) \succeq 0. \end{array} \tag{3.6}$$

3.1 Formulation I (SDH)

From Theorem 3.1, we have the following equivalences (for $t \geq 0 \in \mathbb{R}$):

$$\begin{array}{rl} & \|\hat{F} - H(\mathbf{h})\|_F^2 \leq t \\ \Leftrightarrow & \mathbf{hvec}^T(\hat{F} - H(\mathbf{h}))\mathbf{hvec}(\hat{F} - H(\mathbf{h})) \leq t \ \text{ by Theorem 3.2} \\ \Leftrightarrow & t - \mathbf{hvec}^T(\hat{F} - H(\mathbf{h}))I\mathbf{hvec}(\hat{F} - H(\mathbf{h})) \leq 0 \\ \Leftrightarrow & \begin{bmatrix} I & \mathbf{hvec}(\hat{F} - H(\mathbf{h})) \\ \mathbf{hvec}^T(\hat{F} - H(\mathbf{h})) & t \end{bmatrix} \succeq 0 \ \text{ by Theorem 3.1.} \end{array}$$

Hence, we have the following SDP problem:

$$\begin{array}{ll} (SDH) & \min \ t \\ & \text{s.t.} \\ & \begin{bmatrix} t & 0 & 0 \\ 0 & H(\mathbf{h}) & 0 \\ 0 & 0 & \hat{V} \end{bmatrix} \succeq 0, \end{array} \tag{3.7}$$

where

$$\hat{V} = \begin{bmatrix} I & \mathbf{hvec}(F - H(\mathbf{h})) \\ \mathbf{hvec}^T(F - H(\mathbf{h})) & t \end{bmatrix}.$$

This SDP problem has dimensions $2n$ and $3n+1$ which is far better than (3.5).

3.2 Formulation II (SDQ)

Another way for formulating (1.2) is through the definition of the Frobenius norm being a quadratic function. Indeed,

$$\|F - H(\mathbf{h})\|_F^2 = \mathbf{y}^T P\mathbf{y} + 2\mathbf{q}^T\mathbf{y} + r,$$

where

$$\mathbf{y} = [h_1\ h_2\ \cdots\ h_{2n-1}]^T,$$

$$P = \mathrm{diag}([1\ 2\ \cdots\ n\ \cdots 2\ 1]),$$

$$\mathbf{q}_k = -\sum_{\substack{i,j=1 \\ i+j=k+1}}^{n} F(i,j), \quad k = 1, 2, \cdots 2n-1 \ \text{ and}$$

$$r = \|F\|_F^2.$$

Now, we have for a nonnegative real scalar t

$$\begin{aligned}
& \|F - H\|_F^2 \le t \\
\Leftrightarrow\ & \mathbf{y}^T P\mathbf{y} + 2q^T\mathbf{y} + r \le t \\
\Leftrightarrow\ & (P^{1/2}\mathbf{y})^T(P^{1/2}\mathbf{y}) + 2q^T\mathbf{y} + r \le t \\
\Leftrightarrow\ & t - 2q^T\mathbf{y} - r - (P^{1/2}\mathbf{y})^T I (P^{1/2}\mathbf{y}) \ge 0 \\
\Leftrightarrow\ & \begin{bmatrix} I & (P^{1/2}\mathbf{y}) \\ (P^{1/2}\mathbf{y})^T & t - 2q^T\mathbf{y} - r \end{bmatrix} \succeq 0.
\end{aligned}$$

Hence, we have the following SDP problem:

$$
\begin{aligned}
(SDQ) \quad & \min \ t \\
& \text{s.t.} \\
& \begin{bmatrix} t & 0 & 0 \\ 0 & H(\mathbf{h}) & 0 \\ 0 & 0 & Q \end{bmatrix} \succeq 0,
\end{aligned}
\tag{3.8}
$$

where

$$
Q = \begin{bmatrix} I & (P^{1/2}\mathbf{y}) \\ (P^{1/2}\mathbf{y})^T & t - 2q^T\mathbf{y} - r \end{bmatrix},
$$

This SDP problem is of dimenstions $2n$ and $3n+1$. Although problem (3.8) has the same dimentions as problem (3.7), it is less efficient to solve it over the positive semidefinite cone $\mathcal{S}_n^+$, especially when F is large in size. In practice, as we will see in Section 5, it has been found that the performance of this formulation is poor. The reason for that is the matrix P being of full rank and hence the system is badly conditioned. A more efficient interior point method for this formulation can be developed by using Nesterov and Nemirovsky formulation as a problem over the second-order cone (see [19] Section 6.2.3). This what we will see in the next section.

The last formulation seems to be straight forward, but it was found that using this formulation to solve similar problems was not a good idea. The reasons for that will be discussed in the following section when we talk about second-order cone programming. This fact about SDQ formulation will be clear in Section 5 when we use it to solve numerical examples with $n > 50$. We think also SDV formulation is not good enough to compete with other formulation even with the projection method. This is simply due to the fact that the amount of work per one iteration of interior-point methods that solve SDV fomulation is $\mathcal{O}(n^6)$, where n in the dimension of F. This disappointing fact makes using SDV formulation to solve (1.2) a waste of time. This leaves us with SDH formulation from which we expect good performance; since it does not have the illness of SDQ nor the huge size of SDV.

4 Mixed Semidefinite and Second-Order Cone Approach:

The primal mixed semidefinite, second-order and linear problem SQLP is of the form:

$$\begin{aligned} (P') \quad \min \quad & C_S \bullet X_S + C_Q^T X_Q + C_L^T X_L \\ \text{s.t.} \quad & (A_S)_i \bullet X_S + (A_Q)_i^T X_Q + (A_L)_i^T X_L = b_i, \quad i = 1, \cdots, m \\ & X_S \succeq 0, X_S \geq_Q 0, X_L \geq 0, \end{aligned} \tag{4.9}$$

where $X_S \in \mathcal{S}_n$, $X_Q \in I\!R^k$ and $X_L \in I\!R^{n_L}$ are the variables. C_S, $(A_S)_i \in \mathcal{S}_n$, $\forall i$ C_Q, $(A_Q)_i \in I\!R^k$ $\forall i$ and C_L, $(A_L)_i \in I\!R^{n_L}$ $\forall i$ are given data. Each of the three inequalities has a different meaning: $X_S \succeq 0$ means, as we have seen, that $X_S \in \mathcal{S}_n^+$, $X_S \geq_Q 0$ means that $X_Q \in \mathcal{Q}_k$ and $X_L \geq 0$ means that each component of X_L is nonnegative. It is possible that one or more of the three parts of (4.9) is not present. If the second-order part is not present,

then (4.9) reduces to the ordinary SDP (3.3) and if the semidefinite part is not present, then (4.9) reduces to the so-called convex quadratically constrained linear programming problem.
The standard dual of (4.9) is:

$$
\begin{aligned}
(D') \quad & \max \ \mathbf{b}^T\mathbf{y} \\
& \text{s.t.} \\
& \quad \sum_{i=1}^{m} y_i (A_S)_i \preceq C_S \\
& \quad \sum_{i=1}^{m} y_i (A_Q)_i \leq_Q C_Q \\
& \quad \sum_{i=1}^{m} y_i (A_L)_i \leq C_L .
\end{aligned}
\tag{4.10}
$$

Here, $\mathbf{y} \in I\!R^m$ is the variable.

In our setting, we may drop the third part of the constraints in (4.9) and its dual (4.10), since we do not have explicit linear constraints. One natural claim can be made here: In (1.2) the objective function can be recast as a dual SQLP in three different ways.

4.1 Formulation III (SQV)

One way to define $\|F - H(\mathbf{h})\|_F$ is

$$\|F - H(\mathbf{h})\|_F = \|\mathbf{vec}(F - H(\mathbf{h}))\|_2.$$

So, if we put $\|F - H(\mathbf{h})\|_F \leq t$ for $t \in I\!R^+$, then by the definition of the second-order cone, we have

$$\begin{bmatrix} t \\ \mathbf{vec}(F - H(\mathbf{h})) \end{bmatrix} \in \mathcal{Q}_{1+n^2}$$

Hence, we have the following reformulation of (1.2):

$$\begin{array}{lll} (SQV) & \min & t \\ & \text{s.t.} & \begin{bmatrix} t & 0 \\ 0 & H(\mathbf{h}) \end{bmatrix} \succeq 0 \\ & & \begin{bmatrix} t \\ \mathbf{vec}(F - H(\mathbf{h})) \end{bmatrix} \geq_Q 0. \end{array} \tag{4.11}$$

4.2 Formulation IV (SQQ)

The second definition is as introduced in Subsection 3.2, i.e.,

$$\|F - H(\mathbf{h})\|_F^2 \;\; = \mathbf{y}^T P \mathbf{y} + 2\mathbf{q}^T \mathbf{y} + r$$

Hence, we have the following equivalent problem to (1.2)

$$\begin{array}{ll} \min & \mathbf{y}^T P \mathbf{y} + 2\mathbf{q}^T \mathbf{y} + r \\ \text{s.t.} & H(\mathbf{h}) \;\in\; \mathcal{H}_n, \\ & H(\mathbf{h}) \succeq 0. \end{array} \tag{4.12}$$

But

$$\mathbf{y}^T P \mathbf{y} + 2\mathbf{q}^T \mathbf{y} + r = \|P^{1/2}\mathbf{y} + P^{-1/2}\mathbf{q}\|_2^2 + r - \mathbf{q}^T P^{-1}\mathbf{q}$$

Now, we minimize $\|F - H(\mathbf{h})\|_F^2$ by minimizing $\|P^{1/2}\mathbf{y} + P^{-1/2}\mathbf{q}\|_2$. Thus we have the following problem:

$$
\begin{array}{lll}
(SQQ) & \min & t \\
& \text{s.t.} & \begin{bmatrix} t & 0 \\ 0 & H(\mathbf{h}) \end{bmatrix} \succeq 0 \\
& & \begin{bmatrix} t \\ P^{1/2}\mathbf{y} + P^{-1/2}\mathbf{q} \end{bmatrix} \geq_Q 0,
\end{array}
\tag{4.13}
$$

where $t \in I\!R^+$ is as before. Again, this problem is in the form of problem (4.10). Here, the difference between this form and SQV is in the second-order cone constraint since the SDP part is the same as SQV. The dimension of the second-order cone in SQV is $1 + n^2$ and in SQQ is just $2n$, which makes us expect less efficiency in practice when we work with SQV. The optimal value of SQV is the same as that of problem (1.2), whereas the optimal values of SQQ (4.13) and (4.12) are equal up to a constant. Indeed, the optimal value of (4.12) is equal $(\rho^*)^2 + r - \mathbf{q}^T P^{-1}\mathbf{q}$, where ρ^* is the optimal value of (4.13). One might notice that we did not talk about the constraint of $H(\mathbf{h})$ being Hankel. This is because the Hankel structure of $H(\mathbf{h})$ is embedded in the other constraints.

4.3 Formulation V (SQH)

The last formulation will take advantage of the Hankel structure of $H(\mathbf{h})$ explicitly. The vectorization operator **hvec** on Hankel matrices, introduced in Section 3 will be used. From Theorem 3.2, we have the following:

$$\|\hat{F} - H(\mathbf{h})\|_F = \|\mathbf{hvec}(\hat{F} - H(\mathbf{h}))\|_2,$$

where $\hat{F} = P_H(F)$, so that we have the following problem:

$$\begin{array}{lll} (SQH)\min & t & \\ \text{s.t.} & \begin{bmatrix} t & 0 \\ 0 & H(\mathbf{h}) \end{bmatrix} \succeq 0 & \\ & \begin{bmatrix} t \\ \mathbf{hvec}(\hat{F} - H(\mathbf{h})) \end{bmatrix} \geq_Q 0. & \end{array} \tag{4.14}$$

The dimension of the second-order cone in this form is $2n$, the same as that of SQQ. Furthermore, the optimal solution is the same as that of (1.2).

Table 1 shows the dimensions of the semidefinite part (SD part) and the second-order cone part (SOC part) for each formulation. For the formulations SDH and SDQ, the second-order cone part is not applicable, so the cell in the table corresponding to that is left blank.

In practise, we expect that the mixed formulations are more effecient than the SDP-only formulations, especially the SQQ and SQH which have second-order cone constraint of least dimension. Since, as we have seen, interior point methods for SOCP have better worst-case complexity than an SDP method. However, SDH has a less SDP dimension with no illness such as that SDQ has, which makes SDH a better choise among other SDP. This is due to the

Formulation	SD part	SOC part
SDV	$2n \times (n^2 + n + 2)$	
SDH	$2n \times (3n + 1)$	
SDQ	$2n \times (3n + 1)$	
SQV	$2n \times (n + 1)$	$n^2 + 1$
SQQ	$2n \times (n + 1)$	$2n$
SQH	$2n \times (n + 1)$	$2n$

Table 1: Problem dimensions

economical vectorization operator **hvec**. Indeed, practical experiments show a competitive behaviour of SDH to SQQ and SQH (see Section 5).

5 Numerical Results

We will now present some numerical results comparing the performance of the methods described in Sections 2, 3 and 4. The first is the projection method and the second is the interior-point primal-dual path-following method employing the NT-direction. The latter was used to solve five different formulations of the problem.

A Matlab code was written to implement the projection method. The iteration is stopped when $\|P_S(P_H(F_j)) - P_H(F_j)\|_F \leq 10^{-8}$.

For the other methods, the software SDPT3 ver. 3.0 [27, 25] was used because of its numerical stability [13] and its ability to exploit sparsity very efficiently. The default starting iterates in

SDPT3 were used throughout with the NT-direction. The choice of the NT-direction came after some preliminary numerical results. The other direction is HKM-direction which we found less accurate, although, faster than the NT-direction. However, the difference between the two in speed is not of significant importance.

The problem was converted into the five formulations described in Sections 3 and 4. A Matlab code was written for each formulation. This code formulates the problem and passes it through to SDPT3 for a first time. A second run is done with the optimal iterate from the first run being the initial point. This process is repeated until no progress is detected. This is done when the relative gap:

$$\frac{P\text{-}D}{\max\{1,(P+D)/2\}}$$

of the current run is the same as the preceding one. (Here, P and D denote the optimal and the dual objective values, respectively.)

Our numerical experiments were carried out on eleven randomly generated square matrices with different sizes, namely: 10, 30, 50, 100 and 200, two for each size and one of size 400. Each matrix is dense and its entries vary between -100 and 100 exclusive.

All numerical experiments in this section were executed in Matlab 6.1 on a 1.7GHz Pentium IV PC with 256 MB memory running MS-Windows 2000 Professional.

Size	Time (sec.)					
	Pro.	SDH	SDQ	SQH	SQQ	SQV
10	2	2	1	1	1	1
	9	1	1	1	1	1
30	11	5	4	3	4	2
	14	5	4	2	2	2
50	117	10	12	5	7	5
	30	11	11	4	3	5
100	61	53	64	28	20	28
	1003	48	42	22	25	21
200	16239	389	284	324	322	284
	4883	355	420	255	268	230
400	36556	4970	3913	3775	4098	2505

Table 2: Performance comparison (time) among the projection method and the path-following method with the formulations SDH, SDQ, SQH, SQQ and SQV.

Table 2 compares the CPU time. We notice that the consumed time gets larger more rapidly in the projection method with the size of the data matrix F. An obvious remark is that the projection method is the slowest; indeed, it is at least seven times slower than the slowest of the five formulations of the path-following method. However, the difference in time between the five formulations is not big enough to have a significant importance.

Another clear advantage is in terms of number of iterations as shown in Table 3. Although the amount of work in each iteration is different for each method, it is still fair to consider it to be a

Size	Iterations					
	Pro.	SDH	SDQ	SQH	SQQ	SQV
10	1253	16	18	14	14	11
	6629	18	17	14	14	11
30	1215	34	32	35	47	24
	1443	33	33	29	29	20
50	4849	32	41	25	36	24
	1295	32	42	22	18	26
100	504	34	45	27	19	26
	8310	33	28	23	26	20
200	22672	31	22	33	31	25
	6592	28	32	23	27	22
400	7870	28	25	26	26	18

Table 3: Performance comparison (number of iterations) among the projection method and the path-following method with the formulations SDH, SDQ, SQH, SQQ and SQV.

comparison factor.

Table 4 shows how close, in Frobenius norm, the optimal solution of each method, $H(\mathbf{h})^*$, to the data matrix F. The projection and the path-following methods with the formulation SDH, SQH and SQQ gave the same result to some extent. The formulation SDQ couldn't cope with the others as the problem size gets larger. The poor performance of this formulation is due to the matrix P being of full rank. The formulation SQV is less accurate than SDH, SQH and SQQ which is reasonable especially if we notice that the dimension of the second-order cone in this formualtion is $1 + n^2$. (see Table 1)

Size	Norm					
	Pro.	SDH	SDQ	SQH	SQQ	SQV
10	96.6226	96.6226	96.6226	96.6226	96.6226	96.6226
	94.8320	94.8320	94.8320	94.8320	94.8320	94.8320
30	307.9339	307.9339	307.9406	307.9339	307.9339	307.9339
	327.6784	327.6784	327.6784	327.6784	327.6784	327.6784
50	494.3805	494.3805	494.5038	494.3805	494.3805	494.3805
	497.4383	497.4383	497.6330	497.4383	497.4383	497.4383
100	991.8832	991.8832	994.8612	991.8832	991.8832	991.8833
	997.4993	997.4993	998.8048	997.4993	997.4993	997.4994
200	1986.9397	1986.9398	1990.0924	1986.9402	1986.9402	1986.9414
	1994.8409	1994.8410	1998.6048	1994.8410	1994.8410	1994.8418
400	3998.4967	3998.5047	4001.9242	3998.5007	3998.5007	3998.6166

Table 4: Performance comparison (norm $\|H(\mathbf{h})^* - F\|_F$) among the projection method and the path-following method with the formulations SDH, SDQ, SQH, SQQ and SQV.

To summarize the above discussion, we introduce Table 5. This table gives a measure to how close the optimal solutions of SDH, SDQ, SQH, SQQ and SQV from that of the projection method which is the most accurate. The error is computed simply by evaluating the difference between the norm $\|H(\mathbf{h})^* - F\|_F$ of the projection and the norm obtained by the different formulations of the path-following method.

Conclusion:

We conclude this paper by addressing few remarks. The pro-

Size	Error				
	SDH	SDQ	SQH	SQQ	SQV
10	6.3×10^{-9} 6.4×10^{-9}	3.4×10^{-9} 3.2×10^{-8}	6.1×10^{-9} 3.6×10^{-8}	6.1×10^{-9} 3.6×10^{-8}	1.3×10^{-5} 1.2×10^{-5}
30	7.5×10^{-10} 1.6×10^{-9}	6.7×10^{-3} 9.0×10^{-9}	2.6×10^{-8} 2.0×10^{-9}	3.0×10^{-8} 2.0×10^{-9}	9.7×10^{-8} 1.2×10^{-8}
50	1.9×10^{-9} 3.7×10^{-9}	1.2×10^{-1} 0.2	8.9×10^{-9} 7.8×10^{-9}	9.0×10^{-9} 8.0×10^{-9}	2.1×10^{-5} 2.1×10^{-5}
100	5.1×10^{-10} 9.2×10^{-10}	3.0 1.3	1.8×10^{-8} 5.8×10^{-8}	1.8×10^{-8} 5.8×10^{-8}	1.0×10^{-4} 1.5×10^{-4}
200	6.6×10^{-5} 1.1×10^{-4}	3.2 3.8	4.4×10^{-4} 9.1×10^{-5}	4.2×10^{-4} 9.1×10^{-5}	1.6×10^{-3} 9.3×10^{-4}
400	8.0×10^{-3}	3.4	4.0×10^{-3}	4.0×10^{-3}	1.2×10^{-1}

Table 5: Performance comparison (error)

jection method, despite its accuracy, is very slow. Whereas, the path-following method with SDH, SQH and SQQ formulations is very fast, sometimes more than 40 times faster than the projection method (see table 2 when $n = 200$), and gives results of acceptable accuracy. The other is that we did not gain any considerable advantage out of solving our problem as a mixed semidefinite and second-order cone problem (SQH, SQQ and SQV). This can be seen clearly by noticing the good performance of the formulation SDH, which solves the problem as a semidefinite program. However, it is well-known that positive definite Hankel matrices are extremely ill-conditioned; the optimal condition number for these matrices grows exponentially with the size of the matrix [6]. Therefore, computing the spectral decomposition (projection method) or solving the underlying linear systems (SDP/SOCP methods) might be numerically delicate.

References

[1] S. Al-Homidan. Hybrid methods for approximating Hankel matrix. *Numerical Algorithms.* To appear.

[2] S. Al-Homidan. Combined methods for apprpximating Hankel matrix. *WSEAS Transactions on systems*, 1:35–41, 2002.

[3] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95(1), 2003.

[4] F. Alizadeh, J. A. Haeberly, M. V. Nayakkanakuppann, M. Overton, and S. Schmieta. SDPPack, user's guide, 1997.

[5] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. *SIAM J. Optim.*, 8:746–768, 1998.

[6] B. Beckermann. The condition number of real vandermonde, krylov and positive definite hankel matrices. *Numer. Math.*, 85:553–577, 2000.

[7] R. Bellman and K. Fan. On systems of linear inequalities in Hermitian matrix variables. In V. L. Klee, editor, *Convexity*, volume 7, pages 1–11. Proc. Symposia in Pure Mathematics, Amer. Math. Soc., Providence,RI, 1963.

[8] J. P. Boyle and R. L. Dykstra. A method of finding projections onto the intersection of convex sets in Hilbert space. *Lecture Notes in Statistics*, 37:28–47, 1986.

[9] J.P. Burg, D. G. Luenberger, and D. L. Wenger. Estimation of structured covariance matrices. *Proc. IEEE*, 70:963–974, 1982.

[10] E. de Klerk. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, volume 65 of *Applied Optimization Series.* Kluwer Academic Publishers, 2002.

[11] R. L. Dykstra. An algorithm for restricted least squares regression. *J. Amer. Stat.*, 78:839–842, 1983.

[12] W. Fang and A.E. Yagle. Two methods of Toeplitz-plus-Hankel approximation to a data covariance matrix. *IEEE Trans. Signal Processing*, 40:1490–1498, 1992.

[13] K. Fujisawa, M. Fukuda, M. Kojima, and K. Nakata. Numerical evaluation of SDPA (semidefinite programming algorithm). In H. Frenk, K. Roos, T. Terlakey, and S. Zhang, editors, *High Performance Optimization*, pages 267–301. Kluwer Academic Press, 2000.

[14] W. Glunt, L. Hayden, S. Hong, and L. Wells. An alternating projection algorithm for computing the nearest Euclidean distance matrix. *SIAM J. Matrix Anal. Appl.*, 11(4):589–600, 1990.

[15] K. M. Grigoriadis, A. E. Frazho, and R. E. Skelton. Application of alternating convex projection methods for computing of positive Toeplitz matrices. *IEEE Trans. Signal Processing*, 42:1873–1875, 1994.

[16] S. P. Han. A successive projection method. *Math. Programming*, 40:1–14, 1988.

[17] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and Applications*, (284):193–228, 1998.

[18] C. S. Macinnes. The solution to a structured matrix approximation problem using Grassman coordinates. *SIAM J. Matrix Anal. Appl.*, 211(2):446–453, 1999.

[19] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming*. SIAM, Philadelphia, 1994.

[20] Yu. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM J. Optim.*, 8:324–364, 1998.

[21] J. Von Neumman. *Functional Operators II, The geometry of orthogonal spaces*. Annals of Math. studies No.22, Princeton Univ. Press., 1950.

[22] J. Sturm. Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. Technical report, August 2002.

[23] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[24] Y. J. Suffridge and T. L. Hayden. Approximation by a Hermitian positive semidefinite Toeplitz matrix. *SIAM J. Matrix Analysis and Appl.*, 14:721–734, 1993.

[25] R. Tütüncü, K. Toh, and M. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. to appear.

[26] M. J. Todd. Semidefinite Optimization. *Acta Numerica*, 10:515–560, 2001.

[27] M. J. Todd, K. C. Toh, and R. H. Tütüncü. On the Nesterov-Todd direction in semidefinite programming. *SIAM J. Optim.*, 8:769–796, 1998.

[28] M. J. Todd, K. C. Toh, and R. H. Tütüncü. SDPT3 — a Matlab software package for semidefinite programming. *Optim. Methods Softw.*, 11:545–581, 1999.

[29] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.

[30] N. Wiener. On factorization of matrices. *Comm. Math. Helv.*, 29:97–111, 1955.

[31] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of Semidefinite Programming*. Kluwer Academic Publishers Group, Boston-Dordrecht-London, 2000.