Solving Hankel Matrix Approximation Problem using Semidefinite Programming

Suliman Al-Homidan[*]

February 19, 2006

Abstract

Positive semidefinite Hankel matrices arise in many important applications. Some of their properties may be lost due to rounding or truncation errors incurred during evaluation. The problem is to find the nearest matrix to a given matrix retrieving these properties. The problem is converted into a semidefinite programming problem as well as a problem comprising a semidefined program and second-order cone problem. The duality and optimality conditions are obtained and the primal-dual algorithm is outlined. Explicit expressions for a diagonal preconditioned and crossover criteria have been presented. Computational results are presented. A possibility for further improvement is indicated.

Key words : Primal-dual interior point method; Hankel matrix; semidefinite programming.
AMS (MOS) subject classifications; 65F99, 99C25, 65K30.

Contents

[*] Department of Mathematical Sciences, King Fahd University of Petroleum and Minerals, Dhahran 31261, PO Box 119, Saudi Arabia Email homidan@kfupm.edu.sa

Introduction

In view of its beautiful theory having significant applications, finite and infinite Hankel matrices have attracted attention of researchers working in different disciplines, see for example Peller [21] and Chen et al. [8]. The related concept of structural matrix approximation problem having applications to image processing has been studied by Fuhrmann [13] and Morgera [18]. Park et al. [20] have presented a method for structure preserving low-rank approximation of matrix which is based on structured Total Least Norm (STLN). They have observed that a classical problem with a new STLN-based algorithm can have a significant impact on the model reduction problem with Hankel matrices. It may be noted that such a problem arises in many branches of engineering such as speech encoding, filter design and higher-order state-space models of dynamic systems.

Motivated by [1, 2, 3, 17], we have studied, as an extension of the work by [4], approximation of an arbitrary matrix by positive semidefinite Hankel matrices. Macinnes [17] has used a full rank of a particular Hankel matrix while Al-Homidan [1, 2] has used a low-rank Hankel matrix based on the projection algorithm and the Newton method. The projection algorithm provides global convergence with slow convergence rate. This method has been combined with the Newton method to yield the best features of both approaches.

Anjos et al. [5] have studied a semidefinite programming approach for the nearest correlation matrix problem. It may be recalled that the nearest correlation matrix problem is to find a positive definite matrix with unit diagonal that is nearest in the Frobenius norm to a given symmetric matrix. This problem has been formulated as an optimization problem with quadratic objective function and semidefinite programming constraints by Anjos et al. and they have developed several algorithms to solve this problem. In the present paper we have studied a related problem to the nearest corellation matrix problem where the positive semidefinite matrix with unit diagonal is replaced by a positive semidefinite Hankel matrix. More precisely, we have discussed the solution of the following problem:

Given an arbitrary data matrix $A \in \mathbb{R}^{n\times n}$, find the nearest positive semidefinite Hankel matrix H to A, i. e.,

$$\begin{array}{ll} \text{minimize} & \phi = \frac{1}{2}\|H \ - \ A\|_F^2 \\ \text{subject to} & H \ \in \ \mathcal{P} \cap \mathcal{O}, \end{array} \tag{0.1}$$

where $\mathcal{P}$ is the set of all $n \times n$ symmetric positive semidefinite matrices $\mathcal{P} = \{H : \ H \succeq 0, \ H \in \mathcal{S}^n \}$, $\mathcal{O}$ is the set of all Hankel matrices, and $\mathcal{S}^n$ is the set of all symmetric matrices. Here, $\|A\|_F^2 = \sum_{i,j=1}^n A_{ij}^2$ is the Frobenius norm squared.

Section 1 is devoted to the basic results needed in subsequent sections while the projection method for solving (0.1) is given in Section 2. In Section 3, we formulate the problem first as a semidefined 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 4. The primal-dual algorithm is outlined in Section 5 including explicit expressions for diagonal preconditioner and crossover criteria. Computational results are presented in Section 6 and concluding remarks are given in Section 7.

1 Notation

Define $\mathcal{M}^n$ to be the space of $n \times n$ matrices, then for a general rectangular matrix $M = [m_1 \ m_2 \ \ldots \ m_n] \in \mathcal{M}^n$, $v = \operatorname{vec}(M)$ forms a vector from columns of M. The inverse mapping, vec^{-1}, and the adjoint mapping, vec^{*}, are given by $\operatorname{Mat} = \operatorname{vec}^{-1} = \operatorname{vec}^{*}$, the adjoint formula following from $\langle \operatorname{vec}(M), u\rangle = \langle M, \operatorname{vec}^{*}(u)\rangle$. We use the *trace inner product* $\langle M, N\rangle = \operatorname{trace} M^T N$, which induces the Frobenius norm. With this inner product, Mat (and vec) is an isometry. $M^\dagger$ denotes the *Moore-Penrose* generalized inverse, e.g. [7]. Define e the vector of ones and e_k the zero vector with one in the k-position. Define $P_{\mathcal{O}}(W)$ to be the orthogonal projection of W onto the subspace of Hankel matrices $\mathcal{O}$. We also need the operator $\mathcal{H} : \mathbb{R}^{2n-1} \longrightarrow \mathcal{O}$

$$\mathcal{H}(x) = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ x_2 & x_3 & \dots & x_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n+1} & \dots & x_{2n-1} \end{bmatrix}. \tag{1.1}$$

Also, define the isometry operator hvec $: \mathbb{R}^{2n-1} \longrightarrow \mathcal{O} \longrightarrow \mathbb{R}^{2n-1}$ as

$$\operatorname{hvec}(\mathcal{H}(x)) = \operatorname{hvec}(x) = [x_1 \ \sqrt{2}x_2 \cdots \sqrt{n}x_n \cdots \sqrt{2}x_{2n-2} \ x_{2n-1}]^T \tag{1.2}$$

for any $x \in \mathbb{R}^{2n-1}$. hvec is a linear operator satisfying the following: For any $x, y \in \mathbb{R}^{2n-1}$

$$\mathcal{H}(x) \bullet \mathcal{H}(y) = \operatorname{hvec}(x)^T \operatorname{hvec}(y), \tag{1.3}$$

$$\|\mathcal{H}(x) - \mathcal{H}(y)\|_F^2 = (\operatorname{hvec}(x) - \operatorname{hvec}(y))^T(\operatorname{hvec}(x) - \operatorname{hvec}(y)). \tag{1.4}$$

Here $U \bullet U = \operatorname{trace}(UU) = \sum_{i,j}^{n} U_{i,j}^2$. Let $\operatorname{hMat} = \operatorname{hvec}^{-1}$ denote the inverse mapping into $\mathcal{S}^n$, i.e., the one-one mapping between $\mathbb{R}^{2n-1}$ and $\mathcal{O}$. The adjoint operator $\operatorname{hMat}^{*} = \operatorname{hvec}$, since

$$\begin{aligned} \langle \operatorname{hMat}(v), \mathcal{H}(s)\rangle &= \operatorname{trace} \operatorname{hMat}(v)\mathcal{H}(s) \\ &= \operatorname{trace} \begin{pmatrix} v_1 & \frac{v_2}{\sqrt{2}} & \cdots & \frac{v_n}{\sqrt{n}} \\ \frac{v_2}{\sqrt{2}} & \frac{v_3}{\sqrt{3}} & \cdots & \frac{v_{n+1}}{\sqrt{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{v_n}{\sqrt{n}} & \frac{v_{n+1}}{\sqrt{n-1}} & \cdots & v_{2n-1} \end{pmatrix} \begin{pmatrix} s_1 & s_2 & \dots & s_n \\ s_2 & s_3 & \dots & s_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ s_n & s_{n+1} & \dots & s_{2n-1} \end{pmatrix} \\ &= v_1 s_1 + 2\frac{v_2}{\sqrt{2}} s_2 + \ldots + n\frac{v_n}{\sqrt{n}} s_n + \ldots + v_{2n-1}s_{2n-1} \\ &= \langle \operatorname{hvec}(\mathcal{H}(s)), v\rangle = \langle \operatorname{hMat}^{*}(\mathcal{H}(s)), v\rangle . \end{aligned}$$

2 The Projection Method

The method of successive cyclic projections onto closed subspaces C_i's was first proposed by von Neumann [24] and independently by Wiener [25]. 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:

Algorithm 2.1 *Alternating Projection Algorithm*

```
Let $X_1 = D$
    For $k = 1, 2, 3, \ldots$
        $X_{k+1} = P_1(P_2(X_k))$.
```

Then X_k converges to the nearest point 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] 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 [15]. The modified Neumann's algorithm when applied to (0.1) for a given data matrix A:

Algorithm 2.2 *Modified Alternating Projection Algorithm*

```
Let $A_1 = A$
    For $j = 1, 2, 3, \ldots$
        $A_{j+1} = A_j + [P_{\mathcal{P}}(P_{\mathcal{O}}(A_j)) - P_{\mathcal{O}}(A_j)]$.
```

Then $\{P_{\mathcal{O}}(A_j)\}$ and $\{P_{\mathcal{P}}(P_{\mathcal{O}}(A_j))\}$ converge in Frobenius norm to the solution. Here, $P_{\mathcal{P}}(A)$ is the projection of A onto the convex cone $\mathcal{P}$. It is simply setting the negative eigenvalues of the spectral decomposition of A to zero.

3 Mixed-Cone Formulation

In this section we introduce briefly a direct approach for solving (0.1) which is obtained by formulating it first as an SDP problem then as a mixed SDP and second-order cone problem, more details given in [4].

To take the advantage of the isometry operator hvec , we need A to be Hankel. If we project A onto $\mathcal{O}$, we get $P_{\mathcal{O}}(A)$. The following lemma shows that the nearest symmetric Hankel positive semidefinite matrix to A is exactly equal to the nearest symmetric Hankel positive semidefinite matrix to $P_{\mathcal{O}}(A)$.

Lemma 3.1 *Let $\mathcal{H}(x)$ be the nearest symmetric Hankel positive semidefinite matrix to $P_{\mathcal{O}}(A)$, then $\mathcal{H}(x)$ is so for A.*

Proof. If $P_{\mathcal{O}}(A)$ is positive semidefinite, then we are done. If not, then for any $\mathcal{H}(x) \in \mathcal{O}$, we have

$$(\mathcal{H}(x) - P_{\mathcal{O}}(A)) \bullet (P_{\mathcal{O}}(A) - A) = 0$$

since $P_{\mathcal{O}}(A)$ is the orthogonal projection of A. Thus,

$$||\mathcal{H}(x) - A||_F^2 = ||\mathcal{H}(x) - P_{\mathcal{O}}(A)||_F^2 + ||P_{\mathcal{O}}(A) - A||_F^2.$$

This complete the proof since the second part of the above equation is constant. ■

In view of Lemma 3.1, (0.1) is equivalent to the problem

$$\begin{array}{rll} \mu^* = & \text{minimize} & \frac{1}{2}||\mathcal{H}(x) - P_{\mathcal{O}}(A)||_F^2 \\ & \text{subject to} & \mathcal{H}(x) \succeq 0. \end{array} \tag{3.1}$$

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

$$\begin{array}{ll} & ||\mathcal{H}(x) - P_{\mathcal{O}}(A)||_F^2 \leq \alpha \\ \Leftrightarrow & (\text{hvec}\,(x) - \text{hvec}\,(a))^T(\text{hvec}\,(x) - \text{hvec}\,(a)) \leq \alpha \qquad \text{by} \quad (1.4) \\ \Leftrightarrow & \alpha - (\text{hvec}\,(x) - \text{hvec}\,(a))^T I(\text{hvec}\,(x) - \text{hvec}\,(a)) \geq 0 \\ \Leftrightarrow & \begin{pmatrix} I & (\text{hvec}\,(x) - \text{hvec}\,(a)) \\ (\text{hvec}\,(x) - \text{hvec}\,(a))^T & \alpha \end{pmatrix} \succeq 0 \\ & \qquad\qquad\qquad \text{by Schur Complement} \end{array} \tag{3.2}$$

where $\mathcal{H}(a) = P_{\mathcal{O}}(A)$. Hence we have the following SDP problem:

$$\begin{array}{ll} \text{minimize} & \alpha \\ \text{s t} & \begin{pmatrix} \mathcal{H}(x) & 0 & 0 \\ 0 & I & \text{hvec}\,(x) - \text{hvec}\,(a) \\ 0 & (\text{hvec}\,(x) - \text{hvec}\,(a))^T & \alpha \end{pmatrix} \succeq 0. \end{array} \tag{3.3}$$

This SDP problem has dimensions $2n$ and $3n$. The original problem (0.1) can also be formulated as a mixed SDP and second-order (or Lorentz) cone problem as follows: Since $||\mathcal{H}(x)-P_{\mathcal{O}}(A)||_F^2 = ||\text{hvec}\,(x)-\text{hvec}\,(a)||_2^2$ we have the following equivalent problem:

$$\begin{array}{ll} \text{minimize} & \alpha \\ \text{subject to} & \mathcal{H}(x)\succeq 0 \\ \begin{pmatrix} \alpha \\ \text{hvec}\,(x)-\text{hvec}\,(a) \end{pmatrix} \geq_Q 0. & \end{array} \tag{3.4}$$

The dimension of SOC is $2n$. The constraint $\geq_Q$ is the second-order cone constraint.

Several public domain software packages can solve (3.3) and (3.4). Many of them can be accessed via NEOS [12] at `http://www-neos.mcs.anl.gov/` (see also C. Helmberg's SDP page at `http://www.zib.de/helmberg/semidef.html`). 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.

4 Duality and Optimality Conditions

Recall the primal SDP given (3.3), an equivalent problem to (3.1) and hence to (0.1) is:

$$\mu^* = \min \frac{1}{2}||\text{hvec}\,(x)-\text{hvec}\,(a)||_2^2 \quad \text{subject to} \quad \mathcal{H}(x)\succeq 0, \quad x\in \mathbb{R}^{2n-1}. \tag{4.1}$$

To obtain optimality conditions, we use a dual problem. Slater's condition (strict feasibility) holds for (4.1). This implies that we have strong duality with the Lagrangian dual

$$\mu^* = \nu^* = \max_{\mathcal{H}(s)\succeq 0} \min_x \frac{1}{2}||\text{hvec}\,(x)-\text{hvec}\,(a)||_2^2 - \text{trace}\,\mathcal{H}(s)\mathcal{H}(x). \tag{4.2}$$

Let $\circ$ denotes the Hadamard product, $b=[1,2,\cdots,n,n-1,\cdots,1]^T$, and $f=\frac{1}{2}||\text{hvec}\,(x)-\text{hvec}\,(a)||_2^2$, then

$$\Delta_x f = (x-a)\circ b. \tag{4.3}$$

Similarly

$$\Delta_x(\text{trace}\,\mathcal{H}(s)\ \mathcal{H}(x)) = \Delta_x(\text{hvec}\,(s)^T\text{hvec}\,(x)) = s \circ b. \tag{4.4}$$

Now, we change (4.2) to the Wolfe dual by noting that the inner problem is a convex unconstrained problem. Using (4.3) and (4.4) the optimal solution for (4.2) is characterized by stationarity:

$$s = x - a. \tag{4.5}$$

The equation in the Lagrangian dual can be expressed as

$$\frac{1}{2}||\text{hvec}\,(x) - \text{hvec}\,(a)||_2^2 - \text{hvec}\,(s)^T\text{hvec}\,(x) \ = \ \frac{1}{2}b^T((x-a)^2 - x \circ s). \tag{4.6}$$

Thus we obtain the equivalent dual problem:

$$\begin{array}{rll} \mu^* = & \text{maximize} & \frac{1}{2}b^T((x-a)^2 - x \circ s) \\ & \text{subject to} & s = x - a \\ & & \mathcal{H}(s) \succeq 0. \end{array} \tag{4.7}$$

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

Theorem 4.1 *The optimal values $\mu^* = \nu^*$ and the primal-dual pair x, s are optimal for (4.1) and (4.7) if and only if*

$$\begin{array}{rcll} \mathcal{H}(x) & \succeq & 0 & \textit{(primal feasibility)} \\ x & = & a + s, \quad \mathcal{H}(s) \succeq 0 & \textit{(dual feasibility)} \\ \mathcal{H}(x)\mathcal{H}(s) & = & 0 & \textit{(complementary slackness)} \end{array}$$

■

For our primal-dual interior-point algorithm, we use

$$\mathcal{H}(x)\mathcal{H}(s) = \mu I \quad \text{perturbed complementary slackness.}$$

We can substitute the primal and dual feasibility equations into the perturbed complementary slackness equation and obtain a *single bilinear equation* in s that characterizes optimality for the perturbed log-barrier problem:

$$\begin{array}{rcl} & & F_\mu(s) : \mathbb{R}^{2n-1} \to \mathbb{R}^{n^2}, \\ F_\mu(s) : & = & \text{vec}\,[\{\mathcal{H}(s) + \mathcal{H}(a)\}\mathcal{H}(s)] - \mu\text{vec}\,I = 0. \end{array} \tag{4.8}$$

Note that the original problem has $2n-1$ variables and the semidefiniteness constraint on $\mathcal{H}(x)$. Therefore, the dual problem has $2n-1$ variables. Hence, dual based algorithms do not reduce the size of the problem and standard primal-dual based algorithms have $4n-2$ variables.

We solve (4.8) using an inexact Gauss-Newton method. Linearizing, we obtain a linear system for the search direction Δs:

$$\begin{array}{rcl} F_\mu(s+\Delta s) & = & F_\mu(s) + F'_\mu(\Delta s) + o(\|\Delta s\|) \\ & = & F_\mu(s) + \mathrm{vec}\,\{(\mathcal{H}(s)+\mathcal{H}(a))\mathrm{hMat}\,(\Delta s) \;+\; \mathrm{hMat}\,(\Delta s)\mathcal{H}(s)\} \\ & & \qquad\qquad\qquad + o(\|\Delta s\|). \end{array} \tag{4.9}$$

Therefore,

$$\begin{array}{rcl} F'_\mu(\Delta s) & = & \mathrm{vec}\,\{(\mathcal{H}(s)+\mathcal{H}(a))\mathrm{hMat}\,(\Delta s) \;+\; \mathrm{hMat}\,(\Delta s)\mathcal{H}(s)\} \\ & = & \mathrm{vec}\,(\mathcal{A}+\mathcal{S})\,(\Delta s). \end{array} \tag{4.10}$$

This is a linear, full rank, overdetermined system. We use its least squares solution as the search direction Δs in our algorithm. This solution is found using preconditioned conjugate-gradient (PCG) with diagonal preconditioners. Note that $\Delta s \in \mathbb{R}^{2n-1}$, but the cost of evaluating $(\mathcal{A}+\mathcal{S})\,(\Delta s)$ is equivalent to one matrix-matrix multiplication. The adjoint $(F'_\mu)^*$ for $w \in \mathbb{R}^{n^2}$ is

$$(F'_\mu)^*(w) = \mathrm{hvec}\,\{P_\mathcal{O}\{\mathrm{Mat}\,(w)^T[\mathcal{H}(s)+\mathcal{H}(a)] + \mathcal{H}(s)\mathrm{Mat}\,(w)^T\}\}. \tag{4.11}$$

5 The Algorithm

Using equation (4.8) to develop a primal-dual interior-exterior-point algorithm, 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 include expression for a diagonal preconditioner, then we explain crossover criteria.

5.1 Diagonal Preconditioning

Preconditioning is essential for efficient solution of the least squares problem (4.10). We find operators P and find the least squares solution of

$$(\mathcal{A}+\mathcal{S})\,P^{-1}(\widehat{\Delta s}) = -F_\mu(s),$$

where

$$\widehat{\Delta s} = P(\Delta s).$$

The inverses are not found explicitly. The operator P has a simple structure so that the linear systems can be solved efficiently.

Optimal diagonal scaling has been studied in, e.g., [14, Sect. 10.5], and [10, Prop. 2.1(v)]. In the latter reference, it was shown that for a full rank matrix $A \in \mathbb{R}^{m\times n}$, $m \geq n$, and using the condition number $\omega(K) = n^{-1}\mathrm{trace}\,(K)/\det(K)^{1/n}$, the optimal scaling, i.e., the solution of the optimization problem

$$\min_D \ \omega((AD)^T(AD)) \quad \text{subject to } D \text{ a positive diagonal matrix,} \tag{5.1}$$

is given by $d_{ii} = 1/\|A_{:i}\|_2, i = 1, \dots, n$.

Therefore, the operator P is diagonal and is evaluated using the columns of the operator

$$F'_\mu(\cdot) = \mathcal{A}(\cdot) + \mathcal{S}(\cdot).$$

The columns are ordered using $k = 1, 2, \dots, 2n-1$ corresponding to the elements of s.

For the operators in F', we get

$$\mathcal{A}(e_k) = (\mathcal{H}(s) + \mathcal{H}(a))\mathrm{hMat}\,(e_k)$$

$$\mathcal{S}(e_k) = \mathrm{hMat}\,(e_k)\mathcal{H}(s).$$

Therefore

$$p_k = \|(\mathcal{A}+\mathcal{S})(e_k)\|_F^2 = \frac{1}{2}\|(\mathcal{H}(s)+\mathcal{H}(a))\mathrm{hMat}\,(e_k)+\mathrm{hMat}\,(e_k)\mathcal{H}(s)\|_F^2. \tag{5.2}$$

The diagonal preconditioners are inexpensive to calculate. However, in general, they are not strong enough [14].

5.2 Crossover Criteria

The Gauss-Newton approach has many advantages including full rank of the Jacobian at each iteration and optimality and a zero residual. Therefore, there is a local neighbourhood of quadratic convergence around each point on the central path and this neighbourhood is not restricted to $\mathcal{H}(s), \mathcal{H}(x) \succ 0$. In the neighbourhood of $\mu = 0$, we can set the centering parameter $\sigma = 0$ in the algorithm and use step lengths of one without backtracking to maintain positive definiteness. Standard convergence results, e.g. [9, 16], show that the Gauss-Newton method applied to $F(s) = 0$ is locally q-quadratically convergent, since the Jacobian at the optimum is full column rank (one to one operator). We follow [26] to discuss several constants used to determine the region of quadratic convergence. Since we have a zero residual, then the corresponding constant $\sigma = 0$. Since

$$\begin{array}{rcl} \|F'(\Delta s)\|_F & = & \|\mathcal{H}(x)\text{hMat}\,(\Delta s) + \text{hMat}\,(\Delta s)\mathcal{H}(s)\|_F \\ & \leq & \|\mathcal{H}(x)\text{hMat}\,(\Delta s)\|_F \ + \ \|\text{hMat}\,(\Delta s)\mathcal{H}(s)\|_F \\ & \leq & \|\mathcal{H}(x)\|_F\|\text{hMat}\,(\Delta s)\|_F \ + \ \|\text{hMat}\,(\Delta s)\|_F\|\mathcal{H}(s)\|_F \\ & = & \|\Delta s\|_2(\|\mathcal{H}(x)\|_F \ + \ \|\mathcal{H}(s)\|_F) \\ & \leq & \|\Delta s\|_2\sqrt{\|\text{hvec}\,(x)\|_2^2 \ + \ \|\text{hvec}\,(s)\|_2^2} \\ & & \qquad\qquad\qquad (\text{by Cauchy} - \text{Schwartz inequality}), \end{array} \tag{5.3}$$

the bound on the norm of the Jacobian is $\alpha = \sqrt{\|\text{hvec}\,(x)\|_2^2 \ + \ \|\text{hvec}\,(s)\|_2^2}$

$$\begin{array}{rcl} \|F'(s - \bar{s})(\Delta s)\|_F & = & \|(\mathcal{H}(x) - \mathcal{H}(\bar{x}))\text{hMat}\,\underline{(}\Delta s) + \text{hMat}\,(\Delta s)(\mathcal{H}(s) - \underline{\mathcal{H}}(\bar{s}))\|_F \\ & \leq & \|\Delta s\|_2(\|\text{hvec}\,(x) - \text{hvec}\,(x)\|_2 \ + \ \|\text{hvec}\,(s) - \text{hvec}\,(s)\|_2). \end{array} \tag{5.4}$$

Therefore the Lipschitz constant is $\gamma = 1$.

Now suppose that the optimum s^* is unique and the smallest singular value satisfies $\sigma_{\min}(F'(s)) \geq \sqrt{K}$, for all s in an ϵ_1 neighbourhood of s^*, for some constant $K > 0$. Following [9, Page 223], we define

$$\epsilon = \min\{\epsilon_1; \frac{K}{\alpha\gamma}\} = \min\{\epsilon_1; \frac{K}{\sqrt{\|\text{hvec}\,(x)^*\|_2^2 \ + \ \|\text{hvec}\,(s)^*\|_2^2}}\}.$$

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

$$\rho\sigma_{\min}(F'(s)) \geq \|\bar{\Delta} s\|\sqrt{\|\text{hvec}\,(x)\|_2^2 \ + \ \|\text{hvec}\,(s)\|_2^2}, \quad \rho \in (0, 1). \tag{5.5}$$

Note that 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 $\frac{\text{hvec}\,(x)^T\text{hvec}\,(s)}{\frac{1}{2}\|\mathcal{H}(x) \ - \ A\|_F^2+1} < 0.5$. This simpler heuristic never failed to converge, and q-quadratic convergence was observed.

5.3 Framework

Equation (4.8) and the linearization (4.9) is used to develop the primal-dual interior-point Algorithm 5.1. We let $\mathcal{F}^0$ denote the set of strictly feasible primal-dual points; F' denotes the derivative of the function of optimality conditions.

```
Algorithm 5.1 (Primal-Dual Gauss-Newton via PCG)
Input:    Objective: data matrix $A \in \mathcal{S}^n$
              Tolerances: $\delta_1$ (gap), $\delta_2$ (crossover)
Initialization:
              $\mathcal{H}(s^0), \mathcal{H}(x^0) = \mathcal{H}(s^0) + P_{\mathcal{O}}(A) \succ 0$
gap $=$ trace $\mathcal{H}(s^0)\mathcal{H}(x^0)$;  $\mu = \text{gap}/n$;  $\sigma = 1$;  objval $= \frac{1}{2}\|\mathcal{H}(x) - P_{\mathcal{O}}(A)\|_F^2$.
while $\min\{\frac{\text{gap}}{\text{objval}+1}, \text{objval}\} > \delta_1$
    if  $\min\{\frac{\text{gap}}{\text{objval}+1}, \text{objval}\} < \delta_2$  then
        $\sigma = 0$
    else
        update $\sigma$
    end if
    Find LSS of $F'_{\sigma\mu}(s)\,(\Delta s) = -F_{\sigma\mu}(s)$  (using LSQR)
    update $\mathcal{H}(s) = \mathcal{H}(s) + \alpha\, \text{hMat}\,(\Delta s), \alpha > 0$,  $\mathcal{H}(x) = \mathcal{H}(s) + P_{\mathcal{O}}(A)$,
            $(\mathcal{H}(s), \mathcal{H}(x) \succ 0)$
            gap $=$ trace $\mathcal{H}(s)\mathcal{H}(x)$;   $\mu = \text{gap}/n$;
            objval $= \frac{1}{2}\|\mathcal{H}(x) - P_{\mathcal{O}}(A)\|_F^2$
endwhile
```

At each iteration, the iterate s is available and we find a new iterate by taking a step in the (inexact) Gauss-Newton search direction Δs. The search direction is found using a conjugate gradient method, LSQR [19]. The cost of each CG iteration is a matrix multiplication, see e.g. (4.10). Until the crossover, we ensure that the new iterate $s + \alpha\Delta s$ results in both $\mathcal{H}(x)$, $\mathcal{H}(s)$ sufficiently positive definite; then, we take $\alpha = 1$ after the crossover. By our construction, the iterates maintain both primal and dual feasibility.

There are many advantages of this algorithm: Primal and dual feasibility is exact during each iteration. There is no (costly, dense) Schur complement system to form. There is no need to find $\mathcal{H}(s)^{-1}$. By the robustness of the algorithm, there is no need to enforce positivity of $\mathcal{H}(s)$, $\mathcal{H}(x)$ once μ gets small enough; q-quadratic convergence is obtained. The entire work of the algorithm lies in finding the search direction at each iteration by solving a least squares problem using a CG type algorithm. Each iteration of the CG algorithm involves a matrix-matrix multiplication. The more eficiently we can solve these least squares problems, the faster our algorithm will be.

6 Computational Results

We solve a set of problems using five approaches: (i) Algorithm 5.1, (ii) Algorithm 5.1 with preconditioning, (iii) the mixed-cone SDP formulation (3.4), (iv) the SDP formulation (3.3), and (v) Algorithm 2.2. The SDPT3-3.0 code [22, 23] was used for approaches (iii) and (iv). The tests were done using MATLAB 6.5 on a Pentium IV PC with 512MB of RAM.

We solved problems with dimensions $n = 10, \dots, 500$. The results were obtained by applying the methods as follows: A positive definite Hankel matrix A was formed randomly, then the matrix obtained was perturbed by adding random noise matrix S to A, where elements of S vary between -0.50 and 0.50. The problem is to recover the matrix before the noise was added. In all cases, we found the optimum to high accuracy, at least ten decimals. The results appear in Figures 6.1. We can see the correlation between the cpu time and size of the matrix A. The number of iterations required by Algorithm 5.1 remains essentially constant from 10 to 14 iterations for all the test problems. In the small problems, the number of LSQR iterations required by one iteration of Algorithm 5.1 ranges from 8 to 20 iterations. Then, this number increases for the mid-size problems to ranges from 15 to 32 iterations. However, in the larger problems, the number of iterations ranges from 25 to 44 iterations. Figure 6.2 shows the total number of iterations by LSQR against the size of the matrix A.

The data matrix A is always dense. Even if some of the elements are zeros, they will disappear when we project to the Hankel space $\mathcal{O}$. However, if any of anti-diagonals of A is all zeros, then these anti-diagonals can be eleminated and the dimension of the problem will be reduced by the number of zero anti-diagonals. In the absence of sparsity in the data matrix A, numerical experiments have shown that the diagonal preconditioning, is less efficient and extra time is needed in the calculation of the preconditioning with small reduction in the Gauss-Newton iterations and CPU time see Figures 6.1 and

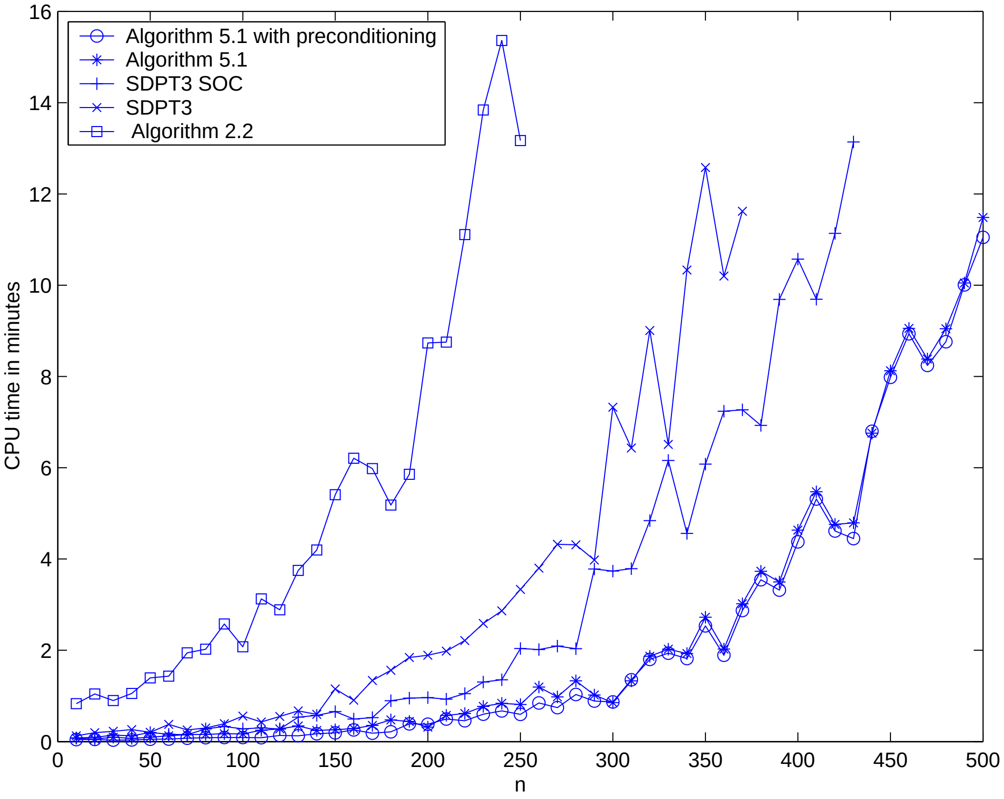

Figure 6.1: Comparing all five approaches

6.2.

As is typical with interior-point methods, the number of iterations of Algorithm 5.1 and SDPT3 are reasonably constant. However, each iteration of Algorithm 5.1 or SDPT3 involves solving a system of linear equations using LSQR. The number of iterations by LSQR depends on the size of the problem. The number of SDPT3 iterations are 14–20, and the number of Algorithm 5.1 iterations are 10–14 while the total number of LSQR iterations vary between 100 when $n = 10$ and 750 when $n = 500$ (see Figure 7.2).

After the crossover, to get q-quadratic convergence, centering σ is set to 0 and steplength α is set to 1. This allows for warm starts. Long steps can be taken beyond the positivity boundary. This improves the convergence rates in all the numerical tests; the crossover starts in most tests at the $10 - 11$th iteration and converges in $2 - 4$ iterations.

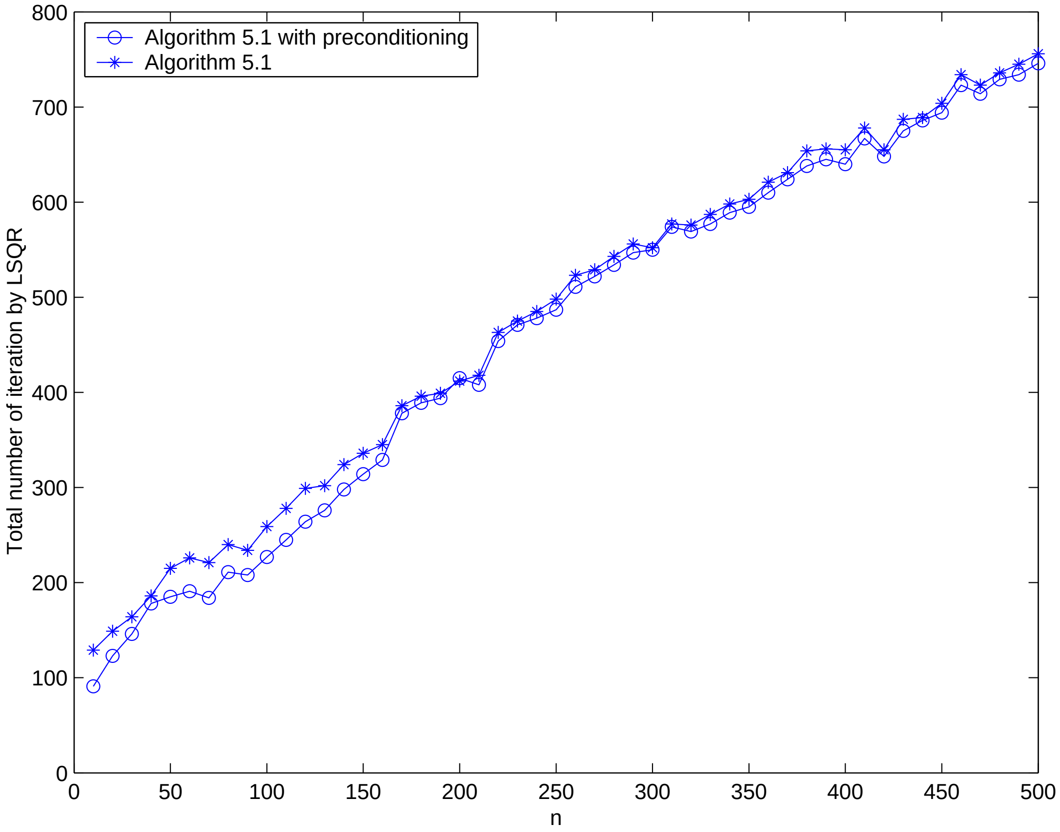

Figure 6.2: Comparing the number of iterations by LSQR against the size of the matrix A.

7 Conclusion

We have presented a p-d-i-e-p algorithm for finding the closest symmetric positive semidefinite Hankel matrix to the data matrix. The numerical tests show promising results. The approach is based on the strong robust primal-dual path-following interior-point framework without using the symmetrization step or the Schur complement system. The method uses basic tools that are successful for solving an overdetermined system of nonlinear equations with zero residual, i.e. PCG applied to the GN method. The total cost of an iteration lies in the solution of a linear least squares problem. This least squares problem is solved using the (preconditioned) conjugate gradient type method of Paige and Saunders [19]. The cost of each CG iteration is a matrix multiplication (essentially $\mathcal{H}(x)\mathrm{hMat}\,(\Delta s)$). We have shown how to derive diagonal preconditioning for this approach. However, the numerical tests show that threre is a small reduction in CPU time and so we would like to improve upon this methods to get better results. Different preconditioning such as preconditioning based for example on incomplete factorization might be more efficient [6].

Acknowledgments
Much of the research necessary for writing this paper was done during a sabbatical year I spent at C&O Department, Waterloo University. I would like to thank Professor Henry Wolkowicz for providing me with background on semidefinite programming and for patiently answering my questions. The author is grateful King Fahd University of Petroleum and Minerals for providing excellent research facilities. The author is also grateful to the referees for comments and suggestions improving the final draft of this paper.

References

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

[2] S. Al-Homidan. Hybrid methods for approximating Hankel matrix. *Numerical Algorithms*, 32:57–66, 2003.

[3] S. Al-Homidan. Structured methods for solving Hankel matrix approximation problems. *Pacif Jouranl of Optimization*, 1:599–610, 2005.

[4] M. M. Alshahrani and S. Al-Homidan. Mixed semidefinite and second-order cone optimization approach for the Hankel matrix approximation problem. *Nonlinear Dynamics and Systems Theory.* to appear.

[5] M. Anjos, N. Higham, P. Takouda, and H. Wolkowicz. A smdefinie programming approach for the nearest correlation matrix problem. preliminary research report, University of Waterloo, Waterloo, Ontario, 2003.

[6] O. Axelsson and V. A. Barker. *Finite element solution of boundary value problems.* Computer Science and Applied Mathematics. Academic Press Inc., Orlando, FL, 1984. Theory and computation.

[7] A. Ben-Israel and T. Greville. *Generalized inverses: theory and applications.* Robert E. Krieger Publishing Co. Inc., Huntington, N.Y., 1980. Corrected reprint of the 1974 original.

[8] Y. Chen and D. S. Lubinsky. Smallest eigenvalues of Hankel matrices for exponential weights. *J. Math. Anal. Appl.*, 293(2):476–495, 2004.

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

[10] J.E. Dennis and H. Wolkowicz. Sizing and least-change secant methods. *SIAM J. Numer. Anal.*, 30(5):1291–1314, 1993.

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

[12] M. C. Ferris, M. P. Mesnier, and J. J. Moré. NEOS and Condor: solving optimization problems over the Internet. *ACM Transactions on Mathematical Software*, 26(1):1–18, March 2000.

[13] D.R. Fuhrmann. Application of toeplitz covariance estimation to adaptive beamforming and detection. *IEEE Trans. Signal Processing*, 39:2194–2198, 1991.

[14] A. Greenbaum. *Iterative methods for solving linear systems.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

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

[16] C. T. Kelley. *Iterative methods for linear and nonlinear equations*, volume 16 of *Frontiers in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. With separately available software.

[17] C. S. Macinnes. The solution to a structured matrix approximation problem using Grassman coordinates. *SIAM J. Matrix Anal. Appl.*, 21(2):446–453, 1999.

[18] S.D. Morgera. The role of abstract algebra in structured estimation theory. *IEEE Transactions on Information Theory Society*, 38:1053–1065, 1992.

[19] C. C. Paige and M. A. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982.

[20] H. Park, L.Zhang, and J. B. Rosen. Low rank approximation of a Hankel matrix by structured total least norm. *BIT*, 39(4):757–779, 1999.

[21] V. V. Peller. *Hankel operators and their applications*. Springer Monographs in Mathematics. Springer-Verlag, New York, 2003.

[22] 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.

[23] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3, 2003. Computational semidefinite and second order cone programming: the state of the art.

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

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

[26] Henry Wolkowicz. Solving semidefinite programs using preconditioned conjugate gradients. *Optim. Methods Softw.*, 19(6):653–672, 2004.