

# Hybrid Methods for Approximating Hankel Matrix

Suliman Al-Homidan

*Department of Mathematical Sciences, King Fahd University of Petroleum & Minerals, Dhahran 31261, PO Box 119, Saudi Arabia E-mail: homidan@kfupm.edu.sa*

## Abstract

Hybrid methods for minimizing least distance functions with Hankel positive semi-definite matrix constraints are considered. Our approach is based on (i) a projection algorithm which converges globally but slowly; and (ii) the Newton method which is faster. Hybrid methods that attempt to combine the best features of both methods are then considered. Comparative numerical results are reported.

**Key words** :Alternating projections, Hankel matrix, least distance functions, non-smooth optimization, positive semi-definite matrix, Newton method.

**AMS (MOS)** subject classifications; 65F99, 99C25, 65F30.

## 1 Introduction

Hankel matrices appear naturally in a variety of problems of engineering interest and statistical application [8]: communication, control engineering, filter design, identification, model reduction and broadband matching, and in different fields of mathematics, e.g., in systems theory, integral equations, and operator theory.

Hankel matrices possess certain properties regarding their rank and positive semi-definite structures depending on the construction or arrangement of their elements. In practical applications, these matrices are constructed from noisy observations and hence some of their nice properties may be destroyed or changed. The signal processing problem is to estimate the matrices with desired properties so that the estimated matrix is close to the given observation in some reasonable sense.

Throughout this paper, the real Hilbert space of all  $n \times n$  symmetric matrices is denoted by  $\mathcal{H}$  and  $\langle \cdot, \cdot \rangle$  an inner product on  $\mathcal{H}$ . The Frobenius norm is given by  $\|A\|_F^2 = \langle A, A \rangle = \text{trace}(A^2)$ , and the distance between  $A \in \mathcal{H}$  and  $B \in \mathcal{H}$  is  $\|A - B\|_F$ . The orthogonal projection from  $\mathcal{H}$  onto a closed convex set  $K_i$  is denoted by  $P_i$ . The operator  $P_i$  defined by

$$P_i(B) = A \quad \text{where } B \in \mathcal{H} \quad \text{and } A \in K_i$$

provides a solution to the optimization problem

$$\|A - B\| = \min\{\|C - B\|; \quad \forall C \in K_i\}.$$

Also,  $A = P_i(B)$  is a unique element in the convex set  $K_i$  characterized by

$$\langle B - A, C - A \rangle \leq 0 \quad \forall C \in K_i.$$

A Hankel matrix  $H$  is denoted by

$$H = \begin{bmatrix} h_1 & h_2 & \dots & h_n \\ h_2 & h_3 & \dots & h_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_n & h_{n+1} & \dots & h_{2n-1} \end{bmatrix} = \text{Hankel}(h_1, h_2, h_3, \dots, h_{2n-1}). \quad (1.1)$$

We consider the following problem: Given an arbitrary data matrix  $F \in \mathbb{R}^{n \times n}$ , find the nearest positive semi-definite Hankel matrix  $H$  to  $F$  that minimizes

$$\text{minimize} \quad \phi = \|F - H\|_F. \quad (1.2)$$

The problem was studied by MacInnes [12]; he proposed a method for finding the best approximation of a matrix  $F$  by a full rank Hankel matrix. In [12], the initial problem of best approximation of one matrix by another is transformed to a problem involving best approximation of a given vector by a second vector whose elements are constrained so that its inverse image is a Hankel matrix. The alternating projection algorithm was successfully used in solving similar approximation problems for distance matrices [3]. Oh et. al. [13] used the alternating projection onto fuzzy convex sets when three or more convex sets do not intersect. Related problems were also studied by [14, 15] and [16] in relation to signal processing problems.

In the past ten years, there has been much interest in the interior point methods applied to problems with semidefinite matrix constraints (e.g. the survey papers [18], [17] and [19] and the references therein). Semidefinite programming optimizes a linear function subject to positive semidefinite matrix. It is a convex programming problem since the objective and constraints are convex. In this paper, we deal with a little different problem since the objective is quadratic; also an additional rank constraint is added which makes the problem unconvex and harder to solve. Here, we use a different approach than the interior point methods. Section 2 contains a brief description of the projection method for solving (1.2). The problem is formulated as a constrained least distance problem in which the constraint is the intersection of two convex sets. In Section 3, the problem is formulated as a smooth unconstrained minimization problem, then solved using the Newton method. Thus we are led to study hybrid methods in Section 4 of the paper. The hybrid method has two different modes of operation. One is a projection method which provides global convergence and enables the correct rank to be determined. The other is a Newton method which enables rapid convergence to be obtained. An important feature concerns the interfacing of these modes of operation. Thus it has to be decided which method to use first, and when to switch between methods. Finally, in Section 4, numerical comparisons of these methods are carried out.

## 2 Alternating Projection Algorithm

In this section, we describe a projection algorithm for solving the Hankel matrix approximation problem (1.2). This algorithm is derived from an alternating projection algorithm due to Dykstra [5] for finding the least distance from a fixed point to an intersection of convex sets. Along with this, we give two convex sets and formulas for certain projection maps that are needed in this section.

It is convenient to define two convex sets for the purpose of constructing the problem. The set of all  $n \times n$  symmetric positive semi-definite matrices

$$K_{\mathbb{R}} = \{A : A \in \mathbb{R}^{n \times n}, A^T = A \text{ and } \mathbf{z}^T A \mathbf{z} \geq 0 \quad \forall \mathbf{z} \in \mathbb{R}^n\} \quad (2.1)$$

is a convex cone of dimension  $n(n+1)/2$ . Also,

$$K_H = \{H : H \in \mathbb{R}^{n \times n}, H \text{ is a Hankel}\}. \quad (2.2)$$

is a convex set of dimension  $2n-1$ .

In addition, we need formulae for the projection maps denoted by  $P_{\mathbb{R}}(\cdot)$  and  $P_H(\cdot)$  on to  $K_{\mathbb{R}}$  and  $K_H$ . The projection map  $P_{\mathbb{R}}(F)$  formula on to  $K_{\mathbb{R}}$  is given by [10]

$$P_{\mathbb{R}}(F) = U \Lambda^+ U^T, \quad (2.3)$$

where

$$\Lambda^+ = \begin{bmatrix} \Lambda_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad (2.4)$$

and  $\Lambda_s = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_s]$  is the diagonal matrix formed from the positive eigenvalues of  $F$ .

The projection map  $P_H(F)$  formula on to  $K_H$  is given by

$$P_H(F) = \text{Hankel}(h_0, h_1, \dots, h_{2n-1}), \quad (2.5)$$

where

$$h_k = \frac{1}{k} \sum_{i=1}^k f_{k+1-i}, \quad k = 1, \dots, n$$

$$h_k = \frac{1}{2n-k} \sum_{i=1}^{2n-k} f_{n-i+1, k+i-n}, \quad k = n+1, \dots, 2n-1. \quad (2.6)$$

In other words,  $P_H$  maps each elements in anti-diagonal of  $F$  into the average of the elements in that anti-diagonal of  $F$ .

The Dykstra-Han algorithm solves the problem

$$\begin{aligned} & \text{minimize} && \|\mathbf{f} - \mathbf{x}\|_2 \\ & \text{subject to} && \mathbf{x} \in \bigcap_{i=1}^m K_i, \end{aligned}$$

where the  $K_i$  are convex sets in  $\mathbb{R}^n$  and  $\mathbf{f}$  is a given arbitrary point. The algorithm initializes  $\mathbf{f}^0 = \mathbf{f}$  and generates a sequence  $\{\mathbf{f}^{(k)}\}$  using the iteration formula

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + P_m(\dots P_1(\mathbf{f}^{(k)})) - P_1(\mathbf{f}^{(k)}). \quad (2.7)$$

Here  $P_i(\mathbf{f})$  denotes the  $l_2$  projection of  $\mathbf{f}$  on to  $K_i$ , that is, the (unique) nearest vector to  $\mathbf{f}$  in  $K_i$ . It is shown by Boyle and Dykstra [4] that  $P_i(\dots P_1(\mathbf{f}^{(k)}) \dots) \rightarrow \mathbf{x}^*$  for any  $i \geq 1$ . However the sequence  $\{\mathbf{f}^{(k)}\}$  does not in general converge to  $\mathbf{x}^*$  (see [2]).

In applying Dykstra-Han algorithm to the Hankel matrix approximation, it is appropriate to use the Frobenius matrix norm, and to express (1.2) as

$$\begin{aligned} & \text{minimize} && \|F - H\|_F \\ & \text{subject to} && H \in K_{\mathbb{R}} \cap K_H \end{aligned} \quad (2.8)$$

where  $K_{\mathbb{R}}$  and  $K_H$  are given by (2.1) and (2.2), respectively. To apply algorithm (2.7) we need the formulae for the projection maps  $P_{\mathbb{R}}(\cdot)$  and  $P_H(\cdot)$ , given by (2.3) and (2.5) and corresponding, respectively, to  $P_1(\cdot)$  and  $P_2(\cdot)$  in (2.7). These are the maps from  $\mathcal{H}$  on to  $K_{\mathbb{R}}$  and  $K_H$ .

Now we use the projection maps  $P_{\mathbb{R}}(F)$  and  $P_H(F)$  given by (2.3) and (2.5) to implement the Dykstra-Han algorithm (2.7). Hence, the new algorithm is as follows: Given an arbitrary distance matrix  $F \in \mathbb{R}^{n \times n}$ , the algorithm is initialized by  $F^{(0)} = F$  and the iteration formula is

$$F^{(k+1)} = F^{(k)} + P_H(P_{\mathbb{R}}(F^{(k)})) - P_{\mathbb{R}}(F^{(k)}). \quad (2.9)$$

The sequences  $\{P_{\mathbb{R}}(F^{(k)})\}$  and  $\{P_H(P_{\mathbb{R}}(F^{(k)}))\}$  converge to the solution  $H$  of (2.8) and hence (1.2).

### 3 Newton Method

In the previous section, the alternating projection algorithm computes a unique solution for (2.8) since the sets  $K_{\mathbb{R}}$  and  $K_H$  are convex. When Newton method are applied to solve the problem, this requires the rank knowledge of the matrix ( $H$ ); hence (2.8) loses convexity and Newton method loses second order rate of convergence and this increases the difficulty. Therefore, in this section, we consider a different approach to (2.8). The main idea is to replace (2.8) by a smooth unconstrained optimization problem in order to use second order convergent Newton method. Problem (2.8) is solved by Newton method, also the relevant formulae for derivatives are given. Partial connection between the problem and signal processing is given in the following factorization.

Classical results about Hankel matrices that go back to [6] may be re-stated according to which a nonsingular positive semi-definite real Hankel matrix can be represented as the product of a Vandermonde matrix and its transpose and a diagonal matrix in between

$$H = VDV^T, \quad (3.1)$$

where  $D$  is an  $m \times m$  diagonal matrix with positive diagonal entries and  $V$  is an  $n \times m$  real Vandermonde matrix

$$V = [x_j^i], \quad i = 0, \dots, n-1, \quad j = 1, \dots, m \quad (3.2)$$

(see [1, 11]).

We assume that the rank of  $H$  is known to be  $m$ . If the rank  $m$  is known, it is possible to express (1.2) as a smooth unconstrained optimization problem in the following way. Since the unknown in (1.2) is the matrix  $H$ , therefore the unknowns are chosen to be the elements of the matrices  $V$ ;  $x_1, \dots, x_m$  and  $D$ ;  $d_1, \dots, d_m$  introduced in (3.1). This gives us an equivalent unconstrained optimization problem to (1.2) in  $2m$  unknowns expressed as

$$\text{minimize} \quad \phi(V, D) = \|F - VDV^T\|_F^2. \quad (3.3)$$

Then the objective function  $\phi(V, D)$  is readily calculated by first forming  $H$  from  $V$  and  $D$  as indicated by (3.1) and (3.2), after which  $\phi$  is given by  $\phi(V, D) = \|F - H\|_F^2 = \|F - VDV^T\|_F^2$ . The elements of the matrix  $H$  take the form

$$h_{ij} = \sum_{k=1}^m d_k x_k^{i+j-2}. \quad (3.4)$$

Hence

$$\phi(V, D) = \sum_{i,j=1}^n (h_{ij} - f_{ij})^2 = \sum_{i,j=1}^n \left( \sum_{k=1}^m d_k x_k^{i+j-2} - f_{ij} \right)^2. \quad (3.5)$$

Our chosen method to minimize  $\phi(X)$  is the Newton method. This requires expressions for the gradient and the Hessian of  $\phi$ . The gradient vector is given by:

$$\frac{\partial \phi}{\partial d_s} = 2 \sum_{i,j=1}^n \left\{ x_s^\alpha \left[ \left( \sum_{k=1}^m d_k x_k^\alpha \right) - f_{ij} \right] \right\}, \quad (3.6)$$

$$\frac{\partial \phi}{\partial x_s} = 2 \sum_{i,j=1}^n \left\{ \alpha d_s x_s^{\alpha-1} \left[ \left( \sum_{k=1}^m d_k x_k^\alpha \right) - f_{ij} \right] \right\}, \quad (3.7)$$

where  $s = 1, \dots, m$  and  $\alpha = i + j - 2$ . The Hessian matrix is given by:

$$\frac{\partial^2 \phi}{\partial d_t \partial d_s} = 2 \sum_{\substack{i,j=1 \\ i \neq j}}^n x_t^\alpha x_s^\alpha, \quad (3.8)$$

$$\frac{\partial^2 \phi}{\partial x_s \partial d_s} = \frac{\partial^2 \phi}{\partial d_s \partial x_s} = 2 \sum_{i,j=1}^n \left\{ \alpha d_s x_s^{2\alpha-1} + \alpha x_s^{\alpha-1} \left( \sum_{k=1}^m d_k x_k^\alpha - f_{ij} \right) \right\}, \quad (3.9)$$

$$\frac{\partial^2 \phi}{\partial x_t \partial d_s} = \frac{\partial^2 \phi}{\partial d_s \partial x_t} = 2 \sum_{i,j=1}^n \alpha d_t x_s^\alpha x_t^{\alpha-1}, \quad t \neq s, \quad (3.10)$$

$$\frac{\partial^2 \phi}{\partial x_s^2} = 2 \sum_{i,j=1}^n \left\{ \alpha^2 d_s^2 x_s^{2\alpha-2} + \alpha(\alpha-1) d_s x_s^{\alpha-2} \left( \sum_{k=1}^m d_k x_k^\alpha - f_{ij} \right) \right\}, \quad (3.11)$$

$$\frac{\partial^2 \phi}{\partial x_t \partial x_s} = 2 \sum_{i,j=1}^n \alpha^2 d_s d_t x_s^{\alpha-1} x_t^{\alpha-1}, \quad t \neq s, \quad (3.12)$$

where  $s = 1, \dots, m$ ,  $t = 1, \dots, m$  and  $\alpha = i + j - 2$ .

Some care has to be taken when choosing the initial value of the matrices  $V$  and  $D$ , in particular the rank  $m$ . If not, the minimization method may not be able to increase  $m$ . An extreme case occurs when the initial matrix  $V = 0$  and  $D = 0$  is chosen, and  $F \neq 0$ . It can be seen from (3.6) and (3.7) that the components of the gradient vector are all zero, so that  $V = 0$  and  $D = 0$  is a stationary point, but not a minimizer. A gradient method will usually terminate in this situation, and so fail to find the solution.

An advantage of the Newton method is that it allows the spatial dimensions to be chosen by the user. This is useful when the rank is already known. For example if the entries in  $F$  are derived from a matrix that has rank  $m = 2$  or  $m = 3$ .

In general however the rank of the matrix is not known, for example the matrix that assumed has rank  $m = 3$  might be irreducibly embedded in matrix with rank  $m = 1$  or  $m = 2$ . We therefore must consider an algorithm in which we are prepared to revise our estimate of  $m$ . A simple strategy is to repeat the entire Newton method for different values of  $m$ . If  $m^*$  denotes the correct value of  $m$  which solves (1.2), then it is observed that the Newton method converges rapidly if  $m \leq m^*$ , and exhibits second order convergence. On the other hand if  $m > m^*$  then slow convergence is observed. One reason is that there are more variables in the problem. Also redundancy in the parameter space may have an effect. Thus it makes sense to start with a small value of  $m$ , and increase it by one until the solution is recognised. One way to recognise termination is when  $H^{(k)}$  agrees sufficiently well with  $H^{(k+1)}$ , where  $H^{(k)}$  denotes the positive semi-definite Hankel matrix obtained by minimizing  $\phi$ .

## 4 Hybrid Methods

A combination of both algorithms is introduced. Projection methods are globally convergent and hence potentially reliable, but often converge slowly, which can be very inefficient. Newton method is reliable and have a second order rate of convergence, but require the correct rank  $m^*$  to be known. We therefore consider hybrid methods in which the projection algorithm is used sparingly as a way of establishing the correct rank, whilst the Newton method is used to provide rapid convergence.

In order to ensure that each component method is used to best effect, it is important to transfer information from one method to the other. In particular, the result from one method is used to provide the initial data for the other, and vice versa. This mechanism has a fixed point property so that if one method finds a solution, then

the other method is initialized with an iterate that also corresponds to the solution. It is clear how to use the output data from Newton in the projection method but, unfortunately, the reverse is not easy.

We will evaluate two different algorithms which differ in respect of how  $m^{(0)}$  is initialized. Algorithm 1 is expressed as follows: Given any data matrix  $F \in \mathbb{R}^{n \times n}$ , let  $s$  be some pre-selected positive integer number and  $\epsilon$  some small number. Then the following algorithm solves (1.2)

**Algorithm 1** ( $F^{(0)} := F, s, \epsilon$ ):

*repeat* projection method

*until*  $m^{(l)} = m^{(l-j)}$   $j = 1, 2, \dots, s$ .

*repeat*

Apply one iteration of projection method;

$m^{(0)} := m^{(l)}$ ;

*repeat* Newton method;

*until*  $\|\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}\| \leq \epsilon$  and  $\|\mathbf{d}^{(k-1)} - \mathbf{d}^{(k)}\| \leq \epsilon$ ;

$F^{(k)} = VDVT$ ; ( $\mathbf{x}^{(k)}$  and  $\mathbf{d}^{(k)}$  from Newton method)

*until*  $\|F^{(k-1)} - F^{(k)}\| \leq \epsilon$ ;

*return* ( $F^* := F^{(k)}, \mathbf{x}^* := \mathbf{x}^{(k)}, \mathbf{d}^* := \mathbf{d}^{(k)}, m^* := m^{(k)}$ ).

The choice of  $s$  is a compromise between two effects. If  $s$  is small then the rank may not be accurately estimated, but the number of (expensive) iterations taken in the projection method is small. On the other hand, if  $s$  is large then a more accurate rank is obtained but the projection method needs more iterations.

In Algorithm 2,  $m^{(0)}$  is supplied by the user. This approach avoids the initial sequence of projection iterations, but works well if the user is able to make a good estimate of the rank, which is often the case. Thus, we can express Algorithm 2 as follows: Given any data matrix  $F \in \mathbb{R}^{n \times n}$ , let  $\epsilon$  be some small number; also choose  $m^{(0)}$  as a small integer number. Then the following algorithm solves (1.2)

**Algorithm 2** ( $F^{(0)} := F, m^{(0)}, \epsilon$ ):

*repeat*

*repeat* Newton method;

*until*  $\|\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}\| \leq \epsilon$  and  $\|\mathbf{d}^{(k-1)} - \mathbf{d}^{(k)}\| \leq \epsilon$ ;

$F^{(k)} = VDVT$ ; ( $\mathbf{x}^{(k)}$  and  $\mathbf{d}^{(k)}$  from Newton method)

Apply one iteration of projection method;

$m^{(0)} := m^{(l)}$ ;

*until*  $\|F^{(k-1)} - F^{(k)}\| \leq \epsilon$ ;

*return* ( $F^* := F^{(k)}, \mathbf{x}^* := \mathbf{x}^{(k)}, \mathbf{d}^* := \mathbf{d}^{(k)}, m^* := m^{(k)}$ ).

## 5 Numerical Results

The tables below show the performance of the four algorithms, projection, Newton, hybrid Algorithm 1 and hybrid Algorithm 2. All calculations were performed with MATLAB 6 on a Pentium 3. The results were obtained by applying the methods as follows. A matrix  $H$  was formed from (3.1) by randomly choosing  $m$  weights  $d_i$  in matrix  $D$ ,  $0 \leq d_j \leq 1.0$  and  $m$  values  $x_j$ ,  $0 \leq x_j \leq 1.0$  to determine the Vandermonde matrix  $V$ . The matrix thus obtained by (3.1) was perturbed by adding random noise matrix  $S$  to  $H$ , where elements of  $S$  vary between  $-0.10$  and  $0.10$ . The problem is to recover the  $m$  frequencies  $x_j$  and weights  $d_j$  that determine the matrix before the noise was added. The convergence criterion is that the maximum changes of the matrix  $H^{(k)}$  should be less than  $1 \times 10^{-5}$ .

$n$	PA			NA		A1					A2	
	$m^*$	PI	CPU	TNI	CPU	$s$	PI	$m^{(k)}$	NI	CPU	TNI	CPU
5	3	321	0.88	6	0.15	5	5	2	4	0.11	4	0.1
10	4	447	2.97	17	3.06	5	6	5	19	1.71	10	1.75
15	5	604	7.86	37	8.24	10	21	7	11	3.27	12	2.69
20	7	701	15.38	84	39.32	20	29	8	16	21.31	17	12.73
25	7	828	27.13	82	69.44	30	121	7*	18	20.1	24	19.3
30	9	644	29.66	97	83.8	30	93	9*	19	23.3	28	25.1

Table 1: Comparing the four algorithms.

PA: projection algorithm. NA: Newton algorithm. A1: Hybrid Algorithm 1. A2: Hybrid Algorithm 2. PI: Number of iterations performed by projection algorithm. CPU: time in seconds. NI: Number of iterations performed by Newton algorithm. TNI: Total number of iterations performed by Newton algorithm.

n	A2			
	$m^{(0)}$	NI in Newton	$m^{(k)}$ from OPA	NI in Newton
5	1	2	3*	2
10	2	3	4*	7
15	2	4	5*	8
20	3	5	7*	12
25	3	6	7*	18
30	3	11	9*	17

Table 2: Detailed progress of Algorithm 2.

OPA: One iteration of the projection algorithm. NI: Number of iterations.



Table 1 summarizes the results for the four different approaches and Table 2 shows the progress of Algorithm 2 in more detail. All four algorithms converge to essentially the same values. An asterisk indicates where the correct rank has been identified. In some cases, with Algorithm 1, the final rank is  $m^* + 1$  or  $m^* + 2$  but the solution is within the required tolerance. However, the time consumed by Algorithm 1 in these cases is larger than the other algorithms, because of the extra variables added on, see for example  $n = 20$ .

For the projection algorithm, each iteration involves an eigensolution, which entails relatively expensive  $O(n^3)$  calculations; also the number of iterations is very large. Thus, the projection algorithm is not competitive. For other algorithms, the housekeeping associated with each iteration is  $O(n^2)$ . Also, if care is taken, it is possible to calculate the gradient and Hessian in  $O(n^2)$  operations. For the Newton algorithm, the initial value  $m^{(0)} = 1$ , and  $m$  is increased by one until the solution is found. The total number of iterations performed by the Newton algorithm is relatively large. This is because Newton algorithm is repeated  $m$  times until the correct rank is found. This makes the Newton algorithm more expensive than the other algorithms. Also, it can be seen that the total number of iterations and time is much greater than is required by the hybrid methods. The initial value  $m^{(0)}$  is rather arbitrary; a smaller value of  $m^{(0)}$  would have given an even larger number of iterations.

Both hybrid algorithms are found to be effective. As  $n$  increases, Algorithm 1 takes an increasing number of projection iterations before the rank settles down. We find it better to increase the value of  $s$  as the value of  $m^*$  increases. Once the projection iteration has settled down, the Newton method finds the solution rapidly and no further projection steps are needed. Algorithm 2 requires more number of iterations (see Table 2) in the first call of the Newton method, after which one projection step finds the correct rank, and the next call of the Newton algorithm finds the solution in a few iterations.

**Acknowledgments** The author is grateful to Professor R. Fletcher for his helpful discussions and advice throughout this work. Also he is grateful to Professor Richard S. Varga for his comments on this paper and to the referee for his constructive criticism and his interesting suggestions. The author is grateful to King Fahd University of Petroleum and Minerals for providing research facilities.

## References

- [1] N. I Akhiezer and M. G. Krein, *Some Questions in the Theory of Moments*, Transl. Math. Monographs 2, Amer. Math. Soc., Providence, 1962.
- [2] S. Al-Homidan, *Hybrid methods for optimization problems with positive semi-definite matrix constraints*, Ph.D. Thesis, Dept of Mathematics and Computer Science, University of Dundee, Dundee, Scotland, 1993.

- [3] S. Al-Homidan and R. Fletcher, Hybrid methods for finding the nearest Euclidean distance matrix, in *Recent Advances in Nonsmooth Optimization* (Eds. D. Du, L. Qi and R. Womersley), World Scientific Publishing Co. Pte. Ltd., Singapore, 1995, pp. 1-17.
- [4] J. P. Boyle and R. L. Dykstra, A method for finding projections onto the intersection of convex sets in Hilbert space, in *Advances in Order Restricted Statistical Inference*, (Eds. R. Dykstra, T. Robertson, and F. T. Wright), Lecture Notes in Statistics 37, Springer-Verlag, Berlin, 1986, pp. 28-47.
- [5] R. L. Dykstra, An algorithm for restricted least squares regression, *J. Amer. Stat. Assoc.* 78 (1983), 839-842.
- [6] E. Fischer, *Über das Caratheodorysche Problem, Potenzreihen mit positivem reellen Teil betreffend* Rend Palermo 32, 1911.
- [7] R. Fletcher, *Practical methods of Optimization*, John Wiley and Sons, Chichester, 1987.
- [8] G. H. Golub and V. Pereyra The differentiation of pseudoinverses and nonlinear problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2)(1973), 413-432.
- [9] S. P. Han, A successive projection method, *Math. Programming*, 40(1988) 1-14.
- [10] N. Higham, Computing a nearest symmetric positive semi-definite matrix, *Linear Alg. and Appl.* 103(1988), 103-118.
- [11] I. S. Iohvidov, *Toeplitz and Hankel Matrices and Forms: Algebraic Theory*, transl. by G. Phillip A. Thijse, Birkhauser, Boston, 1987.
- [12] C. S. MacInnes, The Solution to a Structured Matrix Approximation Problem Using Grassman Coordinates, *SIAM Journal on Matrix Analysis and Applications*, 21(1999), 446-453.
- [13] S. Oh and R. J. Marks II, Alternating projection onto fuzzy convex sets, *Proc. IEEE*, 81(1993), 148-155.
- [14] J. Ponstein, A matrix minimization problem involving ranks, *Linear Alg. and Appl.* 92(1987), 81-106.
- [15] A. K. Shaw, S. Pokala and S. Kumaresan, Toeplitz and Hankel matrix approximation using structured approach, *Proc. IEEE*, 86(1998), 2349-2352.
- [16] A. K. Shaw and S. Kumaresan, Some structured matrix approximation problems, in *Proc. IEEE ICASSP. V. 4, Internat Conf. Acoustics, Speech, and Signal Processing*, 1988 pp. 2324-2327.

- [17] M. J. Todd, Semidefinite optimization, *Acta Numerica* 10(2001), 515–560.
- [18] L. Vandenberghe and S. Boyd, Semidefinite programming, *SIAMN Rev.* 38(1996), 49–95.
- [19] H. Wolkwicz, R. Saigal and L. Vandenberghe, *Handbook semidefinite programming, theory, algorithms, and applications* Kluwer Academic, 2000.