# Combined Methods for Approximating Hankel Matrix

SULIMAN Al-HOMIDAN
Department of Mathematical Sciences
King Fahd University of Petroleum and Minerals
KFUPM Box 119, Dhahran 31261, SAUDI ARABIA

*Abstract:* - Combined 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 filterSQP 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, filterSQP method least distance functions, non-smooth optimization, positive semi–definite matrix.

## 1 Introduction

Hankel matrices appear naturally in a variety of problems of engineering interest: comunication, control engineering, filter design, identification, model reduction and broadband matching, and in different fileds 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.

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

$$\text{minimize} \quad \phi = \|F \; - \; H\|_F \qquad (1)$$

where $\|.\|_F$ denotes the Frobenius norm.

The problem was studied by MacInnes [11]; he proposed a method for finding the best approximation of a matrix $F$ by a full rank Hankel matrix. In [11], 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. [12] used the alternating projection onto fuzzy convex sets when three or more convex sets do not intersect. Related problems were also studied by [13, 14] and [15] in relation to signal processing problems.

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

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

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

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

A Hankel matrix $H$ is denoted by

$$\begin{aligned} H &= \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ a_2 & a_3 & \cdots & a_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n+1} & \cdots & a_{2n-1} \end{bmatrix} \\ &= \text{Hankel}(a_1, a_2, a_3, \ldots, a_{2n-1}). \qquad (2) \end{aligned}$$

## 2 Alternating Projection Algorithm

In this section, we describe a projection algorithm for solving the Hankel matrix approximation problem (1). 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 probem. The set of all $n \times n$ symmetric positive semi–definite matrices

$$K_{\rm I\!R} = \{A : A \in {\rm I\!R}^{n\times n},\ A^T = A \text{ and } \mathbf{z}^T A \mathbf{z} \geq 0 \quad \forall\, \mathbf{z} \in {\rm I\!R}^n\} \tag{3}$$

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

$$K_H = \{H : H \in {\rm I\!R}^{n\times n},\ H \text{ is a Hankel}\}. \tag{4}$$

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

In addition, we need formulae for the projection maps denoted by $P_{\rm I\!R}(.)$ and $P_H(.)$ on to $K_{\rm I\!R}$ and $K_H$. The projection map $P_{\rm I\!R}(F)$ formula on to $K_{\rm I\!R}$ is given by [9]

$$P_{\rm I\!R}\,(F) = U\Lambda^+U^T, \tag{5}$$

where

$$\Lambda^+ = \begin{bmatrix} \Lambda_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{6}$$

and $\Lambda_s = \operatorname{diag}\,[\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_s]$ 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

$$P_H(F) = \text{Hankel}(h_0, h_1, \ldots, h_{2n-2}), \tag{7}$$

where

$$h_k = \frac{1}{k}\sum_{i=1}^{k} f_{k+1-i\ i}, \quad k = 1, \ldots, n$$
$$h_k = \frac{1}{2n-k}\sum_{i=1}^{2n-k} f_{n-i+1\ k+i-n}, \quad k = n+1, \ldots, 2n-1. \tag{8}$$

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

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

where the $K_i$ are convex sets in ${\rm I\!R}^n$ 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

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + P_m(\ldots P_1(\mathbf{f}^{(k)})\ldots) - P_1(\mathbf{f}^{(k)}). \tag{9}$$

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 $P_i(\ldots P_1(\mathbf{f}^{(k)})\ldots) \to \mathbf{x}^*$ 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) as

$$\begin{array}{ll} \text{minimize} & \|F - H\|_F \\ \text{subject to} & H \in K_{\rm I\!R} \cap K_H \end{array} \tag{10}$$

where $K_{\rm I\!R}$ and $K_H$ are given by (3) and (4), respectively. To apply algorithm (9) we need the formulae for the projection maps $P_{\rm I\!R}(.)$ and $P_H(.)$, given by (5) and (7) and corresponding, respectively, to $P_1(.)$ and $P_2(.)$ in (9). These are the maps from $K = \{A :\ A \in {\rm I\!R}^{n\times n}\}$ on to $K_{\rm I\!R}$ and $K_H$.

Now we use the projection maps $P_{\rm I\!R}(F)$ and $P_H(F)$ given by (5) and (7) to implement the Dykstra-Han algorithm (9). Hence, the new algorithm is as follows: Given an arbitrary distance matrix $F \in {\rm I\!R}^{n\times n}$, the algorithm is initialized by $F^{(0)} = F$ and the iteration formula is

$$F^{(k+1)} = F^{(k)} + P_H(P_{\rm I\!R}(F^{(k)}))) - P_{\rm I\!R}(F^{(k)}). \tag{11}$$

The sequences $\{P_{\rm I\!R}(F^{(k)})\}$ and $\{P_H(P_{\rm I\!R}((F^{(k)}))\}$ converge to the solution $H$ of (10) and hence (1).

## 3 The SQP Algorithms

In the previous section, the alternating projection algorithm computes a unique solution for (10) since the sets $K_{\mathbb{R}}$ and $K_H$ are convex. When SQP methods are applied to solve the problem, this requires the knowledge of the rank$(H)$; hence (10) loses convexity and this increases the difficulty. Therefore, in this section, we consider a different approach to (10). The main idea is to replace (10) by a smooth nonlinear programming problem in order to use a second order convergent filterSQP method. Problem (10) is solved by filterSQP method, also the relevant formulae for derivatives are given.

It is difficult to deal with the matrix set constraint in (3) and (4) since it is not easy to specify if the elements are feasible. Using partial $LDL^T$ factorization of $H$, this difficulty can be overcome. We assume that the rank of $H$ is known to be $m$. Permuting rows and columns if necessary, and partitioning $H$, then for $H$ sufficiently close to $H^*$, the partial factors $H = LDL^T$ can be calculated such that

$$\begin{aligned} L &= \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix}, D = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix}, \\ H &= \begin{bmatrix} H_{11} & H_{21}^T \\ H_{21} & H_{22} \end{bmatrix}, \end{aligned} \qquad (12)$$

where $L_{11}$, $D_1$ and $H_{11}$ are $m \times m$ matrices; $I$, $D_2$ and $H_{22}$ are $n-m \times n-m$ matrices; $L_{21}$ and $H_{21}$ are $n-m \times m$ matrices; $D_1$ is diagonal and $D_1 > 0$ and $D_2$ have no particular structure other than symmetry. At the solution, $D_2 = 0$ and $H$ is symmetric positive semi-definite Hankel matrix. In general,

$$D_2(H) \;=\; H_{22} \;-\; H_{21}H_{11}^{-1}H_{21}^T. \qquad (13)$$

Now if the structure of the matrix $H$ is in a Hankel form, i.e.,

$$H \;=\; \begin{bmatrix} x_1 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_n & \cdots & x_{2n-1} \end{bmatrix} = \text{Hankel}(x_1, \cdots, x_{2n-1}), \qquad (14)$$

then (13) enables the constraint $H \in K = K_{\mathbb{R}} \cap K_H$ to be written in the form

$$D_2(H(\mathbf{x})) = 0. \qquad (15)$$

Hence, (1) can now be expressed as

$$\begin{aligned} &\text{minimize} \quad \phi \\ &\text{subject to} \;\; D_2(H(\mathbf{x})) \;=\; 0 = Z^THZ, \end{aligned} \qquad (16)$$

where $Z \;=\; \begin{bmatrix} -H_{11}^{-1}H_{21}^T \\ I \end{bmatrix}$ is the basis matrix for the null space of $H$ when $D_2 = 0$. The Lagrange multipliers for the constraint (15) are $\Lambda$ relative to the basis $Z$ and the Lagrangian for porblem (16) is

$$\mathcal{L}(\mathbf{x}^{(k)}, \Lambda^{(k)}) \;=\; \phi - \Lambda : Z^THZ. \qquad (17)$$

The above approach has been studied in a similar way by [6].

The structure of the Hankel matrix $D$ has been given in (14), then

$$\phi = \sum_{i,j=1}^{n} (f_{ij} - h_{ij})^2 = \sum_{i,j=1}^{n} (f_{ij} - x_{i+j-1})^2, \qquad (18)$$

and $\bigtriangledown\phi \;=\; [\frac{\partial\phi}{\partial x_1} \cdots \frac{\partial\phi}{\partial x_{2n-1}}]^T$ where $\nabla$ denotes the gradient operator $(\partial/\partial x_1,\;\ldots,\;\partial/\partial x_{2n-1})^T$. Therefore

$$\begin{aligned} \frac{\partial\phi}{\partial x_s} &= 2\sum_{i=1}^{s}(x_s - f_{i\;s-i+1}) \qquad\qquad s = 1, \ldots, n \\ \frac{\partial\phi}{\partial x_s} &= 2\sum_{i=1}^{2n-s}(x_s - f_{n-i+1\;s+i-n}) \\ &\qquad\qquad\qquad s = n+1, \ldots, 2n-1. \end{aligned} \qquad (19)$$

Differentiating again gives

$$\frac{\partial^2\phi}{\partial x_r \partial x_s} = 0 \qquad \text{if} \quad r \neq s,$$

where $s, r = 1, \cdots, 2n-1$, and

$$\begin{aligned} \frac{\partial^2\phi}{\partial x_s^2} &= 2s, \quad s = 1, \ldots, n \\ \frac{\partial^2\phi}{\partial x_s^2} &= 2(2n-s). \quad s = n+1, \ldots, 2n-1 \end{aligned} \qquad (20)$$

The advantage of formula (15) is that expressions for both first and second derivatives of the constraints with respect to the elements of $H$ can be obtained. The simple form of (13) is utilized by writing the constraints $D_2(H) = 0$ in the following form:

$$d_{ij}(\mathbf{x}) \;=\; x_{i+j-1} - \sum_{k,l=1}^{m} x_{i+k-1}[H_{11}^{-1}]_{kl}\; x_{j+l-1} = 0 \qquad (21)$$

where $i, j = m+1, \cdots, n$ and $[H_{11}^{-1}]_{kl}$ denotes the element of $H_{11}^{-1}$ in $kl$–position. Thus (16) can be expressed as

$$\begin{aligned} &\text{minimize} \quad \phi = \sum_{i,j=1}^{n} (f_{ij} - x_{i+j-1})^2. \\ &\text{subject to } d_{ij}(\mathbf{x}) \;=\; 0 \qquad\qquad (22) \end{aligned}$$

In this problem, since the equivalent constraints $d_{ij}(\mathbf{x}) = 0$ and $d_{ji}(\mathbf{x}) = 0$ are both present, they would be stated only for $i \geq j$.

In order to write down the SQP method applied to (22), it is necessay to derive first and second derivatives of $d_{ij}$ which enable a second order rate of convergence to be achieved.

Let $I_s$ be an $m \times m$ matrix given by

$$I_s \;=\; \text{Hankel}(0, \ldots, 0, 1, 0, \ldots, 0),$$

where the "1" appearing in the first row is in the $s$th column and the "1" appearing in the first column is in the $s$th row. Hence the matrix $I_s$ is a matrix that contains "1"s in one across anti–diagonal and zeros elsewhere. Now differentiating $H_{11}H_{11}^{-1} \;=\; I$ gives

$$\begin{aligned} \frac{\partial H_{11}^{-1}}{\partial x_s} &= -H_{11}^{-1}\; I_s \; H_{11}^{-1} \quad s < 2m. \quad (23) \\ \frac{\partial H_{11}^{-1}}{\partial x_s} &= \mathbf{0} \qquad s \geq 2m \end{aligned}$$

Hence from (13),

$$\frac{\partial D_2}{\partial x_s} \;=\; II_s + V^T I_s V + U^T + U, \qquad (24)$$

where

$$V^T = -H_{21}H_{11}^{-1}, \;\; U = III_s V, \;\; II_s = \frac{\partial H_{22}}{\partial x_s}$$
and $III_s = \frac{\partial H_{21}}{\partial x_s}$,

$II_s$ and $III_s$ are matrices similar to $I_s$ with $II_s$ being an $n-m \times n-m$ matrix which contains ones in one across anti–diagonal and zeros elsewhere, and $III_s$ is an $n - m \times m$ matrix which contains ones in one across anti–diagonal and zeros elsewhere.

Furthermore, differentiating (23), we get

$$\frac{\partial^2 D_2}{\partial x_s \partial x_r} = Y + Y^T,$$

where

$$Y = -Z_r^T H_{11}^{-1} Z_s \quad \textit{and} \quad Z_t = I_t V - III_t^T.$$

Table 1 summarizes the state of the gradient and Hessian of $D_2$ with respect to $x_s$

| $\frac{\partial D_2}{\partial x_s}$ | $Z_t$ | $s$ |
|---|---|---|
| $V^T I_s V$ | $I_t V$ | $0 < s \leq m$ |
| $V^T I_s V + U^T + U$ | $I_t V - III_t^T$ | $m < s < 2m$ |
| $U^T + U$ | $-III_t^T$ | $s = 2m$ |
| $II_s + U^T + U$ | $-III_t^T$ | $2m < s < n+m$ |
| $II_s$ | $\mathbf{0}$ | $n+m<s<2n-1$ |

Table 1: Gradient and Hessian formulas for $D_2$.

Now, let

$$\begin{aligned} W \;&=\; \nabla^2 \mathcal{L}(\mathbf{x}, \Lambda) \\ &=\; \nabla^2 \phi \;-\; \sum_{i,j=m+1}^{n} \lambda_{ij} \nabla^2 d_{ij} \qquad (25) \end{aligned}$$

where $\nabla^2 \phi$ is given by (20) and

$$\sum_{i,j=m+1}^{n} \lambda_{ij} \nabla^2 d_{ij} = \begin{bmatrix} \sum_{i,j} \lambda_{ij} \frac{\partial^2 d_{ij}}{\partial x_1 \partial x_1} & \cdots & \sum_{i,j} \lambda_{ij} \frac{\partial^2 d_{ij}}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \sum_{i,j} \lambda_{ij} \frac{\partial^2 d_{ij}}{\partial x_n \partial x_1} & \cdots & \sum_{i,j} \lambda_{ij} \frac{\partial^2 d_{ij}}{\partial x_n \partial x_n} \end{bmatrix}.$$

Usually, $\nabla^2 \mathcal{L}$ is positive definite, in which case, if $\mathbf{x}^{(k)}$ is sufficiently close to $\mathbf{x}^*$, the basic SQP method converges and the rate is second order (Fletcher [7]). Globally, however (22) may not converge. An algorithm with better convergence properties, when $\mathbf{x}^{(k)}$ is remote from $\mathbf{x}^*$, is suggested by Fletcher at. al. [8] in which the filterSQP can be used to slove (22). Now since the gradient and Hessian are both available, therefore filterSQP can be used to slove the problem.

This description of iterative schemes for solving (22) has so far ignored an important constraint, that, is $D_1 > 0$ in which the varibles $\mathbf{x}^{(k)}$ must permit the matrix $A^{(k)}$ to be factorized as in (12). However, since $m$ is identified correctly and $\mathbf{x}^{(k)}$ is near the solution, this restriction will usually be inactive at the solution. If $\mathbf{x}^{(k)}$ is remote from the solution, additional constraints

$$d_{rr}^{(k)} > 0. \qquad r = 1, 2, \ldots, m$$

are introduced. However, strict inequalities are not permissible in an optimization problem and it is also advisable not to allow $d_{rr}(\mathbf{x}^{(k)})$ to come too close to zero, especially for small $r$, as this is likely to cause the factorization to fail. Hence the constraints

$$md_{rr}^{(k)}/r \geq 0 \qquad r = 1, 2, \ldots, m$$

are added to problem (22). Finally, it is possible that partial factors of the matrix $H^{(k)}$ in the form (12) do not exist for some iterates. In this case, the parameter in the filterSQP method $\rho^{(k+1)} = \rho^{(k)}/4$, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$ and $\Lambda^{(k+1)} = \Lambda^{(k)}$ are chosen for the next iteration in the trust region method.

An advantage of the filterSQP 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 filterSQP method for different values of $m$. If $m^*$ denotes the correct value of $m$ which solves (1), then it is observed that the filterSQP methodd 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 are introduced. Projection methods are globally convergent and hence potentially reliable, but often converge slowly, which can be very inefficient. SQP methods are reliable and have a second order rate of convergence, but require that the correct rank $m^*$ is known. We therefore consider hybrid methods in which the projection algorithm is used sparingly as a way of establishing the correct rank, whilst the filterSQP 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.

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)

```
Algorithm 1 ($F^{(0)} := F, s, \epsilon$):
  repeat projection method
  until $m^{(l)} = m^{(l-j)}$   $j = 1, 2, \ldots, s$.
  repeat
     Apply one iteration of projection method;
      $m^{(0)} := m^{(l)}$;
      $\mathbf{x}^{(0)} := \mathbf{x}^{(l)}$;        ($\mathbf{x}^{(l)}$ from PM)
    repeat fSQPM;
    until $\|D_2(\mathbf{x})\| \leq \epsilon$;
  until $\|\mathbf{x}^{(k)} - \mathbf{x}^{(l)}\| \leq \epsilon$;   ($\mathbf{x}^{(k)}$from fSQPM)
  return ($F^* := F^{(k)}, \mathbf{x}^* := \mathbf{x}^{(k)}, m^* := m^{(k)}$).
```

where PM is the projection method and fSQPM is the filter SQP method. 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)

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

```
        repeat fSQPM;
        until $\|D_2(\mathbf{x})\| \le \epsilon$;
        $\mathbf{x}^{(0)} := \mathbf{x}^{(k)}$;        ($\mathbf{x}^{(k)}$ from fSQPM)
      Apply one iteration of projection method;
        $m^{(0)} := m^{(l)}$;
        $\mathbf{x}^{(0)} := \mathbf{x}^{(l)}$;        ($\mathbf{x}^{(l)}$ from PM)
    until $\|\mathbf{x}^{(k)} - \mathbf{x}^{(l)}\| \le \epsilon$;
    return ($F^* := F^{(k)}, \mathbf{x}^* := \mathbf{x}^{(k)}, m^* := m^{(k)}$).
```

## 5 Numerical Results

In this section, we report our numerical results. Fortran codes have been written to program solver for (1) using filterSQP. Projection computations have been coded in Matlab 5.3.

The results were obtained by applying the methods as follows. Consider the matrix $V$ as a Vandermonde matrix then

$$H = VDV^T, \tag{26}$$

where $D$ is an $m \times m$ diagonal matrix with positive diagonal entries, then $H$ is Hankel positive semi-definite matrix (see [1, 10]). A matrix $H$ was formed from (26) by randomly choosing $m$ weights $d_i$ in matrix $D$, $0 \le d_j \le 1.0$ and $m$ values $x_j, 0 \le x_j \le 1.0$ to determine the Vandermonde matrix $V$. The matrix thus obtained by (26) 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 $1 \times 10^{-5}$.

| | $PA$ | | filterSQP | $A1$ | | | | $A2$ |
|---|---|---|---|---|---|---|---|---|
| $n$ | $m^*$ | $NPI$ | $TNI$ | $s$ | $NPI$ | $m^{(k)}$ | $NI$ | $TNI$ |
| 5 | 3 | 321 | 13 | 5 | 5 | 2 | 9 | 11 |
| 10 | 4 | 447 | 23 | 5 | 6 | 5 | 11 | 25 |
| 15 | 5 | 604 | 32 | 10 | 21 | 7 | 7 | 19 |
| 20 | 7 | 701 | 86 | 20 | 29 | 8 | 17 | 28 |
| 25 | 7 | 828 | 42 | 30 | 121 | 7* | 14 | 21 |
| 30 | 9 | 644 | 57 | 30 | 93 | 9* | 13 | 20 |

Table 2: Comparing the four algorithms. PA: The projection algorithm (Section 2). A1: Hybrid Algorithm 1. A2: Hybrid Algorithm 2. NPI: Number of projection iterations. NI: Number of iterations in the filterSQP method. TNI: Total number of iterations in the filterSQP method.

| | $A2$ | | | |
|---|---|---|---|---|
| $n$ | $m^{(0)}$ | $NL$ in filterSQP | $m^{(k)}$ from OPA | $NL$ in filterSQP |
| 5 | 1 | 5 | 3* | 6 |
| 10 | 2 | 12 | 4* | 13 |
| 15 | 2 | 9 | 5* | 10 |
| 20 | 3 | 11 | 7* | 17 |
| 25 | 3 | 7 | 7* | 14 |
| 30 | 3 | 8 | 9* | 12 |

Table 3: Detailed progress of Algorithm 2. OPA: One iteration of the projection algorithm.

Table 2 summarizes the results for the four different approaches, the projection method, filterSQP method, and the hybrid Algorithms 1 and 2. All four algorithms converge to essentially the same values. Table 2 shows the comparative results for all methods and Table 3 shows the progress of Algorithm 2 in more detail. 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.

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. Thus each iteration is much less expensive than an iteration of the projection method. For the filterSQP method, the initial value $m^{(0)} = 1$, and $m$ is increased by one until the solution is found. The total number of iterations is tabulated. It can be seen that the total number of iterations is much greater than is required by the hybrid methods. Also 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 seen 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 filterSQP method finds the solution rapidly and no further projection steps are needed. Algorithm 2 requires a relatively large number of iterations (see Table 3) in the first call of the filterSQP method, after which one projection step finds the correct rank, and the next call of filterSQP finds the solution in a few iterations. Because the projection steps in Algorithm 1 are relatively expensive, the difference in computing time between these algorithms is not very significant.

**Acknowledgement**. The author is grateful to King Fahd University of Petroleum and Minerals for providing excellent research facilities.

*References:*

[1] Akhiezer N. I, and Krein M. G., *Some Questions in the Theory of Moments,* Transl. Math. Monographs 2, Amer. Math. Soc., Providence, 1962.

[2] Al-Homidan, S. 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] Al-Homidan, S. and Fletcher, R. 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, pp. 1–17, 1995.

[4] Boyle, J. P. and Dykstra, R. L. 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, pp. 28–47, 1986.

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

[6] Fletcher R., Semi-definite matrix constraints in optimization, *SIAM J. Control and Optimization*, V. 23, pp. 493–513, 1985.

[7] Fletcher R., *Practical methods of Optimization,* John Wiley and Sons, Chichester, 1987.

[8] Fletcher R. and Leyffer S., User manual for filterSQP, *Univerity of Dundee Numerical Analysis Report NA/181*, 1999.

[9] Higham N., Computing a nearest symmetric positive semi-definite matrix, *Linear Alg. and Appl.*, V. 103, pp. 103–118, 1988.

[10] Iohvidov I. S., R., *Toeplitz and Hankel Matrices and Forms: Algebraic Theory,* transl. by G. Phillp A. Thijse, Birkhauser, Boston, 1987.

[11] MacInnes C. S., The Solution to a Structured Matrix Approximation Problem Using Grassman Coordinates *SIAM Journal on Matrix Analysis and Applications*, V. 21, pp. 446–453, 1999.

[12] Oh S, and Marks II R. J., Alternating projection onto fuzzy convex sets, *Proc. IEEE*, V. 81, pp. 148–155, 1993.

[13] Ponstein J., A matrix minimization problem involving ranks *Linear Alg. and Appl.*, V. 92, pp. 81–106, 1987.

[14] Shaw A. K., Pokala S., and Kumaresan R., Toeplitz and Hankel matrix approximation using structured approach, *Proc. IEEE*, V. 86, pp. 2349–2352, 1998.

[15] Shaw A. K. and Kumaresan R., Some structured matrix approximation problems, in *Proc. IEEE ICASSP*, V. 4, Internat Conf. Acoustics, Speech, and Signal Proccessing. pp. 2324–2327, 1988.