King Fahd University of Petroleum & Minerals

DEPARTMENT OF MATHEMATICAL SCIENCES

Technical Report Series

TR 259

December 2000

Positive Semi-Definite Toeplitz Approximation Methods

Suliman Al-Homidan

DHAHRAN 31261 ● SAUDI ARABIA ● www.kfupm.edu.sa/math/ ● E-mail: mathdept@kfupm.edu.sa

Positive Semi-Definite Toeplitz Approximation Methods

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

Abstract

In this paper we proposed new methods for solving the positive semi-definite Toeplitz matrix approximation problem. Our approach is based on (i) a projection algorithm which converges globally but slowly; (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, filterSQP method, least distance functions, non-smooth optimization, positive semi-definite matrix, Toeplitz matrix.

1 Introduction

The problem we are interested in is the best approximation of a given matrix by a positive semi-definite symmetric Toeplitz matrix. Toeplitz matrices appear naturally in a variety of problems in engineering. Since positive semi-definite Toeplitz matrices can be viewed as shift-invariant autocorrelation matrices, considerable attention has been paid to them, especially in the areas of stochastic filtering and digital signal processing applications [14] and [20]. Several problems in digital signal processing and control theory require the computation of a positive definite Toeplitz matrix that closely approximates a given matrix. For example, because of rounding or truncation errors incurred while evaluating F, F does not satisfy one or all conditions. Another example in the power spectral estimation of a wide-sense stationary process from a finite number of data, the matrix F formed from the estimated autocorrelation coefficients, is often not a positive definite Toeplitz matrix [19]. In control theory, the Gramian assignment problem for discrete-time single input system requires the computation of a positive definite Toeplitz matrix which also satisfies certain inequality constraints [17]. Consider the following problems:

A) Given a data matrix $F \in \mathbb{R}^{n\times n}$, find the nearest symmetric positive semi-definite Toeplitz matrix A to F that minimizes

$$\text{minimize} \quad \Phi = \|F - A\|_F \tag{1.1}$$

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

B) This problem is similar to problem A except that there is an extra condition on the matrix A, that is, $\text{rank}(A) = m$.

Problem B was studied by Suffridge et. al. [18]. They solve the problem using the self-inversive polynomial $P(x)$. The roots of the derivative of $\frac{P(\zeta)}{z^{n-1}}$ enable them to approximate the data matrix. They also solve problem A using the ideas of a modified alternating projection algorithm that was successfully used in solving similar approximation problems for distance matrices [2]. In [11], alternating convex projection techniques are used to solve problem B. Oh et. al. [16] use alternating projection onto fuzzy convex sets when three or more convex sets do not intersect. Toeplitz matrix approximations are also discussed in [4, 5] and [15].

In Section 2, problem A is solved using the von Neumann algorithm. In Section 3, problem B is formulated as a nonlinear minimization problem with positive semi-definite Toeplitz matrix as constraints, where a constraints formulation is also given. Then an algorithm with rapid convergence is obtained by the filter Sequential Quadratic Programming (SQP) method [10]. In Section 4, two new hybrid methods are described to solve problem A: firstly, there is Algorithm 1, which starts with the projection method to determine the rank $m^{(k)}$ and continues with the filterSQP method; and secondly, Algorithm 2 is described which solves the problem by the filterSQP method and uses the projection method to update the rank. Numerical compressions are reported in Section 5.

A symmetric Toeplitz matrix A is denoted by

$$A = \begin{bmatrix} a_1 & a_2 & \dots & a_n \\ a_2 & a_1 & \dots & a_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n-1} & \dots & a_1 \end{bmatrix} = \text{Toeplitz}(a_1, a_2, \dots, a_n). \tag{1.2}$$

2 The Projection Algorithm

In this section, we describe a projection algorithm for solving problem A. This algorithm is derived from an alternating projection algorithm due to Dykstra [6] for finding the least distance from a fixed point to an intersection of convex sets. This algorithm is given independently by Han [12]. An important feature of this algorithm is the generation of formulae for certain projection maps that are needed.

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

$$\mathbf{f}^{(k+1)} = \mathbf{f}^{(k)} + P_m(\ldots P_1(\mathbf{f}^{(k)})\ldots) - P_1(\mathbf{f}^{(k)}). \tag{2.1}$$

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 [3] 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 [1]).

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

$$K_{\mathrm{R}} = \{A : A \in \mathbb{R}^{n\times n},\ A^T = A \text{ and } \mathbf{z}^T A\mathbf{z} \geq 0 \quad \forall\, \mathbf{z} \in \mathbb{R}^n\} \tag{2.2}$$

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

$$K_{\mathrm{T}} = \{A : A \in \mathbb{R}^{n\times n},\ A \text{ is Toeplitz}\}, \tag{2.3}$$

which is a subspace of dimension n.

In applying the Dykstra-Han algorithm to the Toeplitz matrix approximation, it is appropriate to use the Frobenius matrix norm, and to express (1.1) as

$$\begin{aligned} &\text{minimize} && \|F - A\|_F \\ &\text{subject to} && A \in K_{\mathbf{R}} \cap K_T, \end{aligned} \tag{2.4}$$

where $K_{\mathbf{R}}$ and K_T are given by (2.2) and (2.3), respectively.

To apply algorithm (2.1), we need formulae for the projection maps $P_{\mathbf{R}}(.)$ and $P_T(.)$, corresponding, respectively, to $P_1(.)$ and $P_2(.)$ in (2.1). These are the maps from $K = \{A :\ A \in \mathbb{R}^{n\times n}\}$ on to $K_{\mathbf{R}}$ and K_T. The projection map $P_{\mathbf{R}}(F)$ formula on to $K_{\mathbf{R}}$ is given by [13]

$$P_{\mathbf{R}}(F) = U\Lambda^+U^T, \tag{2.5}$$

where

$$\Lambda^+ = \begin{bmatrix} \Lambda_m & 0 \\ 0 & 0 \end{bmatrix}, \tag{2.6}$$

and $\Lambda_m = \operatorname{diag}[\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_m]$ is the diagonal matrix formed from the positive eigenvalues of F.

The projection map $P_T(F)$ formula on to K_T is given by

$$P_T(F) = \text{Toeplitz}(t_1, t_2, \ldots, t_n), \tag{2.7}$$

where

$$t_{k+1} = \frac{1}{2(n-k)} \sum_{i=1}^{n-k} (f_{i\ i+k} + f_{i+k\ i}), \qquad k = 1, 2, \ldots, n. \tag{2.8}$$

We can now use the projection maps $P_{\mathbf{R}}(F)$ and $P_T(F)$ given by (2.5) and (2.7) to implement the Dykstra-Han algorithm (2.1). Given a distance matrix $F \in \mathbb{R}^{n\times n}$, the algorithm is initialized by $F^{(0)} = F$ and the iteration formula is

$$F^{(k+1)} = F^{(k)} + (P_T(P_{\mathbf{R}}(F^{(k)}))) - P_{\mathbf{R}}(F^{(k)}). \tag{2.9}$$

The sequences $\{P_{\mathbf{R}}(F^{(k)})\}$ and $\{P_T(P_{\mathbf{R}}(F^{(k)}))\}$ converge to the solution A^* of (2.4) and hence (1.1). This algorithm was also given by [18] and [11] in a similar manner.

3 The SQP Algorithms

In the previous section, the alternating projection algorithm computes a unique solution for problem A. It is the loss of convexity of the sets $K_{\mathbf{R}}$ and K_T that increases the difficulty of problem B. In this section, We use techniques related to FilterSQP [10] for solving nonlinear programming problems in order to develop an algorithm to solving problem B.

It is difficult to deal with the matrix cone constraints in (2.4) since it is not easy to specify if the elements are feasible. Using partial LDL^T factorization of A, this difficulty can be overcome. Since m, the rank of A^*, is known, therefore for A sufficiently close to A^*, the partial factors $A = LDL^T$ can be calculated such that

$$L = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix}, \quad A = \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix}, \tag{3.1}$$

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

$$D_2(A) \;=\; A_{22} \;-\; A_{21}A_{11}^{-1}A_{21}^T. \tag{3.2}$$

Now, if the structure of the matrix A is in a Toeplitz form, i.e.

$$A = \text{Toeplitz}(x_1, x_2, \ldots, x_n), \tag{3.3}$$

then (3.2) enables the constraint $A \in K$ to be written in the form

$$D_2(A(\mathbf{x})) = 0. \tag{3.4}$$

Hence, problem B can now be expressed as

$$\begin{array}{ll} \text{minimize} & \Phi \\ \text{subject to} & D_2(A) \;=\; 0 = Z^TAZ, \end{array} \tag{3.5}$$

where $Z = \begin{bmatrix} -A_{11}^{-1}A_{21}^T \\ I \end{bmatrix}$ is the basis matrix for the null space of A when $D_2 = 0$. The Lagrange multiplier for the constraint (3.4) is Λ relative to the basis Z and the Lagrangian for porblem (3.5) is

$$\mathcal{L}(\mathbf{x}^{(k)}, \Lambda^{(k)}) \;=\; \Phi - \Lambda : Z^TAZ. \tag{3.6}$$

This approach has been studied in a similar way by [8].

The structure of the Toeplitz matrix A as given in (3.3), is

$$\Phi = \sum_{i,j=1}^{n} (f_{ij} - x_{|i-j+1|})^2, \tag{3.7}$$

and $\nabla\Phi = (\partial\Phi/\partial x_1, \ldots, \partial\Phi/\partial x_n)^T$ where ∇ denotes the gradient operator $(\partial/\partial x_1, \ldots, \partial/\partial x_n)^T$. Therefore,

$$\frac{\partial\Phi}{\partial x_1} = 2\sum_{i=1}^{n}(x_1 - f_{ii}),$$

and

$$\frac{\partial\Phi}{\partial x_{s+1}} = 2\{\sum_{i=1}^{n-s}(x_{s+1} - f_{i+s,i}) + (x_{s+1} - f_{i,i+s})\},$$

where $s = 1, \cdots, n-1$. Differentiating gives

$$\frac{\partial^2\Phi}{\partial x_1^2} = 2(n), \qquad \frac{\partial^2\Phi}{\partial x_{s+1}^2} = 4(n-s) \quad s = 1, \cdots, n-1, \tag{3.8}$$

and

$$\frac{\partial^2\Phi}{\partial x_r \partial x_s} = 0 \quad \text{if} \quad r \neq s, \tag{3.9}$$

where $s, r = 1, \cdots, n$.

The advantage of formula (3.4) is that expressions for both the first and second derivatives of the constraints with respect to the elements of A can be obtained. The simple form of (3.2) is utilized by writing the constraints $D_2(A) = 0$ in the following form:

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

where $i, j = m+1, \cdots, n$ and $[A_{11}^{-1}]_{kl}$ denote the element of A_{11}^{-1} in kl-position.

Thus (3.5) can be expressed as

$$\begin{aligned} &\text{minimize} \quad \Phi = \sum_{i,j=1}^{n} (f_{ij} - x_{|i-j+1|})^2, \\ &\text{subject to} \;\; d_{ij}(\mathbf{x}) = 0. \end{aligned} \tag{3.10}$$

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 (3.10), it is necessay to derive the 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

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

where the "1" appears in the s-position. Hence the matrix I_s is a matrix that contains "1"s in two off diagonal and zeros elsewhere. Now, differentiating $A_{11}A_{11}^{-1} = I$ gives

$$\frac{\partial A_{11}^{-1}}{\partial x_s} = - A_{11}^{-1}\, I_s\, A_{11}^{-1}. \tag{3.11}$$

Hence from (3.2)

$$\frac{\partial D_2}{\partial x_s} = II_s + V^T I_s V + U_s^T + U_s \quad \text{and} \quad \frac{\partial^2 D_2}{\partial x_s \partial x_r} = Y + Y^T,$$

where

$$II_s = \frac{\partial A_{22}}{\partial x_s}, \qquad V^T = -A_{21}^T A_{11}^{-1},$$

$$U_s = III_s V, \qquad III_s = \frac{\partial A_{21}}{\partial x_s},$$

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

II_s, III_s are matrices similar to I_s with II_s being an $n-m \times n-m$ matrix which contains "1"s in two off-diagonal and zeros elsewhere, and III_s is an $n-m \times m$ matrix which contains "1"s in one off-diagonal and zeros elsewhere.

Now, let

$$\begin{aligned} W &= \nabla^2 \mathcal{L}(\mathbf{x}, \Lambda) \\ &= \nabla^2 \Phi - \sum_{i,j=m+1}^{n} \lambda_{ij} \nabla^2 d_{ij} \end{aligned} \tag{3.12}$$

where $\nabla^2 \Phi$ is given by (3.8) and (3.9) and

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

Now since the gradient and Hessian are both avalible, therefore the filterSQP can be used to slove (3.10).

This description of iterative schemes for solving (3.10) 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 (3.1). However, if 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

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

are added to problem (3.10). Finally, it is possible that partial factors of the matrix $A^{(k)}$ in the form (3.1) do not exist for some iterates. In this case, the parameters 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.

4 Hybrid Methods

A combination of both algorithms are introduced in Sections 2 and 3. 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 ϵ some small number. Then the following algorithm solves problem A

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

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 ϵ be some small number; also choose $m^{(0)}$ as a small integer number. Then the following algorithm solves problem A

Algorithm 2 $(F^{(0)} := F, m^{(0)}, \epsilon)$:
repeat
repeat filter SQP method;
until $\|D_2(\mathbf{x})\| \le \epsilon;$

$\mathbf{x}^{(0)} := \mathbf{x}^{(k)}$; ($\mathbf{x}^{(k)}$ from filter SQP method)
Apply one iteration of projection method;
$m^{(0)} := m^{(l)}$;
$\mathbf{x}^{(0)} := \mathbf{x}^{(l)}$; ($\mathbf{x}^{(l)}$ from projection method)
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 some numerical experiments with the proposed methods.

First, for testing the algorithms described above, the following example is considered: Consider problem (3.10) in which

$$F = \begin{bmatrix} 3 & 2 & 3 & 4 \\ 5 & 7 & 2 & -1 \\ 6 & 2 & 5 & 4 \\ 5 & 3 & 1 & 2 \end{bmatrix}, \quad \text{with } m = 1. \tag{5.1}$$

Let

$$A^* = \begin{bmatrix} x & y & z & u \\ y & x & y & z \\ z & y & x & y \\ u & z & y & x \end{bmatrix}. \tag{5.2}$$

In general, when $n = 4$, the number of constraints is six, of which three are:

$$d_{22} = x^2 - y^2 = 0 \implies x = y, \quad d_{33} = x^2 - z^2 = 0 \implies x = z,$$

and

$$d_{44} = x^2 - u^2 = 0 \implies x = u.$$

Therefore

$$x = y = z = u,$$

and this satisfies the rest of the constraints. Hence the problem will be reduced to minimizing

$$\Phi = 16x^2 - 106x + c, \tag{5.3}$$

where c is a constant. Thus the minimum value of $\Phi = 7.8382$ for $x = 106/32$. However, if the required rank is two, then we have three new constraints $d_{33} = 0$, $d_{44} = 0$ and $d_{34} = 0$. One of these constraints is

$$d_{33} = x^3 - 2xy^2 - xz^2 + 2zy^2 = 0 \implies x = z.$$

This reduces the next constraint to

$$d_{44} = x^3 - xy^2 - xz^2 - xu^2 + 2yzu = (y - u)^2 = 0 \implies y = u,$$

and this satisfies the constraint d_{34}. Hence the problem will be reduced to minimizing

$$\Phi = 8x^2 - 56x + 8y^2 - 50y + c \tag{5.4}$$

where c is a constant. Thus the minimum value of $\Phi = 7.8022$ for $x = z = 56/16$ and for $y = u = 50/16$.

Now if the rank is three then the filterSQP is used and we find that the minimum value of $\Phi = 7.1707$ for $x = 4.3345$ $y = 2.6714$ $z = 2.7428$ $u = 4.3314$ and this is identical to the projection algorithm.

The algorithm has been tested on randomly generated matrices with values distributed between 10^{-3} and 10^{3}. Thus to test the efficiency of the algorithm, a Fortran codes have been written to program solver for (3.10) using the filterSQP. Projection computations have been coded in Matlab 5.3 and executed on SUN workstation. The termination criterion for algorithm is $\|F^{(k)} - F^{(k-1)}\| < 10^{-5}$. All four algorithms converge to essentially the same values. Table 1 summarizes the results for the four different approaches, the projection method, the SQP Algorithm, and the hybrid Algorithms 1 and 2. All four algorithms converge to essentially the same values. An asterisk indicates where the correct rank has been identified. In some cases, in particular, with Algorithm 1, the final rank is $r^* + 1$ or $r^* + 2$ but the solution is within the required tolerance.

	PA		filter		A1				A2				ϕ
n	r^*	NI	$r^{(0)}$	nq	s	NI	$r^{(k)}$	nq	$r^{(0)}$	nq	$r^{(k)}$ PA	nq	
4	1	55	1	10	3	3	2(1*)	14	1*	10			2644.1
4	3	58	1	36	3	5	3*	12	1	16	3*	11	2656.5
5	4	245	2	29	5	11	4*	13	2	15	4*	12	4013.6
6	3	250	2	28	5	6	4(3*)	14^+	2	13	3*	14	5741.2
7	6	78	3	35	6	9	6*	17	3	15	5*	16	6059.3
8	6	44	3	49	6	8	6*	29	3	19	6*	22	6591.4
8	5	356	3	39	6	12	5*	18	3	15	5*	17	8270.9
10	6	140	3	73	6	9	8(6*)	49	3	27	6*	34	9769.8
15	10	2661	5	64	10	15	11(10*)	21	5	18	10*	15	14274
20	15	272	7	79	13	22	15*	34	7	29	15*	31	19860

Table 1: Comparing the four methods.
PA: Projection Algorithm. A1: Algorithm 1. A2: Algorithm 2.
NI: Number of iteration in projection algorithm.
nq: Number of quadratic programming problem solved.

For the projection algorithm, each iteration involves an eigensolution, which entails relatively expensive $O(n^3)$ calculations. 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 filter-SQP algorithm, the initial value $r^{(0)}$ is tabulated, and r is increased by one until the solution is found. The total number of iterations is tabulated, and within this figure, it is found that fewer iterations are required as r increases. It can be seen that the total number of iterations is much greater than is required by the hybrid methods. Also the initial value $r^{(0)}$ is rather arbitrary: a smaller value of $r^{(0)}$ would have given an even larger number of line searches.

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 r^* increases. Once the projection iteration has settled down, the filter-SQP method finds the solution rapidly and no further projection steps are needed. Algorithm 2 requires a relatively large number of iterations in the first call of the filter-SQP method, after which one projection step finds the correct rank, and the next call of filter-SQP finds the solution in a few iterations. This is because of the good initial starting vector $\mathbf{x}$ given by the projection method. Because the projection steps in Algorithm 1 are relatively expensive, the difference in computing time between these algorithms is not very significant.

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

References

[1] 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.

[2] 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.

[3] 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.

[4] Cabay S. and Meleshko R. A Weakly Stable Algorithm for Pade' Approximants and the Inversion of Hankel Matrices, *SIAM J. Matrix Analysis and Appl., V. 14 pp. 735-765.* 1992.

[5] Cybenko G., Moment problems and low rank Toeplitz Approximations, *Circuits Systems Signal process V. 1, pp. 345-365*, 1982.

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

[7] Fletcher R., Numerical experiments with an exact filter penalty function method, *Nonlinear Programming 4,* (Eds. O. L. Mangasarian, R. R. Meyer and S. M. Robinson), Academic Press, New York, 1981.

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

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

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

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

[12] Han, S. P. A successive projection method, *Math. Programming, 40, pp. 1-14,* 1988.

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

[14] Kailath T. A view of three decades of linear filtering theory, *IEEE Trans. Inf. Th., It-20, pp. 145-181,* 1974.

[15] Kung S. Y., Toeplitz approximation method and some applications, in *Internat, Sympos. On Mathematical Theory of Networks and Systems V. IV,* Western Periodicals Co., Morth Hollywood, CA, 1981.

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

[17] Skelton R. E., The jury test and covariance control in *Proc Symp. Fundamentals Discrete-Time Syst.* (Chicago), 1992.

[18] Suffridge Y. J. and Hayden T. L., Approximation by a hermitian positive semidefinite Toeplitz matrix, *SIAM J. Matrix Analysis and Appl., V. 14 pp. 721-734,* 1993.

[19] Therrien C. W., *Discrete Random Signals and Statistical signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1992.

[20] Willsky A. S. *Digital signal processing and control and estimation theory,* MIT Press, Cambridge, Mass, 1979.