Semidefinite and Second-Order Cone Optimization Approach for the Toeplitz Matrix Approximation Problem

Suliman Al-Homidan*

December 19, 2005

Abstract

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

Contents

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

1 Introduction

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, 7] and [27]. 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 the data matix, it 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 data matrix formed from the estimated autocorrelation coefficients, is often not a positive definite Toeplitz matrix [23]. 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 [20].

Our work is mainly casting the problem: first as a semidefinite programming problem and second as a mixed semidefinite and second-order cone optimization problem. A semidefinite programming (SDP) problem is to minimize a linear objective function subject to constraints over the cone of positive semidefinite matrices. SDP problems are of great interest due to many reasons , e.g., SDP contains important classes of problems as special cases, such as linear and quadratic programming. Applications of SDP exist in combinatorial optimization, approximation theory, system and control theory, and mechanical and electrical engineering. SDP problems can be solved very efficiently in polynomial time by interior point algorithms [25, 28, 5, 18].

The constraints in a mixed semidefinite and second-order cone optimization problem are constraints over the positive semidefinite and the second-order cones. Although the second-order cone constraints can be seen as positive semidefinite constraints, recent research has shown that it is more effecient to deal with mixed problems rather than the semidefinite programming problem. Nesterov et. al. [18] can be considered as the first paper to deal with mixed semidefinite and second-order cone optimization problems. However, the area was really brought to life by Alizadeh et al. [4] with the introduction of SDPPack, a software package for solving optimization problems from this class. The practical importance of second-order programming was demonstrated by Lobo et al. [16] and many subsequent papers. In [21] Sturm presented implementational issues of interior point methods for mixed SDP and SOCP problems in a unified framework. One class of these interior point methods is the primal-dual path-following methods. These methods are considered the most successful interior point algorithms for linear programming. Their extension from linear to semidefinite and then mixed problems has followed the same trends. One of the successful implementation of primal-dual path-following methods is in the software SDPT3 by Toh et al. [24].

A similar problem to our problem was studied by Suffridge et. al. [22]. They solve the problem using the self-inversive polynomial $P(x)$. The roots of the derivative of $\frac{P(z)}{z^{n-1}}$ enable them to approximate the data matrix. They also solve the problem using the ideas of a modified alternating projection algorithm that was successfully used in solving similar approximation problems for distance matrices [3]. In [11], alternating convex projection techniques are used to solve the problem. Toeplitz matrix approximations are also discussed in [7, 15].

In [2] a similar problem is studied. One approach followed is a projection algorithm which converges globally but the rate of convergence is very slow. Another approach is the quasi-Newton method which is faster. Then a hybrid method to combine the best features of both is used. A similar problem which requires the knowledge of the rank was studied in [1, 6] and formulated as a nonlinear minimization problem and then solved using techniques related to filterSQP [9].

1.1 Notation

Throughout this paper, we will denote the set of all $n \times n$ real symmetric matrices by $\mathcal{S}^n$, the cone of the $n \times n$ real symmetric positive semidefinite matrices by $\mathcal{P}$ and the second-order cone of dimension k by $\mathcal{Q}_k$, and is defined as

$$\mathcal{Q}_k = \{x \in \mathbb{R}^k : \|x_{2:k}\|_2 \leq x_1\},$$

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

$$T(x) = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ x_2 & x_1 & \cdots & x_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n-1} & \cdots & x_1 \end{bmatrix}, \qquad x \in \mathbb{R}^n.$$

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

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

Here $U \bullet V = \text{trace}(UV) = \sum_{i,j}^n U_{ij}V_{ij}$ and vec$\,(U)$ stands for the vectorization operator found by stacking the columns of U together. The symbols $\succeq$, $\geq_Q$ and $\geq$ will be used to denote the partial orders induced by $\mathcal{P}$, $\mathcal{Q}_k$ on $\mathcal{S}^n$ and $\mathbb{R}^k$, respectively. That is,

$$U \succeq V \quad \Leftrightarrow \quad U - V \in \mathcal{P}, \;\; \forall\, U,\; V \in \mathcal{S}^n$$

and

$$u \geq_Q v \Leftrightarrow \quad u - v \in \; \mathcal{Q}_k, \;\; \forall\, u,\; v \;\in \mathbb{R}^k.$$

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

1.2 The Problem and Outline

Our problem in mathematical notation can, now, be formulated as follows: Given a data matrix $F \in \mathbb{R}^{n\times n}$, find the nearest positive semidefinite Toeplitz matrix $T(x)$ to F such that $\|F - T(x)\|_F^2$ is minimal. Thus, we have the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \|F - T(x)\|_F^2 \\ \text{subject to} & T(x) \in \mathcal{T}, \\ & T(x) \succeq 0. \end{array} \tag{1.2}$$

The alternating projection method is described briefly in Section 2; since it converges to the optimal solution globally. However, the rate of convergence is slow. A brief description of semidefinite and second-order cone optimization problems along with reformulations of problem (1.2) in the form of the respective class will be given in Sections 3 and 4, respectively. Numerical results, showing the performance of the projection method against the primal-dual path-following method acting on our formulations, will be reported in Section 5.

2 The projection Method

The method of successive cyclic projections onto closed subspaces C_i's was first proposed by von Neumann [19] as an extension of the method of Kaczmarz [13], then independently by Wiener [26]. 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 $k = 1, 2, 3, \ldots$
    $X_{k+1} = P_1(P_2(X_k))$.

X_k converges to the near point to D in $C_1 \cap C_2$, where P_1 and P_2 are the orthogonal projections on C_1 and C_2, respectively. Dykstra [8] modified von Neumann's algorithm to handle the situation when C_1 and C_2 are replaced by convex sets. Other proofs and connections to duality along with applications were given in Han [12]. The modified Neumann's algorithm when applied to (1.2) yields:

Algorithm 2.2 *Modified Alternating Projection Algorithm*
Let $F_1 = F$

For $j = 1, 2, 3, \ldots$
$F_{j+1} = F_j + [P_{\mathcal{P}}(P_{\mathcal{T}}(F_j)) - P_{\mathcal{T}}(F_j)]$

Then $\{P_{\mathcal{T}}(F_j)\}$ and $P_{\mathcal{P}}(P_{\mathcal{T}}(F_j))$ converge in Frobenius norm to the solution. Here, $P_T(F)$ is the orthogonal projection onto the subspace of Toeplitz matrices $\mathcal{T}$. It is simply setting each diagonal to be the average of the corresponding diagonal of F. $P_{\mathcal{P}}(F)$ is the projection of F onto the convex cone of positive semidefinite symmetric matrices. Simply $P_{\mathcal{P}}(F)$ is finding the spectral decomposition of F and setting the negative eigenvalues to zero.

3 Semidefinite Programming Approach

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

$$\begin{array}{ll} \text{minimize} & C \bullet X \\ \text{subject to} & A_i \bullet X = b_i, \quad i = 1, \cdots, m \\ & X \succeq 0. \end{array} \tag{3.1}$$

where all A_i, $C \in \mathcal{S}^n$, $b \in \mathbb{R}^m$ are given, and $X \in \mathcal{S}^n$ is the variable. This optimization problem (3.1) is a convex optimization problem since its objective and constraint are convex. The dual problem of (3.1) is

$$\begin{array}{ll} \text{minimize} & b^T y \\ \text{subject to} & \sum_{i=1}^{m} y_i A_i \preceq C \end{array} \tag{3.2}$$

where $y \in \mathbb{R}^m$ is the variable. Although (3.1) and (3.2) seem to be quite specialized, it includes, as we said before, many important problems as special cases. It also appears in many applications. One of these applications is problem (1.2) as we will show now.

Theorem 3.1 (Schur Complement) *If*

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

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

This matrix $C - B^T A^{-1} B$ is called the schur complement of A in M. Letting $\|F - T(x)\|_F^2 \leq \alpha$, α is a nonnegative real scalar and noting that:

$$\|F - T(x)\|_F^2 = \text{vec}^T(F - T(x))\text{vec}\,(F - T(x)),$$

we have:

$$\begin{aligned} & \text{vec}^T(F - T(x))\text{vec}\,(F - T(x)) \leq \alpha \\ \Leftrightarrow \quad & \alpha - \text{vec}^T(F - T(x)) I \text{vec}\,(F - T(x)) \geq 0 \\ \Leftrightarrow \quad & \begin{pmatrix} I & \text{vec}\,(F - T(x)) \\ \text{vec}^T(F - T(x)) & \alpha \end{pmatrix} \succeq 0 \end{aligned} \tag{3.3}$$

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

$$\begin{aligned} \text{(SDV)} \quad \text{minimize} \quad & \alpha \\ \text{subject to} \quad & \\ & \begin{pmatrix} \alpha & 0 & 0 \\ 0 & T(x) & 0 \\ 0 & 0 & V \end{pmatrix} \succeq 0, \end{aligned} \tag{3.4}$$

where

$$V = \begin{pmatrix} I & \text{vec}\,(F - T(x)) \\ \text{vec}^T(F - T(x)) & \alpha \end{pmatrix}$$

which is an SDP problem in the dual form (3.2) with block dimensions $n+1$ and n^2+n+2, SDP problem (3.4) is very large even for a small data matrix F. For example, a 50×50 matrix F will give rise to a problem with dimensions 51 and 2552, hence solving (1.2) using formula (3.4) is not efficeint. Furthermore, we do not exploit the structure of $T(x)$ being symmetric Toeplitz . Which leads to another way of formulation that produces an SDP problem with reasonable dimensions and exploits the symmetric Toeplitz structure of $T(x)$. This can be done by means of the following isometry operator:

Definition 3.2 *Let* tvec $: \mathcal{T} \longrightarrow \mathbb{R}^n$ *be defined as* $\text{tvec}\,(T(u)) = [\sqrt{n}u_1 \; \sqrt{2(n-1)}u_2 \; \sqrt{2(n-2)}u_3 \; \cdots \; \sqrt{2}u_n]^T$ *for any* $T(u) \in \mathcal{T}$.

tvec is a linear operator from $\mathcal{T}$ to $\mathbb{R}^n$, which satisfy the following characterizations:

Corollary 3.3 *For any* $u, v \in \mathbb{R}^n$

1. $T(u) \bullet T(v) = \text{tvec}^T(T(u))\text{tvec}(T(v))$.

2. $\|T(u) - T(v)\|_F^2 = \text{tvec}^T(T(u) - T(v))\text{tvec}(T(u) - T(v))$. □

Part 1 implies that tvec is an isometry. To take the advantage of the isometry operator tvec, we need F to be Toeplitz. If we project F onto $\mathcal{T}$ to get $P_{\mathcal{T}}(F)$. The following lemma show that the nearest symmetric Toeplitz positive semidefinite matrix to F is exactly equal to the nearest symmetric Toeplitz positive semidefinite matrix to $P_{\mathcal{T}}(F)$.

Lemma 3.4 *Let $T(x)$ be the nearest symmetric Toeplitz positive semidefinite matrix to $P_{\mathcal{T}}(F)$, then $T(x)$ is so for F.*

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

$$(T(x) - P_{\mathcal{T}}(F)) \bullet (P_{\mathcal{T}}(F) - F) = 0$$

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

$$\|T(x) - F\|_F^2 = \|T(x) - P_{\mathcal{T}}(F)\|_F^2 + \|P_{\mathcal{T}}(F) - F\|_F^2$$

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

As a consequence of this lemma, (1.2) equivalent to :

$$\begin{array}{ll} \text{minimize} & \|P_{\mathcal{T}}(F) - T(x)\|_F \\ \text{subject to} & T(x) \in \mathcal{T} \\ & T(x) \succeq 0. \end{array} \tag{3.5}$$

3.1 Formulation I (SDT)

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

$$\begin{array}{lll} & \|P_{\mathcal{P}}(F) - T(x)\|_F^2 \leq \alpha & \\ \Leftrightarrow & \text{tvec}^T(P_{\mathcal{P}}(F) - T(x))\text{tvec}(P_{\mathcal{P}}(F) - T(x)) \leq \alpha & \text{by Corollary 3.3} \\ \Leftrightarrow & \alpha - \text{tvec}^T(P_{\mathcal{P}}(F) - T(x)) I \text{tvec}(P_{\mathcal{P}}(F) - T(x)) \geq 0 & \\ \Leftrightarrow & \begin{pmatrix} I & \text{tvec}(P_{\mathcal{P}}(F) - T(x)) \\ \text{tvec}^T(P_{\mathcal{P}}(F) - T(x)) & \alpha \end{pmatrix} \succeq 0 & \text{by Theorem 3.1.} \end{array}$$

Hence, we have the following SDP problem:

$$\begin{array}{lll} \text{(SDT)} & \text{minimize} & \alpha \\ & \text{subject to} & \\ & & \begin{pmatrix} \alpha & 0 & 0 \\ 0 & T(x) & 0 \\ 0 & 0 & \hat{V} \end{pmatrix} \succeq 0, \end{array} \tag{3.6}$$

where

$$\hat{V} = \begin{pmatrix} I & \text{tvec}\,(F - T(x)) \\ \text{tvec}^T(F - T(x)) & \alpha \end{pmatrix}.$$

This SDP problem has block dimensions $n+1$ and $2n+2$ which is far better than (3.4).

3.2 Formulation II (SDQ)

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

$$\|F - T(x)\|_F^2 = x^T P x + 2q^T x + \beta,$$

where

$$\begin{aligned} P &= 2\text{diag}([\frac{n}{2} \quad n-1 \quad n-2 \ \cdots \ 1]), \\ q_1 &= -\sum_{i=1}^{n} F_{ii}, \\ q_k &= -\sum_{i=k}^{n} (F_{i-k+1,i} + F_{i,i-k+1}), \quad k = 2, \cdots n \ \text{ and} \\ \beta &= \|F\|_F^2. \end{aligned}$$

Now, we have for a nonnegative real scalar α

$$\begin{aligned} & \|F - T(x)\|_F^2 \le \alpha \\ \Leftrightarrow \quad & x^T P x + 2q^T x + \beta \le \alpha \\ \Leftrightarrow \quad & (P^{1/2}x)^T (P^{1/2}x) + 2q^T x + \beta \le \alpha \\ \Leftrightarrow \quad & \alpha - 2q^T x - \beta - (P^{1/2}x)^T I (P^{1/2}x) \ge 0 \\ \Leftrightarrow \quad & \begin{pmatrix} I & (P^{1/2}x) \\ (P^{1/2}x)^T & \alpha - 2q^T x - \beta \end{pmatrix} \succeq 0. \end{aligned}$$

Hence, we have the following SDP problem:

$$\begin{array}{lll}(\text{SDQ}) & \text{minimize} & \alpha \\ & \text{subject to} & \\ & & \begin{pmatrix} \alpha & 0 & 0 \\ 0 & T(x) & 0 \\ 0 & 0 & Q \end{pmatrix} \succeq 0, \end{array} \qquad (3.7)$$

where

$$Q = \begin{pmatrix} I & (P^{1/2}x) \\ (P^{1/2}x)^T & \alpha - 2q^T x - \beta \end{pmatrix},$$

This SDP problem is of block dimenstions $n+1$ and $2n+2$. Although problem (3.7) has the same dimentions as problem (3.6), it is less efficient to solve it over the positive semidefinite cone $\mathcal{P}$, especially when we have large size F. In practice, as we will see in Section 5, it has been found that the performance of this formulation is poor. The reason for that is the matrix P being of full rank. A more efficient interior point method for this formulation can be developed by using Nestrov and Nemirovsky formulation as a problem over the second-order cone [17, Section 6.2.3]. This what we will see in the next section.

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

4 Mixed Semidefinite and Second-Order Cone Approach:

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

$$
\begin{array}{lll}
(P') & \text{minimize} & C_S \bullet X_S + C_Q^T X_Q + C_L^T X_L \\
 & \text{subject to} & (A_S)_i \bullet X_S + (A_Q)_i^T X_Q + (A_L)_i^T X_L = b_i, \quad i = 1, \cdots, m \\
 & & X_S \succeq 0, X_S \geq_Q 0, X_L \geq 0
\end{array}
\tag{4.1}
$$

where $X_S \in \mathcal{S}^n$, $X_Q \in \mathbb{R}^k$ and $X_L \in \mathbb{R}^{n_L}$ are the variables. C_S, $(A_S)_i$ $\in \mathcal{S}^n$, $\forall i$ C_Q, $(A_Q)_i \in \mathbb{R}^k$ $\forall i$ and C_L, $(A_L)_i \in \mathbb{R}^{n_L}$ $\forall i$ are given data. It is possible that one or more of the three parts of (4.1) is not present. If the second-order part is not present, then (4.1) reduces to the ordinary SDP (3.1) and if the semidefinite part is not present, then (4.1) reduces to the so-called convex quadratically constrained linear programming problem.
The standard dual of (4.1) is:

$$
\begin{array}{lll}
(D') & \text{maximize} & b^T y \\
 & \text{subject to} & \sum_{i=1}^m y_i (A_S)_i \preceq C_S \\
 & & \sum_{i=1}^m y_i (A_Q)_i \leq_Q C_Q \\
 & & \sum_{i=1}^m y_i (A_L)_i \leq C_L.
\end{array}
\tag{4.2}
$$

Here, $y \in \mathbb{R}^m$ is the variable.

In our setting, we may drop the third part of the constraints in (4.1) and its dual (4.2), since we do not have explicit linear constraints. One natural claim can be made here: In (1.2) the objective function can be recast as a dual SQLP in three different ways.

4.1 Formulation III (SQV)

One way to minimize $\|F - T(x)\|_F^2$ is to minimize $\|F - T(x)\|_F = \|\text{vec}\,(F - T(x))\|_2$. So, if we put $\|F - T(x)\|_F \leq \alpha$ for $\alpha \in \mathbb{R}^+$, then by the definition of the second-order cone, we have

$$\begin{pmatrix} \alpha \\ \text{vec}\,(F - T(x)) \end{pmatrix} \in \mathcal{Q}_{1+n^2}$$

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

$$\begin{array}{lll} \text{(SQV)} & \text{minimize} & \alpha \\ & \text{subject to} & \\ & & \begin{pmatrix} \alpha & 0 \\ 0 & T(x) \end{pmatrix} \succeq 0 \\ & & \begin{pmatrix} \alpha \\ \text{vec}\,(F - T(x)) \end{pmatrix} \geq_Q 0. \end{array} \tag{4.3}$$

4.2 Formulation IV (SQQ)

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

$$\|F - T(x)\|_F^2 = x^T P x + 2q^T x + \beta \tag{4.4}$$

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

$$\begin{array}{ll} \text{minimize} & x^T P x + 2q^T x + \beta \\ \text{subject to} & T(x) \in \mathcal{T} \\ & T(x) \succeq 0. \end{array} \tag{4.5}$$

But

$$x^T P x + 2q^T x + \beta = \|P^{1/2} x + P^{-1/2} q\|_2^2 + \beta - q^T P^{-1} q$$

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

$$\begin{array}{lll} \text{(SQQ)} & \text{minimize} & \alpha \\ & \text{subject to} & \\ & & \begin{pmatrix} \alpha & 0 \\ 0 & T(x) \end{pmatrix} \succeq 0 \\ & & \begin{pmatrix} \alpha \\ P^{1/2} x + P^{-1/2} q \end{pmatrix} \geq_Q 0, \end{array} \tag{4.6}$$

where $\alpha \in \mathbb{R}^+$ is as before. Again, this problem is in the form of problem (4.2). Here, the difference between this form and SQV is in the second-order cone constraint since the SDP part is the same as SQV. The dimension of the second-order cone in SQV is $1+n^2$ and in SQQ is just $n+1$, which makes us expect less efficiency in practice when we work with SQV. The optimal value of SQV is the same as that of problem (1.2), whereas the optimal values of SQQ (4.6) and (4.5) are equal up to a constant. Indeed, the optimal value of (4.5) is equal $(\rho^*)^2 + \beta - q^T P^{-1} q$, where ρ^* is the optimal value of (4.6). It might notice that we did not talk about the constraint of $T(x)$ being Toeplitz. This is because the Toeplitz structure of $T(x)$ is embedded in the other constraints.

4.3 Formulation V (SQT)

The last formulation will take advantage of the Toeplitz structure of $T(x)$ explicitly. The vectorization operator tvec on Toeplitz matrices, introduced in Section 3 will be used. From Corollary 3.3, we have the following:

$$\|P_{\mathcal{P}}(F) - T(x)\|_F = \|\text{tvec}\,(P_{\mathcal{P}}(F) - T(x))\|_2,$$

so that we have the following problem:

$$\begin{array}{lll} \text{(SQT)} & \text{minimize} & \alpha \\ & \text{subject to} & \begin{pmatrix} \alpha & 0 \\ 0 & T(x) \end{pmatrix} \succeq 0 \qquad (4.7) \\ & & \begin{pmatrix} \alpha \\ \text{tvec}\,(P_{\mathcal{P}}(F) - T(x)) \end{pmatrix} \geq_Q 0. \qquad (4.8) \end{array}$$

The dimension of the second-order cone in this form is $n+1$, the same as that of SQQ. Furthermore, the optimal solution is the same as that of (1.2).

Table 4.1 shows the dimensions of the semidefinite part (SD part) and the second-order cone part (SOC part) for each formulation. For the formulations SDV, SDT and SDQ, the second-order cone part is not applicable, so the cell in the table corresponding to that is left blank.

In practice, we expect that the mixed formulations are more effecient than the SDP-only formulations, especially the SQQ and SQT which have second-order cone constraint of least dimension. Since, as we have seen, interior point methods for SOCP have better worst-case complexity than an

Formulation	SD part	SOC part
SDV	$n+1 \times (n^2+n+2)$	
SDT	$n+1 \times (2n+2)$	
SDQ	$n+1 \times (2n+2)$	
SQV	$n+1 \times (n+1)$	n^2+1
SQQ	$n+1 \times (n+1)$	$n+1$
SQT	$n+1 \times (n+1)$	$n+1$

Table 4.1: Problem dimensions

SDP method. However, SDT has a less SDP dimension with no illness such as that SDQ has, which makes SDT a better choice among other SDP. This is due to the economical vectorization operator tvec . Practical experiments show a competitive behaviour of SDT to SQQ and SQT (see Section 5).

5 Computational Results

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

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

For the other methods, the software SDPT3 ver. 3.0 [24] was used because of its numerical stability [10] and its ability to exploit sparsity very efficiently. The default starting iterates in SDPT3 were used throughout with the NT-direction. The choice of the NT-direction came after some preliminary numerical results. The other direction is HKM-direction which we found less accurate, although, faster than the NT-direction. However, the difference between the two in speed is not of significant importance.

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

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

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

All numerical experiments in this section were executed in Matlab 6.1 on a 2.0 GHz Pentium IV PC with 512 MB memory.

Size	Pro.		SDT		SDQ		SQT		SQQ		SQV	
n	It.	CPU	It.	CPU	It.	CPU	It.	CPU	It.	CPU	It.	CPU
50	5665	91.97	24	2.234	28	2.584	16	1.312	20	1.532	21	1.653
100	3337	290.97	34	17.896	33	17.454	17	8.332	23	10.866	24	11.887
150	8898	1921.2	38	69.16	37	66.576	20	33.98	23	38.024	23	39.076
200	26162	12504	71	329.12	66	306.15	21	96.98	26	115.17	26	117.57
250	21573	13521	70	711.67	83	845.46	20	195.94	27	259.92	25	245.81

Table 5.1: Performance comparison among the projection method and the path-following method with the formulations SDT, SDQ, SQT, SQQ and SQV.

The numerical experiments were carried out on two set of randomly generated square data matrices, each matrix is dense and its entries vary between -100 and 100 exclusive. First, we applied all approaches on small matrices F of dimension n ranging from 50 to 250, and typical results are summarized in Table 5.1. In all cases, we found the optimum to high accuracy, at least eight decimals. Table 5.1 compares the CPU time and the number of iterations of all six approaches. We notice that the consumed time gets larger more rapidly in the projection method with the size of the data matrix F. An obvious remark is that the projection method is the slowest. It is at least seven times slower than the slowest of the five formulations of the path-following method. However, the difference in time between the five formulations is not big enough to have a significant importance.

The projection method is expencive hence we exclude it from the second set of test problems, which we applied to the SDP and SOC formula for a randomly generated data matrices ranges from $n = 10, \ldots, 500$. The computations for the formulations SDT and SDQ are slow hence we stop the

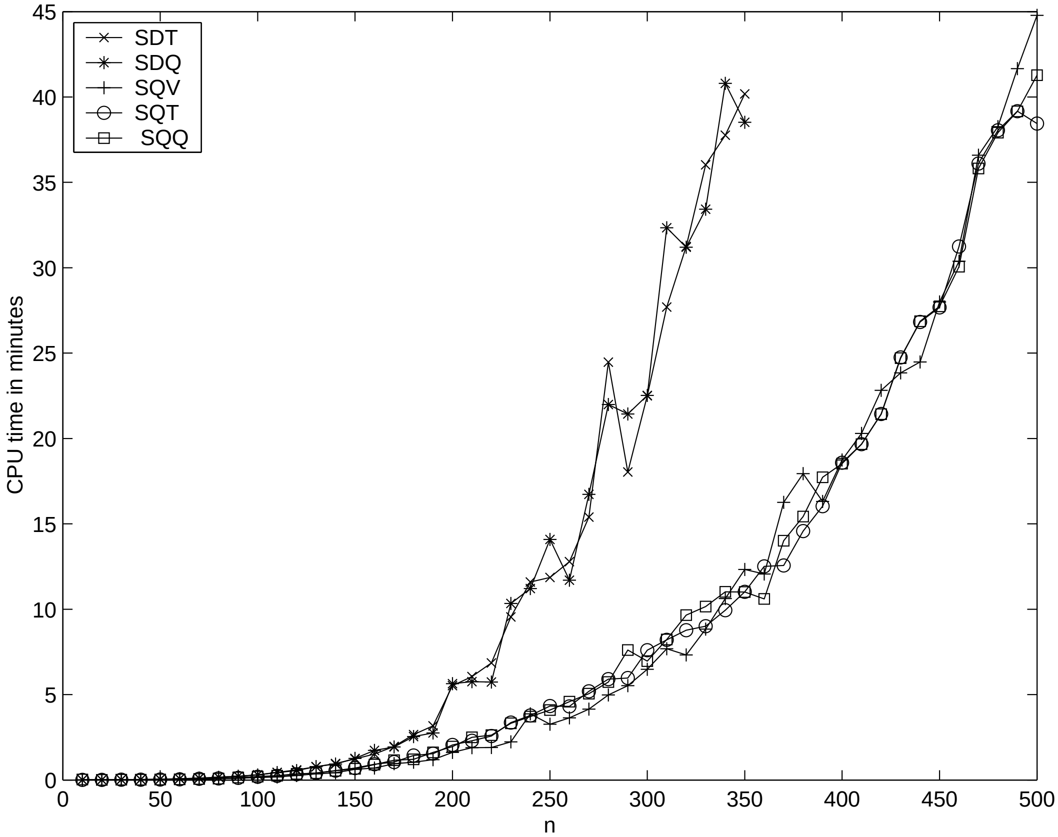

Figure 5.1: Comparing the CPU times against the size of the matrix F.

computations for them when n reaches 350. The results appear in Figure 5.1 which compare the CPU time in minutes against the size of the data matrix F. We can see the correlation between the CPU time and the size of the matrix. Figure 5.2 shows the number of iterations by SDPT3 against the size of the matrix F.

6 Conclusion:

We conclude this paper by addressing few remarks. The projection method, despite its accuracy, is very slow. Whereas, the path-following method with SDT, SQT and SQQ formulations is very fast, sometimes more than 120 times faster than the projection method (see Table 5.1 e.g. when $n = 200$), and

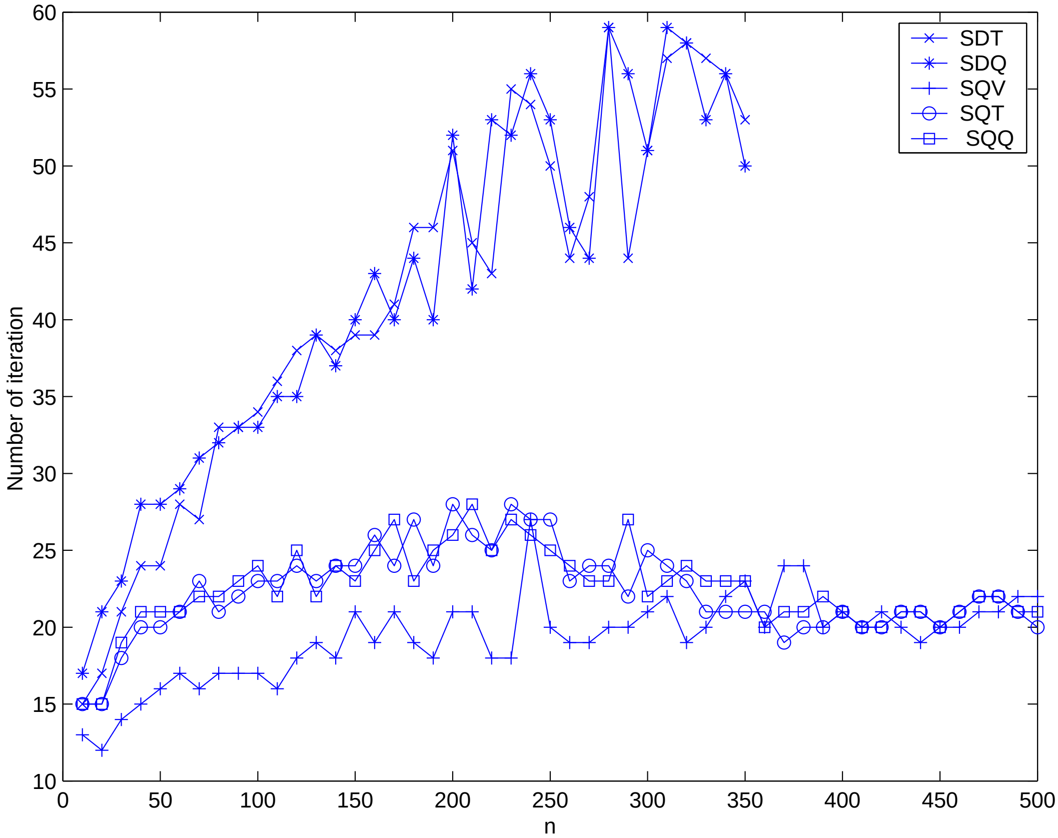

Figure 5.2: Comparing the number of iterations by SDPT3 against the size of the matrix F.

gives results of acceptable accuracy. The other is that we gain considerable advantage out of solving our problem as a mixed semidefinite and second-order cone problem (SQT, SQQ and SQV). This can be seen clearly by noticing the bad performance of the formulation SDT and SDQ, (see Figures 5.1 and 5.2) which solves the problem as a semidefinite program.

References

[1] S. Al-Homidan. Sqp algorithms for solving toeplitz matrix approximation problem. *Numerical Linear Algebra with Applications*, 9:619–627, 2002.

[2] S. Al-Homidan. Toeplitz matrix approximation. *Mathematical Sciences Research Journal*, 6:104–111, 2002.

[3] S. Al-Homidan and R. Fletcher. Hybrid methods for finding the nearest Euclidean distance matrix. In D. Du, L. Qi, and R. Womersley, editors, *Recent Advances in Nonsmooth optimization*, pages 1–7. World Scientific Publishing Co. Pte. Ltd., 1995.

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

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

[6] 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.

[7] G. Cybenko. Moment problems and low rank toeplitz approximations. *Circuits Systems Signal processing*, 1:345–365, 1982.

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

[9] R. Fletcher and S. Leyffer. User manual for filtersqp. Technical Report NA/181, Univerity of Dundee Numerical Analysis Report, Dundee, Scotland, 1999.

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

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

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

[13] S. Kaczmarz. Approximate solution of systems of linear equations. *Internat. J. Control*, 57(6):1269–1271, 1993. Translated from the German.

[14] T. Kailath. A view of three decades of linear filtering theory. *IEEE Transactions on Information Theory*, IT-20:145–181, 1974.

[15] S. Y. Kung. 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, 1981.

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

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

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

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

[20] R. Skelton. The jury test and covariance control. In , editor, *Proceedings of the Symposium on Fundamentals of Discrete Time Systems*. Chicago, 1992.

[21] J. Sturm. Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. *Optimization Methods and Software*, 17(6):1105–1154, 2002.

[22] Y. Suffridge and T. Hayden. Approximation by a hermitian positive semi–definite toeplitz matrix. *SIAM J. on Matrix Analysis and Applications*, 14:721–734, 1993.

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

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

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

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

[27] A. Willsky. *Digital Signal Processing and Control and Estimation Theory.* MIT Press: Cambridge: Mass,, 1979.

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