Applied Mathematical Sciences, Vol. 2, 2008, no. 38, 1855 - 1872

Recursive Solutions of the Matrix Equations $X + A^T X^{-1} A = Q$ and $X - A^T X^{-1} A = Q$

Nicholas Assimakis

Department of Electronics, Technological Educational Institute of Lamia
35100 Lamia, Greece (assimakis@teilam.gr)

Francesca Sanida

Department of Electronics, Technological Educational Institute of Lamia
35100 Lamia, Greece (fsanida@teilam.gr)

Maria Adam

Department of Informatics with Applications to Biomedicine
University of Central Greece, 35100 Lamia, Greece
(maria@math.ntua.gr)

Abstract

Two classes of recursive algorithms for computing the extreme solutions of the matrix equations $X+A^T X^{-1} A = Q$ and $X-A^T X^{-1} A = Q$ are presented. The Per Step Algorithms are based on the fixed point iteration method and its variations; the proposed Per Step Algorithm is an inversion free variant of the fixed point iteration method. The Doubling Algorithms are based on the cyclic reduction method and the Riccati equation solution method; the proposed Doubling Algorithm uses recursive solutions of the corresponding Riccati equations to solve any of the above matrix equations. Simulation results are given to illustrate the efficiency of the proposed algorithms.

Mathematics Subject Classification: 93E25, 15A24, 93E10
Keywords: Computational methods, Matrix equations, Riccati equation

1 Introduction

The central issue of this paper is to propose recursive solutions of the matrix equations:

$$X + A^T X^{-1} A = Q \tag{1}$$

and

$$X - A^T X^{-1} A = Q \tag{2}$$

where Q is a $n \times n$ Hermitian positive definite matrix, A is a $n \times n$ matrix and A^T denotes the transpose of A.

These equations arise in many applications in various research areas including control theory, ladder networks, dynamic programming, stochastic filtering and statistics: see [2], [11] for references concerning equation (1) and [6] for references concerning equation (2). These equations have been studied recently by many authors [2]-[8], [10]-[12]: the theoretical properties such as necessary and sufficient conditions for the existence of a positive definite solution have been investigated and numerical methods for solving these equations have been proposed; the available methods are recursive algorithms based mainly on the fixed point iteration and on applications of the cyclic reduction method.

Concerning equation (1), a solution X of (1) is a $n \times n$ Hermitian positive definite matrix, as stated in [11]. It is well known [5], [10] that if (1) has a positive definite solution X, then there exist minimal and maximal solutions X_+ and X_-, respectively, such that $0 < X_- \leq X \leq X_+$ for any positive definite solution X. Here, if X and Y are Hermitian matrices, then $X \leq Y$ ($X < Y$) means that $Y - X$ is a nonnegative definite (positive definite) matrix.

Concerning equation (2), it is well known [6], [10] that there always exists a unique positive definite solution, which is the maximal solution X_+ of (2), and if A is nonsingular, then there exists a unique negative definite solution, which is the minimal solution X_- of (2).

The minimal and maximal solutions X_+ and X_- are referred as the extreme solutions of (1) and (2).

Furthermore, it is well known [10] that the minimal solution of $X + A^T X^{-1} A = Q$ and the maximal solution of the following equation

$$Y + A Y^{-1} A^T = Q \tag{3}$$

satisfy the relation:

$$X_- = Q - Y_+ \tag{4}$$

Thus, it becomes obvious that the minimal solution of (1) can be derived through the maximal solution of (1): the minimal solution of (1) can be computed using (4) via the maximal solution of the equation (3), which is of type (1).

It is also well known [10] that if A is nonsingular, then the minimal solution of $X - A^T X^{-1} A = Q$ and the maximal solution of the following equation,

$$Y - A Y^{-1} A^T = Q \tag{5}$$

satisfy the relation:

$$X_- = Q - Y_+ \tag{6}$$

Thus, it becomes obvious that the minimal solution of (2) can be derived through the maximal solution of (2): the minimal solution of (2) can be computed using (6) via the maximal solution of the equation (5), which is of type (2).

Moreover, the relation between the solutions of (1) and (2) is described in [3]: The solution X of $X - A^T X^{-1} A = Q$ can be computed as:

$$X = Y - AQ^{-1}A^T \tag{7}$$

where Y is the solution of the following equation, which is of type (1):

$$Y + B^T Y^{-1} B = R \tag{8}$$

with

$$B = AQ^{-1}A \tag{9}$$

$$R = Q + A^T Q^{-1} A + AQ^{-1}A^T \tag{10}$$

Thus, it becomes obvious that the solution of (2) can be derived through the solution of (1): the extreme solutions of (2) can be computed using (7) via the extreme solutions of (8), which is of type (1).

Hence, it becomes clear that the extreme solutions of (1) and (2) can be derived through the maximal solution of (1).

In this paper, recursive algorithms for computing the extreme solutions of the matrix equations (1) and (2) are proposed. The efficiency of the proposed algorithms is verified trough simulation experiments.

2 Recursive algorithms

2.1 Per Step Algorithms

2.1.1 Recursive algorithms for $X + A^T X^{-1} A = Q$

Algorithm 2.1 Fixed Point Iteration Method
The basic fixed point iteration is described in the following (algorithm 3.2 in [10]):

$$X_{n+1} = Q - A^T X_n^{-1} A$$
$$X_0 = Q$$
$$X_n \to X_+$$

Note that all per step algorithms are applicable for $n = 0, 1, \ldots$ and that the convergence is achieved, when $\|X_{n+1} - X_n\| < \epsilon$, where ϵ is a small positive number and $\|M\|$ denotes the norm of the matrix M (the largest singular value of M).

Algorithm 2.2 Inversion Free Fixed Point Iteration Method
In order to avoid possible numerical instability problems due to matrix inversions, the inversion free variant has been proposed (algorithm 3.1 in [8]):

$$\begin{aligned}
&X_{n+1} = Q - A^T Y_n A \\
&Y_{n+1} = Y_n \left[2I - X_n Y_n\right] \\
&X_0 = Q \\
&Y_0 = \frac{1}{\|Q\|_\infty} I \\
&X_n \to X_+
\end{aligned}$$

where $\|M\|_\infty$ denotes the maximum row sum for the matrix M.
Note that this algorithm may have convergence problems; for example, in the case $Q = I$, the algorithm computes $X_1 = X_2 = I - A^T A$.

Algorithm 2.3 Modified Inversion Free Fixed Point Iteration Method
In order to improve the convergence properties, the modified inversion free variant has been proposed (algorithm 3.4 in [8] and algorithm 3.3 in [2]):

$$\begin{aligned}
&Y_{n+1} = Y_n \left[2I - X_n Y_n\right] \\
&X_{n+1} = Q - A^T Y_{n+1} A \\
&X_0 = Q \\
&Y_0 = \frac{1}{\|Q\|_\infty} I \\
&X_n \to X_+
\end{aligned}$$

Algorithm 2.4 Proposed Inversion Free Fixed Point Iteration Method
The proposed Per Step Algorithm is an inversion free variant of the fixed point iteration method; the basic idea is to replace the computation of Y_{n+1} in Algorithm 2.2 by the Schulz iteration for X_n^{-1} as described in [11]:

$$\begin{aligned}
&X_{n+1} = Q - A^T Y_n A \\
&\quad Z_{i+1} = Z_i \left[2I - X_n Z_i\right] \\
&\quad Z_0 = \frac{1}{\|X_n\|_\infty} I \\
&\quad Z_i \to Y_n \\
&X_0 = Q \\
&X_n \to X_+
\end{aligned}$$

2.1.2 Recursive algorithms for $X - A^T X^{-1} A = Q$

Algorithm 2.5 Fixed Point Iteration Method
The basic fixed point iteration is described in the following (algorithm 4.2 in [10]):

$$
\begin{aligned}
&X_{n+1} = Q + A^T X_n^{-1} A \\
&X_0 = Q \\
&X_n \to X_+
\end{aligned}
$$

Algorithm 2.6 Inversion Free Fixed Point Iteration Method
The idea to replace the computation of $X_{n+1} = Q - A^T Y_n A$ in Algorithm 2.2 by $X_{n+1} = Q + A^T Y_n A$ does not work, because the quantity X_n may tend to infinite. So, another approach is proposed in the following.
The relation between the solutions of $X + A^T X^{-1} A = Q$ and $X - A^T X^{-1} A = Q$ is described in [3] and presented in (7)-(10). Then it becomes clear that we are able to compute the extreme solutions of (2) via the extreme solutions of (8), which is of type (1). Thus, in order to avoid possible numerical instability problems due to matrix inversions, the following inversion free variant is proposed:

$$
\begin{aligned}
&B = AQ^{-1}A \\
&R = Q + A^T Q^{-1} A + AQ^{-1} A^T
\end{aligned}
$$

solve $Y + B^T Y^{-1} B = R$ via Inversion Free Fixed Point Iteration Method (Algorithm 2.2)

$$X_+ = Y_+ - AQ^{-1}A^T$$

Algorithm 2.7 Modified Inversion Free Fixed Point Iteration Method
Using the ideas of Algorithms 2.3 and 2.6, the modified inversion free variant is proposed:

$$
\begin{aligned}
&B = AQ^{-1}A \\
&R = Q + A^T Q^{-1} A + AQ^{-1} A^T
\end{aligned}
$$

solve $Y + B^T Y^{-1} B = R$ via Modified Inversion Free Fixed Point Iteration Method (Algorithm 2.3)

$$X_+ = Y_+ - AQ^{-1}A^T$$

Algorithm 2.8 Proposed Inversion Free Fixed Point Iteration Method
Using the idea of Algorithm 2.4, the proposed Per Step Algorithm is proposed as an inversion free variant of the fixed point iteration method:

$$\begin{aligned}
&X_{n+1} = Q + A^T Y_n A \\
&\quad Z_{i+1} = Z_i \left[2I - X_n Z_i\right] \\
&\quad Z_0 = \frac{1}{\|X_n\|_\infty} I \\
&\quad Z_i \to Y_n \\
&X_0 = Q \\
&X_n \to X_+
\end{aligned}$$

All Per Step Algorithms presented in this section are summarized in Table 1.

Table 1. Per Step Algorithms

	$X + A^T X^{-1} A = Q$	$X - A^T X^{-1} A = Q$
Fixed Point Iteration Method	$X_{n+1} = Q - A^T X_n^{-1} A$ $X_0 = Q$ $X_n \to X_+$	$X_{n+1} = Q + A^T X_n^{-1} A$ $X_0 = Q$ $X_n \to X_+$
Inversion Free Fixed Point Iteration Method	$X_{n+1} = Q - A^T Y_n A$ $Y_{n+1} = Y_n \left[2I - X_n Y_n\right]$ $X_0 = Q$ $Y_0 = \frac{1}{\|Q\|_\infty} I$ $X_n \to X_+$	$B = AQ^{-1}A$ $R = Q + A^T Q^{-1} A + AQ^{-1} A^T$ solve $Y + B^T Y^{-1} B = R$ via Inversion Free Fixed Point Iteration Method $X_+ = Y_+ - AQ^{-1} A^T$
Modified Inversion Free Fixed Point Iteration Method	$Y_{n+1} = Y_n \left[2I - X_n Y_n\right]$ $X_{n+1} = Q - A^T Y_{n+1} A$ $X_0 = Q$ $Y_0 = \frac{1}{\|Q\|_\infty} I$ $X_n \to X_+$	$B = AQ^{-1}A$ $R = Q + A^T Q^{-1} A + AQ^{-1} A^T$ solve $Y + B^T Y^{-1} B = R$ via Modified Inversion Free Fixed Point Iteration Method $X_+ = Y_+ - AQ^{-1} A^T$
Proposed Inversion Free Fixed Point Iteration Method	$X_{n+1} = Q - A^T Y_n A$ $Z_{i+1} = Z_i \left[2I - X_n Z_i\right]$ $Z_0 = \frac{1}{\|X_n\|_\infty} I$ $Z_i \to Y_n$ $X_0 = Q$ $X_n \to X_+$	$X_{n+1} = Q + A^T Y_n A$ $Z_{i+1} = Z_i \left[2I - X_n Z_i\right]$ $Z_0 = \frac{1}{\|X_n\|_\infty} I$ $Z_i \to Y_n$ $X_0 = Q$ $X_n \to X_+$

2.2 Doubling Algorithms

2.2.1 Recursive algorithms for $X + A^T X^{-1} A = Q$

Algorithm 2.9 Cyclic Reduction Method
The cyclic reduction method is presented in the following (algorithm 3.1 in [10]):

$$\begin{aligned}
&A_{n+1} = A_n Q_n^{-1} A_n \\
&Q_{n+1} = Q_n - A_n Q_n^{-1} A_n^T - A_n^T Q_n^{-1} A_n \\
&X_{n+1} = X_n - A_n^T Q_n^{-1} A_n \\
&Y_{n+1} = Y_n - A_n Q_n^{-1} A_n^T \\
&A_1 = A Q^{-1} A \\
&Q_1 = Q - A Q^{-1} A^T - A^T Q^{-1} A \\
&X_1 = Q - A^T Q^{-1} A \\
&Y_1 = Q - A Q^{-1} A^T \\
&X_n \to X_+
\end{aligned}$$

Note that all doubling algorithms are applicable for $n = 1, 2, \ldots$ and that the convergence is achieved, when $\|X_{n+1} - X_n\| < \epsilon$, where ϵ is a small positive number and $\|M\|$ denotes the norm of the matrix M (the largest singular value of M).

At this point we remark that we are able to use the initial condition $A_1 = -AQ^{-1}A$ instead of the initial condition $A_1 = AQ^{-1}A$. In fact, it can be trivially verified that the use of both these initial conditions leads to the calculation of the same quantities involved in the algorithm for $n = 2$, (A_2, Q_2, X_2 and Y_2) and hence to the same sequences for $n = 2, 3, \ldots$ of all the quantities involved in the algorithm (A_n, Q_n, X_n and Y_n). In the following, we use this algorithm with the initial condition $A_1 = -AQ^{-1}A$.

In addition, note that Algorithm 2.9 provides the minimal solution of (1) due to $\lim_{n\to\infty} Y_n = Q - X_-$ as stated in [10].

Algorithm 2.10 Riccati Equation Solution Method
Working as in [5] and [6] we are able to derive a Riccati equation, which is equivalent to the matrix equation (1). The proposed method takes advantage of this relation and provides a recursive solution of (1) via the solution of the related Riccati equation. In fact, the matrix equation (1) can be written as

$$X = Q - A^T X^{-1} A,$$

then, we have:

$$X = Q - A^T \left[Q - A^T X^{-1} A\right]^{-1} A = Q - A^T A^{-1} \left[A^{-T} Q A^{-1} - X^{-1}\right]^{-1} A^{-T} A$$

$$
\begin{aligned}
&= Q + A^T A^{-1}\left[-A^{-T} Q A^{-1} + X^{-1}\right]^{-1} A^{-T} A \\
&= Q + \left(A^T A^{-1}\right)\left[X^{-1} + \left(-A^{-T} Q A^{-1}\right)\right]^{-1}\left(A^T A^{-1}\right)^T \qquad (11)
\end{aligned}
$$

The equivalent related Riccati equation is:

$$
\begin{aligned}
P &= q + fPf^T - fPh^T(hPh^T + r)^{-1}hPf^T \\
&= q + f(P^{-1} + h^T r^{-1} h)^{-1} f^T \qquad (12)
\end{aligned}
$$

with

$$
\begin{aligned}
f &= A^T A^{-1} \qquad (13) \\
q &= Q \qquad (14) \\
b &= h^T r^{-1} h = -A^{-T} Q A^{-1} \qquad (15)
\end{aligned}
$$

It becomes obvious that the matrix equation (1) is equivalent to the related Riccati equation (12) and that the two equations have equivalent solutions. Thus, the unique maximal solution of (1) coincides with the unique positive definite solution of the related Riccati equation:

$$
X_+ = P \qquad (16)
$$

Thus, it is clear that we will be able to solve (1), if we know the solution of the related Riccati equation. The solution of the related Riccati equation can be derived using various available recursive solutions in [1] and [9]. The proposed method uses the efficient and fast Doubling Algorithm DAKF3 in [9], which is summarized as follows:

$$
\begin{aligned}
&\alpha_{n+1} = \alpha_n \left(\beta_n + \pi_n\right)^{-1} \alpha_n \\
&\beta_{n+1} = \beta_n - \alpha_n^T \left(\beta_n + \pi_n\right)^{-1} \alpha_n \\
&\pi_{n+1} = \pi_n - \alpha_n \left(\beta_n + \pi_n\right)^{-1} \alpha_n^T \\
&\alpha_1 = q^{-1} f \\
&\beta_1 = f^T q^{-1} f + h^T r^{-1} h \\
&\pi_1 = q^{-1} \\
&\pi_n \to \Pi \\
&P = \Pi^{-1}
\end{aligned}
$$

Substituting (13)-(15) in the initial conditions of this algorithm, the following initial conditions are derived:

$$
\begin{aligned}
\alpha_1 &= Q^{-1} A^T A^{-1} \qquad (17) \\
\beta_1 &= A^{-T} A Q^{-1} A^T A^{-1} - A^{-T} Q A^{-1} \qquad (18) \\
\pi_1 &= Q^{-1} \qquad (19)
\end{aligned}
$$

provided that A is nonsingular. Finally, using the above initial conditions and using (16), the proposed Riccati Equation Solution Method is derived:

$$\begin{aligned}
&\alpha_{n+1} = \alpha_n(\beta_n + \pi_n)^{-1}\alpha_n \\
&\beta_{n+1} = \beta_n - \alpha_n^T(\beta_n + \pi_n)^{-1}\alpha_n \\
&\pi_{n+1} = \pi_n - \alpha_n(\beta_n + \pi_n)^{-1}\alpha_n^T \\
&\alpha_1 = Q^{-1}A^T A^{-1} \\
&\beta_1 = A^{-T}AQ^{-1}A^T A^{-1} - A^{-T}QA^{-1} \\
&\pi_1 = Q^{-1} \\
&\pi_n \to \Pi \\
&P = \Pi^{-1} \\
&X_+ = P
\end{aligned}$$

The Cyclic Reduction Method (Algorithm 2.9) and the Riccati Equation Solution Method (Algorithm 2.10) are equivalent to each other. The proof is given in the Appendix.

In addition, note that Algorithm 2.10 provides the minimal solution X_- of (1) due to $\lim_{n\to\infty} \beta_n = B$, since $X_- = Q + A^T BA$, as it is shown in the Appendix.

Finally, note that the Riccati Equation Solution Method (Algorithm 2.10) is slightly faster (by one matrix addition per iteration) than the Cyclic Reduction Method (Algorithm 2.9).

2.2.2 Recursive algorithms for $X - A^T X^{-1} A = Q$

Algorithm 2.11 Cyclic Reduction Method
The cyclic reduction method is presented in the following (algorithm 4.1 in [10]):

$$\begin{aligned}
&A_{n+1} = A_n Q_n^{-1} A_n \\
&Q_{n+1} = Q_n - A_n Q_n^{-1} A_n^T - A_n^T Q_n^{-1} A_n \\
&X_{n+1} = X_n - A_n^T Q_n^{-1} A_n \\
&Y_{n+1} = Y_n - A_n Q_n^{-1} A_n^T \\
&A_1 = AQ^{-1}A \\
&Q_1 = Q + AQ^{-1}A^T + A^T Q^{-1} A \\
&X_1 = Q + A^T Q^{-1} A \\
&Y_1 = Q + AQ^{-1}A^T \\
&X_n \to X_+
\end{aligned}$$

In addition, note that Algorithm 2.9 provides the minimal solution X_- of (1) due to $\lim_{n\to\infty} Y_n = Q - X_-$ as stated in [10].

Algorithm 2.12 Riccati Equation Solution Method
Working as in [6] we are able to derive a Riccati equation, which is equivalent to the matrix equation (2). The proposed method takes advantage of this relation and provides a recursive solution of (2) via the solution of the related Riccati equation. In fact, the matrix equation (2) can be written as

$$X = Q + A^T X^{-1} A,$$

then, we have:

$$\begin{aligned} X &= Q + A^T \left[Q + A^T X^{-1} A\right]^{-1} A \\ &= Q + A^T A^{-1} \left[A^{-T} Q A^{-1} + X^{-1}\right]^{-1} A^{-T} A \\ &= Q + \left(A^T A^{-1}\right) \left[X^{-1} + A^{-T} Q A^{-1}\right]^{-1} \left(A^T A^{-1}\right)^T \end{aligned} \tag{20}$$

The equivalent related Riccati equation is:

$$\begin{aligned} P &= q + fPf^T - fPh^T(hPh^T + r)^{-1} hPf^T \\ &= q + f(P^{-1} + h^T r^{-1} h)^{-1} f^T \end{aligned} \tag{21}$$

with

$$f = A^T A^{-1} \tag{22}$$

$$q = Q \tag{23}$$

$$b = h^T r^{-1} h = A^{-T} Q A^{-1} \tag{24}$$

It becomes obvious that the matrix equation (2) is equivalent to the related Riccati equation (21) and that the two equations have equivalent solutions. Thus, the unique maximal solution of (2) coincides with the unique positive definite solution of the related Riccati equation:

$$X_+ = P \tag{25}$$

The solution of the related Riccati equation can be derived using various available recursive solutions [1], [9]. The proposed method uses the efficient and fast Doubling Algorithm DAKF3 in [9], as presented before.

Substituting (22)-(24) in the initial conditions of this algorithm, the following initial conditions are derived:

$$\alpha_1 = Q^{-1} A^T A^{-1} \tag{26}$$

$$\beta_1 = A^{-T} A Q^{-1} A^T A^{-1} + A^{-T} Q A^{-1} \tag{27}$$

$$\pi_1 = Q^{-1} \tag{28}$$

provided that A is nonsingular. Finally, using the above initial conditions and using (25), the proposed Riccati Equation Solution Method is derived:

$$
\begin{aligned}
&\alpha_{n+1} = \alpha_n(\beta_n + \pi_n)^{-1}\alpha_n \\
&\beta_{n+1} = \beta_n - \alpha_n^T(\beta_n + \pi_n)^{-1}\alpha_n \\
&\pi_{n+1} = \pi_n - \alpha_n(\beta_n + \pi_n)^{-1}\alpha_n^T \\
&\alpha_1 = Q^{-1}A^T A^{-1} \\
&\beta_1 = A^{-T}AQ^{-1}A^T A^{-1} + A^{-T}QA^{-1} \\
&\pi_1 = Q^{-1} \\
&\pi_n \to \Pi \\
&P = \Pi^{-1} \\
&X_+ = P
\end{aligned}
$$

The Cyclic Reduction Method (Algorithm 2.11) and the Riccati Equation Solution Method (Algorithm 2.12) are equivalent to each other. The proof is in the Appendix.

In addition, note that Algorithm 2.12 provides the minimal solution X_- of (2) due to $\lim_{n\to\infty} \beta_n = B$, since $X_- = Q - A^T BA$, as it is shown in the Appendix.

Finally, note that the Riccati Equation Solution Method (Algorithm 2.12) is slightly faster (by one matrix addition per iteration) than the Cyclic Reduction Method (Algorithm 2.11).

All Doubling Algorithms presented in this section are summarized in Table 2 (Cyclic Reduction Method) and Table 3 (Riccati Equation Solution Method).

Table 2. Cyclic Reduction Method

$X + A^T X^{-1} A = Q$	$X - A^T X^{-1} A = Q$
$A_{n+1} = A_n Q_n^{-1} A_n$ $Q_{n+1} = Q_n - A_n Q_n^{-1} A_n^T - A_n^T Q_n^{-1} A_n$ $X_{n+1} = X_n - A_n^T Q_n^{-1} A_n$ $Y_{n+1} = Y_n - A_n Q_n^{-1} A_n^T$ $A_1 = -AQ^{-1}A$ $Q_1 = Q - AQ^{-1}A^T - A^T Q^{-1} A$ $X_1 = Q - A^T Q^{-1} A$ $Y_1 = Q - AQ^{-1}A^T$ $X_n \to X_+$	$A_{n+1} = A_n Q_n^{-1} A_n$ $Q_{n+1} = Q_n - A_n Q_n^{-1} A_n^T - A_n^T Q_n^{-1} A_n$ $X_{n+1} = X_n - A_n^T Q_n^{-1} A_n$ $Y_{n+1} = Y_n - A_n Q_n^{-1} A_n^T$ $A_1 = AQ^{-1}A$ $Q_1 = Q + AQ^{-1}A^T + A^T Q^{-1} A$ $X_1 = Q + A^T Q^{-1} A$ $Y_1 = Q + AQ^{-1}A^T$ $X_n \to X_+$

Table 3. Riccati Equation Solution Method

$X + A^T X^{-1} A = Q$	$X - A^T X^{-1} A = Q$
$\alpha_{n+1} = \alpha_n(\beta_n + \pi_n)^{-1}\alpha_n$ $\beta_{n+1} = \beta_n - \alpha_n^T(\beta_n + \pi_n)^{-1}\alpha_n$ $\pi_{n+1} = \pi_n - \alpha_n(\beta_n + \pi_n)^{-1}\alpha_n^T$ $\alpha_1 = Q^{-1}A^T A^{-1}$ $\beta_1 = A^{-T}AQ^{-1}A^T A^{-1} - A^{-T}QA^{-1}$ $\pi_1 = Q^{-1}$ $\pi_n \to \Pi$ $P = \Pi^{-1}$ $X_+ = P$	$\alpha_{n+1} = \alpha_n(\beta_n + \pi_n)^{-1}\alpha_n$ $\beta_{n+1} = \beta_n - \alpha_n^T(\beta_n + \pi_n)^{-1}\alpha_n$ $\pi_{n+1} = \pi_n - \alpha_n(\beta_n + \pi_n)^{-1}\alpha_n^T$ $\alpha_1 = Q^{-1}A^T A^{-1}$ $\beta_1 = A^{-T}AQ^{-1}A^T A^{-1} + A^{-T}QA^{-1}$ $\pi_1 = Q^{-1}$ $\pi_n \to \Pi$ $P = \Pi^{-1}$ $X_+ = P$

3 Simulation results

Simulation results are given to illustrate the efficiency of the proposed methods. The proposed methods compute accurate solutions as verified trough the following simulation examples.

Example 3.1 This example concerns the equation $X + A^T X^{-1} A = Q$ and is taken from [8] (example 7.1). Consider the equation (1) with

$$A = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}, \quad Q = \begin{bmatrix} 6.0 & 5.0 \\ 5.0 & 8.6 \end{bmatrix}.$$

Algorithms 2.1, 2.2, 2.3, 2.4, 2.9, 2.10 have been applied. All algorithms compute the same maximal solution of (1):

$$X_+ = \begin{bmatrix} 3.8832 & 2.4009 \\ 2.4009 & 4.3460 \end{bmatrix}.$$

Example 3.2 This example concerns the equation $X + A^T X^{-1} A = Q$ and is taken from [8] (example 7.2). Consider the equation (1) with

$$A = \begin{bmatrix} 0.20 & 0.20 & 0.10 \\ 0.20 & 0.15 & 0.15 \\ 0.10 & 0.15 & 0.25 \end{bmatrix}, \quad Q = I.$$

Algorithms 2.1, 2.2, 2.3, 2.4, 2.9, 2.10 have been applied. All algorithms compute the same maximal solution of (1):

$$X_+ = \begin{bmatrix} 0.8266 & -0.1684 & -0.1581 \\ -0.1684 & 0.8317 & -0.1632 \\ -0.1581 & -0.1632 & 0.8215 \end{bmatrix}$$

Example 3.3 This example concerns the equation $X + A^T X^{-1} A = Q$ and is taken from [8] (example 7.3) and from [10] (example 5.1). Consider the equation (1) with

$$A = \begin{bmatrix} 0.37 & 0.13 & 0.12 \\ -0.30 & 0.34 & 0.12 \\ 0.11 & -0.17 & 0.29 \end{bmatrix}, \quad Q = \begin{bmatrix} 1.20 & -0.30 & 0.10 \\ -0.30 & 2.10 & 0.20 \\ 0.10 & 0.20 & 0.65 \end{bmatrix}$$

Algorithms 2.1, 2.2, 2.3, 2.4, 2.9, 2.10 have been applied. All algorithms compute the same maximal solution of (1):

$$X_+ = \begin{bmatrix} 0.9463 & -0.1987 & -0.0596 \\ -0.1987 & 1.8674 & 0.3252 \\ -0.0596 & 0.3252 & 0.4158 \end{bmatrix}$$

Example 3.4 This example concerns the equation $X - A^T X^{-1} A = Q$ and is taken from [8] (example 7.4) and from [10] (example 5.3). Consider the equation (2) with

$$A = \begin{bmatrix} 50 & 20 \\ 10 & 60 \end{bmatrix}, \quad Q = \begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix}.$$

Algorithms 2.5, 2.6, 2.7, 2.8, 2.11, 2.12 have been applied. All algorithms compute the same maximal solution of (2):

$$X_+ = \begin{bmatrix} 51.7994 & 16.0999 \\ 16.0999 & 62.2516 \end{bmatrix}$$

4 Conclusions

Recursive algorithms for computing the extreme solutions of the matrix equations $X + A^T X^{-1} A = Q$ and $X - A^T X^{-1} A = Q$ are presented. The Per Step Algorithms are based on the fixed point iteration method and its variations. The Doubling Algorithms are based on the cyclic reduction method and the Riccati equation solution method, which uses the recursive solutions of the corresponding discrete time Riccati equations. Simulation results are given to illustrate the efficiency of the proposed algorithms.

It is clear [3]-[6] that all available recursive algorithms for solving the matrix equations $X + A^T X^{-1} A = Q$ and $X - A^T X^{-1} A = Q$ can be used to solve special cases of corresponding Riccati equations. In this paper recursive solutions of the corresponding Riccati equations are used to solve any of the above matrix equations.

5 Appendix

5.1 Equivalence of the Cyclic Reduction Method and the Riccati Equation Solution Method

5.1.1 Equivalence of Algorithms 2.9 and 2.10 for $X + A^T X^{-1} A = Q$

In order to prove the equivalence of the Cyclic Reduction Method (Algorithm 2.9) and the Riccati Equation Solution Method (Algorithm 2.10), it is essential to express the relations between the quantities involved in the two algorithms. The recursive part of Algorithm 2.9 is:

$$A_{n+1} = A_n Q_n^{-1} A_n \tag{A.1}$$
$$Q_{n+1} = Q_n - A_n Q_n^{-1} A_n^T - A_n^T Q_n^{-1} A_n \tag{A.2}$$
$$X_{n+1} = X_n - A_n^T Q_n^{-1} A_n \tag{A.3}$$
$$Y_{n+1} = Y_n - A_n Q_n^{-1} A_n^T \tag{A.4}$$

for $n = 1, 2, \dots$, with initial conditions

$$A_1 = -AQ^{-1}A \tag{A.5}$$
$$Q_1 = Q - AQ^{-1}A^T - A^T Q^{-1} A \tag{A.6}$$
$$X_1 = Q - A^T Q^{-1} A \tag{A.7}$$
$$Y_1 = Q - AQ^{-1}A^T \tag{A.8}$$

and

$$X_n \to X_+ \tag{A.9}$$

It is clear that the matrices Q_n, X_n and Y_n are symmetric due to the fact that Q is a symmetric matrix.

The recursive part of Algorithm 2.10 is:

$$\alpha_{n+1} = \alpha_n (\beta_n + \pi_n)^{-1} \alpha_n \tag{A.10}$$
$$\beta_{n+1} = \beta_n - \alpha_n^T (\beta_n + \pi_n)^{-1} \alpha_n \tag{A.11}$$
$$\pi_{n+1} = \pi_n - \alpha_n (\beta_n + \pi_n)^{-1} \alpha_n^T \tag{A.12}$$

for $n = 1, 2, \dots$, with initial conditions

$$\alpha_1 = Q^{-1} A^T A^{-1} \tag{A.13}$$
$$\beta_1 = A^{-T} A Q^{-1} A^T A^{-1} - A^{-T} Q A^{-1} \tag{A.14}$$
$$\pi_1 = Q^{-1} \tag{A.15}$$

and

$$\pi_n \to \Pi \tag{A.16}$$

where

$$P = \Pi^{-1} \tag{A.17}$$

$$X_+ = P \tag{A.18}$$

It is clear that the matrices β_n and π_n are symmetric due to the fact that Q is a symmetric matrix.

We are going to prove have by induction that the following relations between the quantities involved in the two algorithms hold:

$$Q_n = -A^T(\beta_n + \pi_n)A \tag{A.19}$$

$$A_n = -A^T \alpha_n^T A \tag{A.20}$$

$$X_n = Q - A^T \pi_n A \tag{A.21}$$

$$Y_n = -A^T \beta_n A \tag{A.22}$$

for $n = 1, 2, \ldots$.

Concerning the initial conditions of the two algorithms, it can be trivially verified that:

$Q_1 = -A^T(\beta_1 + \pi_1)A,$ from (A.6), (A.14), (A15)
$A_1 = -A^T \alpha_1^T A,$ from (A.5), (A.13)
$X_1 = Q - A^T \pi_1 A,$ from (A.7), (A.15)
$Y_1 = -A^T \beta_1 A,$ from (A.8), (A.14)

Concerning the recursive parts of the two algorithms, after some algebra we have:

$Q_{n+1} = -A^T(\beta_{n+1} + \pi_{n+1})A,$ from (A.2), (A.11), (A.12), (A.19), (A.20)
$A_{n+1} = -A^T \alpha_{n+1}^T A,$ from (A.1), (A.10), (A.19), (A.20)
$X_{n+1} = Q - A^T \pi_{n+1} A,$ from (A.3), (A.12), (A.19), (A.20), (A.21)
$Y_{n+1} = -A^T \beta_{n+1} A,$ from (A.4), (A.11), (A.19), (A.20), (A.22)

Finally, from (A.9), (A.16), (A.17), (A.18) and (A.21), we conclude that the limiting solutions X_+ of the two algorithms satisfy $X_+ + A^T X_+^{-1} A = Q$. It is obvious that both algorithms compute the maximal solution X_+ of (1). This completes the proof that the Cyclic Reduction Method (Algorithm 2.9) and the Riccati Equation Solution Method (Algorithm 2.10) are equivalent to each other.

Concerning the minimal solution X_- of (1), it is known [10] that Algorithm 2.9 provides the minimal solution X_- of (1) due to $\lim_{n\to\infty} Y_n = Q - X_-$ and hence Algorithm 2.10 provides the minimal solution X_- of (1) due to $\lim_{n\to\infty} \beta_n = B$, since it is easy to derive $X_- = Q + A^T B A$ using (A.22).

5.1.2 Equivalence of Algorithms 2.11 and 2.12 for $X - A^T X^{-1} A = Q$

In order to prove the equivalence of the Cyclic Reduction Method (Algorithm 2.11) and the Riccati Equation Solution Method (Algorithm 2.12), it is essential to express the relations between the quantities involved in the two algorithms. The recursive part of Algorithm 2.11 is :

$$A_{n+1} = A_n Q_n^{-1} A_n \tag{B.1}$$
$$Q_{n+1} = Q_n - A_n Q_n^{-1} A_n^T - A_n^T Q_n^{-1} A_n \tag{B.2}$$
$$X_{n+1} = X_n - A_n^T Q_n^{-1} A_n \tag{B.3}$$
$$Y_{n+1} = Y_n - A_n Q_n^{-1} A_n^T \tag{B.4}$$

for $n = 1, 2, \ldots$, with initial conditions

$$A_1 = AQ^{-1}A \tag{B.5}$$
$$Q_1 = Q + AQ^{-1}A^T + A^T Q^{-1} A \tag{B.6}$$
$$X_1 = Q + A^T Q^{-1} A \tag{B.7}$$
$$Y_1 = Q + AQ^{-1}A^T \tag{B.8}$$

and

$$X_n \to X_+ \tag{B.9}$$

It is clear that the matrices Q_n, X_n and Y_n are symmetric due to the fact that Q is a symmetric matrix.

The recursive part of Algorithm 2.12 is:

$$\alpha_{n+1} = \alpha_n (\beta_n + \pi_n)^{-1} \alpha_n \tag{B.10}$$
$$\beta_{n+1} = \beta_n - \alpha_n^T (\beta_n + \pi_n)^{-1} \alpha_n \tag{B.11}$$
$$\pi_{n+1} = \pi_n - \alpha_n (\beta_n + \pi_n)^{-1} \alpha_n^T \tag{B.12}$$

for $n = 1, 2, \ldots$, with initial conditions

$$\alpha_1 = Q^{-1} A^T A^{-1} \tag{B.13}$$
$$\beta_1 = A^{-T} A Q^{-1} A^T A^{-1} + A^{-T} Q A^{-1} \tag{B.14}$$
$$\pi_1 = Q^{-1} \tag{B.15}$$

and

$$\pi_n \to \Pi \tag{B.16}$$

where

$$P = \Pi^{-1} \tag{B.17}$$
$$X_+ = P \tag{B.18}$$

It is clear that the matrices β_n and π_n are symmetric due to the fact that Q is a symmetric matrix.

We are going to prove have by induction that the following relations between the quantities involved in the two algorithms hold:

$$Q_n = A^T(\beta_n + \pi_n)A \tag{B.19}$$
$$A_n = A^T \alpha_n^T A \tag{B.20}$$
$$X_n = Q + A^T \pi_n A \tag{B.21}$$
$$Y_n = A^T \beta_n A \tag{B.22}$$

for $n = 1, 2, \ldots.$

Concerning the initial conditions of the two algorithms, it can be trivially verified that:

$Q_1 = A^T(\beta_1 + \pi_1)A,$ from (B.6), (B.14), (B.15)
$A_1 = A^T \alpha_1^T A,$ from (B.5), (B.13)
$X_1 = Q + A^T \pi_1 A,$ from (B.7), (B.15)
$Y_1 = A^T \beta_1 A,$ from (B.8), (B.14)

Concerning the recursive parts of the two algorithms, after some algebra we have:

$Q_{n+1} = A^T(\beta_{n+1} + \pi_{n+1})A,$ from (B.2), (B.11), (B.12), (B.19), (B.20)
$A_{n+1} = A^T \alpha_{n+1}^T A,$ from (B.1), (B.10), (B.19), (B.20)
$X_{n+1} = Q + A^T \pi_{n+1} A,$ from (B.3), (B.12), (B.19), (B.20), (B.21)
$Y_{n+1} = A^T \beta_{n+1} A,$ from (B.4), (B.11), (B.19), (B.20), (B.22)

Finally, from (B.9), (B.16), (B.17), (B.18) and (B.21), we conclude that the limiting solutions X_+ of the two algorithms satisfy $X_+ - A^T X_+^{-1} A = Q$. It is obvious that both algorithms compute the maximal solution X_+ of (2). This completes the proof that the Cyclic Reduction Method (Algorithm 2.11) and the Riccati Equation Solution Method (Algorithm 2.12) are equivalent to each other.

Concerning the minimal solution X_- of (2), it is known [10] that Algorithm 2.11 provides the minimal solution X_- of (2) due to $\lim_{n\to\infty} Y_n = Q - X_-$ and hence Algorithm 2.12 provides the minimal solution X_- of (2) due to $\lim_{n\to\infty} \beta_n = B,$ since it is easy to derive $X_- = Q - A^T B A$ using (B.22).

References

[1] B.D.O. Anderson and J.B. Moore, *Optimal Filtering,* Prentice Hall inc., 1979.

[2] W.N.Jr. Anderson, T.D. Morley and G.E. Trapp, Positive solutions to $X = A - BX^{-1}B^*$, *Linear Algebra Appl.,* **134** (1990), 53-62.

[3] P. Benner and H. Fassbender, On the solution of the rational matrix equation $X = Q + LX^{-1}L^T$, *EURASIP Journal on Advanced Signal Processing,* **2007** (2007), 1-10.

[4] J.C. Engwerda, On the existence of a positive definite solution of the matrix equation $X + A^T X^{-1} A = I$, *Linear Algebra Appl.,* **194** (1993), 91-108.

[5] J.C. Engwerda, A.C.M. Ran and A.L. Rijkeboer, Necessary and sufficient conditions for the existence of a positive definite solution of the matrix equation $X + A^* X^{-1} A = Q$, *Linear Algebra Appl.,* **186** (1993), 255-275.

[6] A. Ferrante and B.C. Levy, Hermitian solutions of the equation $X = Q + NX^{-1}N^*$, *Linear Algebra Appl.,* **247** (1996), 359-373.

[7] C.H. Guo, Newton's method for discrete algebraic Riccati equations when the closed-loop matrix has eigenvalues on the unit circle equation, *SIAM J. Matrix Anal. Appl.,* **20** (1999), 279-294.

[8] C.H. Guo and P. Lancaster, Iterative solution of two matrix equations, *Mathematics of Computation,* **68** no. 228, (1999), 1589-1603.

[9] D.G. Lainiotis, N.D. Assimakis and S. K. Katsikas, Fast and numerically robust recursive algorithms for solving the discrete time Riccati equation: The case of nonsingular plant noise covariance matrix, *Neural Parallel and Scientific Computations,* **3** no. 4, (1995), 565-584.

[10] B. Meini, Efficient computation of the extreme solutions of $X + A^* X^{-1} A = Q$ and $X - A^* X^{-1} A = Q$, Hermitian solutions of the equation $X = Q + NX^{-1}N^*$, *Mathematics of Computation,* **71** no. 239, (2001), 1189-1204.

[11] X. Zhan, Computing the extremal positive definite solutions of a matrix equation, *SIAM J. Sci. Comput.,* **17** no. 5, (1996), 1167-1174.

[12] X. Zhan and J. Xie, On the matrix equation $X + A^T X^{-1} A = I$, *Linear Algebra Appl.,* **247** (1996), 337-345.

Received: December 15, 2007