J. Austral. Math. Soc. Ser. B 40(1999), 459–468

ON THE SQUARE-ROOT METHOD FOR CONTINUOUS-TIME ALGEBRAIC RICCATI EQUATIONS

LINZHANG LU[1] and C. E. M. PEARCE[2]

(Received 12 September 1995; revised 2 October 1997)

Abstract

We give a simple and transparent proof for the square-root method of solving the continuous-time algebraic Riccati equation. We examine some benefits of combining the square-root method with other solution methods. The iterative square-root method is also discussed. Finally, paradigm numerical examples are given to compare the square-root method with the Schur method.

1. Introduction

Algebraic Riccati equations play a fundamental role in the analysis, synthesis and design of linear-quadratic Gaussian control and estimation systems. A central question is the efficient determination of the unique nonnegative-definite, symmetric solution X of the continuous-time algebraic Riccati equation

$$A^T X + XA - XBR^{-1}B^T X + Q = O. \tag{1.1}$$

Here the matrices are real, A, X and Q are $n \times n$, B is $n \times m$ and R is $m \times m$. The matrix R is positive definite and Q nonnegative-definite. Both are symmetric. For convenience we shall also express this equation as

$$A^T X + XA - XGX + Q = O.$$

There are no entirely satisfactory solution procedures. There are some efficient ones, but they are not stable. Laub [5] proposed a Schur method based on the associated Hamiltonian matrix

$$H = \begin{pmatrix} A & -G \\ -Q & -A^T \end{pmatrix}. \tag{1.2}$$

[1]Mathematics Department, Xiamen University, Fujian, P. R. China
[2]Applied Mathematics Department, Adelaide University, Adelaide SA 5005, Australia

https://doi.org/10.1017/S0334270000010547 Published online by Cambridge University Press

A $2n \times 2n$ real matrix H is called (skew-) Hamiltonian if JH is (skew-) symmetric, where

$$J = \begin{pmatrix} O & I \\ -I & O \end{pmatrix}.$$

Following Byers [1], there have been a number of methods for solving (1.1) involving finding a basis for the stable invariant subspace of H. One approach is to use a series of similarity transformations to reduce H to a block upper-triangular form $\begin{pmatrix} C & D \\ O & -C^T \end{pmatrix}$ with C containing only stable eigenvalues. As is observed in [1], it is difficult to do this with a stable similarity transformation. However, as van Loan [9] has shown, it is easy to reduce a skew-Hamiltonian matrix to such a form by orthogonal and symplectic similarity transformations. We call a matrix S symplectic if $S^T J S = J$. Here and subsequently the superscript T denotes 'transpose'.

Recently Hongguo Xu and Linzhang Lu [9] proposed a way of utilizing van Loan's idea *via* a "square-root" technique. It is readily verified that JH^2 is skew–symmetric, so that H^2 is skew-Hamiltonian and van Loan's algorithm is applicable. The main task of the technique proposed in [9] is the computation of the principal square root of H^2.

The justification of the square-root technique in [9] turned out to be quite lengthy. In Section 2 we present a very short and simple justification.

We then turn to the implementation of the square-root approach. It can be beneficial to use it in combination with other techniques. In Section 3 we examine it in conjunction with the sign-function method and show how the latter can be used to prevent our having to solve an overdetermined system. In Section 4 we consider the determination of the principal square root of H^2 by iteration. We conclude in Section 5 with some numerical experiments which compare the square-root approach with a Schur approach using benchmark examples given in Laub [5].

2. A simple proof of the square-root method

Let $\lambda = \rho e^{i\theta}$ be a complex scalar, with $\rho > 0$ and $|\theta| < \pi$. The principal square root of λ is defined as $\rho^{1/2} e^{i\theta/2}$. This definition may be extended to cover a general square matrix as follows.

DEFINITION 2.1. Let A be a nonsingular matrix. A matrix Y is called the principal square root of A if $Y^2 = A$ and $\operatorname{Re}\lambda(Y) > 0$ for each eigenvalue $\lambda(Y)$ of Y.

It is well-known that if A is a real nonsingular matrix having no negative real eigenvalues, then A has a unique principal square root (see, for example, Gantmacher

[3]). We shall denote the principal square root of a matrix A by sqrt(A). It is obvious that for any nonsingular matrix P,

$$\text{sqrt}(A) = P^{-1}\,\text{sqrt}(PAP^{-1})P. \tag{2.1}$$

The matrix square-root technique for solving (1.1) is based on the following result given in [9].

THEOREM 2.2. *Let H be a $2n \times 2n$ Hamiltonian matrix with no eigenvalues on the imaginary axis. Then the first n columns of $H -$ sqrt(H^2) span the invariant subspace of H corresponding to its eigenvalues with negative real part, that is, the stable invariant subspace.*

Suppose that the coefficient matrices in (1.1) are such that (A, B) is stabilizable and (C, A) detectable, where C arises from the full-rank factorization $Q = C^T C$ of Q. It is well-known [5] that under these mild conditions we have that

(a) the Hamiltonian matrix H corresponding to (1.1) has no purely imaginary eigenvalues;
(b) a nonnegative-definite solution X exists, is unique and satisfies

$$\text{Re}\,\lambda(A - GX) < 0; \tag{2.2}$$

(c) if $[Z_1^T, Z_2^T]^T$ is a basis for the stable invariant subspace of H, then $X = Z_2 Z_1^{-1}$.

In this paper we suppose these results hold, so that H^2 has no zero or negative real eigenvalues and sqrt(H^2) exists. Put $W = H -$ sqrt(H^2) and let W be partitioned as

$$W = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix}, \tag{2.3}$$

where each W_{ij} is an $n \times n$ matrix.

It was shown in [9] that the unique nonnegative-definite solution to (1.1) is

$$X = W_{21} W_{11}^{-1}. \tag{2.4}$$

This we now derive in a much simpler and shorter way. We restate Theorem 2.2 in the following direct form.

THEOREM 2.3. *Let H as defined in* (1.3) *be a $2n \times 2n$ Hamiltonian matrix corresponding to* (1.1) *and let $W = H -$ sqrt(H^2) be partitioned as in* (2.3). *Then the unique nonnegative-definite solution X to* (1.2) *is given by* (2.4).

https://doi.org/10.1017/S0334270000010547 Published online by Cambridge University Press

PROOF. Let

$$S = \begin{pmatrix} I & Y \\ O & I \end{pmatrix} \begin{pmatrix} I & O \\ -X & I \end{pmatrix} = \begin{pmatrix} I - YX & Y \\ -X & I \end{pmatrix},$$

where X is the unique nonnegative-definite solution to (1.1) and the symmetric matrix Y satisfies the Lyapunov equation

$$(A - GX)Y + Y(A - GX)^T = -G. \tag{2.5}$$

It is easy to verify from the symmetry of X and Y that S is a symplectic matrix. Further, we have

$$S^{-1} = \begin{pmatrix} I & -Y \\ X & I - XY \end{pmatrix}.$$

From (1.2), (2.5) and the definition of G, we derive

$$SHS^{-1} = S \begin{pmatrix} A & -G \\ -Q & -A^T \end{pmatrix} S^{-1} = \begin{pmatrix} A - GX & 0 \\ 0 & -(A - GX)^T \end{pmatrix},$$

so that

$$SH^2S^{-1} = \operatorname{diag}((A - GX)^2, ((A - GX)^T)^2).$$

From (2.1), (2.2) and Definition 2.1, we must have that

$$\operatorname{sqrt}(H^2) = S^{-1} \operatorname{diag}(-(A - GX), -(A - GX)^T)S.$$

Therefore

$$W = H - \operatorname{sqrt}(H^2) = S^{-1} \operatorname{diag}(2(A - GX), O)S$$

and

$$\begin{pmatrix} W_{11} \\ W_{21} \end{pmatrix} = \begin{bmatrix} 2(A - GX)(I - YX) \\ 2X(A - GX)(I - YX) \end{bmatrix}. \tag{2.6}$$

The matrix $A - GX$ is nonsingular because of (2.2). Also $I - YX$ is nonsingular because S is symplectic and $I - YX$ is a (1,1) block of S (see Laub [5]). Therefore $(A - GX)(I - YX)$ is nonsingular. The desired result (2.4) follows directly from (2.6).

3. Utilization of other methods

The square-root technique can be sharpened by judicious combination with other algorithms. For example, we may utilize van Loan's algorithm [8] when computing $\text{sqrt}(H^2)$. Since H^2 is skew-Hamiltonian, we can, as in [8], easily compute an orthogonal symplectic matrix P such that

$$P^T H^2 P = \begin{bmatrix} U & V \\ O & U^T \end{bmatrix} \equiv M, \tag{3.1}$$

where U is upper Hessenberg and V skew-Hamiltonian.

By (2.1), we get

$$\text{sqrt}(H^2) = P\,\text{sqrt}(P^T H^2 P) P^T = P\,\text{sqrt}(M) P^T = P\,\text{sqrt}\begin{bmatrix} U & V \\ O & U^T \end{bmatrix} P^T.$$

To compute $\text{sqrt}(M)$ we have only to compute $\text{sqrt}(U)$ and then solve a special Lyapunov equation

$$\text{sqrt}(U) Y + Y(\text{sqrt}(U))^T = V. \tag{3.2}$$

Note that $\text{sqrt}(U^T) = (\text{sqrt}(U))^T$ and U is only half the size of H.

We may also use iteration to compute $\text{sqrt}(M)$ directly, as discussed in the next section. Either way we can save on operations and storage requirements.

We now analyze the relationship between the square-root method and the sign-function method and exploit another advantage of the square-root approach.

Let λ be a complex scalar with $\text{Re}(\lambda) \neq 0$. Then the sign of λ is defined by

$$\text{sign}(\lambda) = \begin{cases} 1, & \text{if } \text{Re}(\lambda) > 0, \\ -1, & \text{if } \text{Re}(\lambda) < 0. \end{cases}$$

The scalar sign function can also be expressed as

$$\text{sign}(\lambda) = \lambda / \text{sqrt}(\lambda^2).$$

This can be seen easily by taking $\lambda = \rho e^{i(\theta+\pi k)}$ with $\rho > 0$ and $|\theta| < \pi/2$, where $k = 0$ or 1 according as $\text{Re}(\lambda) > 0$ or < 0. By squaring we have

$$\lambda^2 = \rho^2 e^{i2(\theta+\pi k)}.$$

Since $\text{sqrt}(\lambda^2) = \rho e^{i\theta}$, we obtain

$$\lambda / \text{sqrt}(\lambda^2) = \rho e^{i(\theta+\pi k)} / \rho e^{i\theta} = e^{i\pi k} = \text{sign}(\lambda).$$

To extend the scalar function definition to a general square matrix A, we use

$$\mathrm{sign}(A) = A(\mathrm{sqrt}(A^2))^{-1} = A^{-1}(\mathrm{sqrt}(A^2)),$$

that is,

$$\mathrm{sqrt}(A^2) = A(\mathrm{sign}(A)) = (\mathrm{sign}(A))A. \tag{3.3}$$

Note that once $\mathrm{sign}(H)$ is computed by the sign-function method (see Denman and Beavers [2]), to obtain the unique nonnegative-definite solution X to (1.1) we have to solve an overdetermined system

$$\begin{bmatrix} V_{12} \\ V_{22} + I \end{bmatrix} X = - \begin{bmatrix} V_{11} + I \\ V_{21} \end{bmatrix}, \tag{3.4}$$

where

$$\begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} = \mathrm{sign}(H).$$

In terms of Theorem 2.3 and (3.3), solving the overdetermined system (3.4) can be avoided. Only the first n columns of $\mathrm{sign}(H)$ are needed for the computation. Once $\begin{bmatrix} V_{11} \\ V_{21} \end{bmatrix}$ is computed, premultiplication by H suffices to derive X.

4. Iteration to compute $\mathrm{sqrt}(H^2)$

The Newton-Raphson algorithm for computing $\mathrm{sqrt}(H^2)$ is based on

$$Y_{k+1} = (Y_k + Y_k^{-1}H^2)/2, \qquad Y_0 = I.$$

A faster and more stable algorithm proposed in [4] and [7] employs

$$X_{k+1} = \alpha_k X_k + \beta_k Z_k^{-1} \tag{4.1a}$$

and

$$Z_{k+1} = \alpha_k Z_k + \beta_k X_k^{-1}, \tag{4.1b}$$

with $X_0 = H^2$ and $Z_0 = I$, where α_k and β_k are scale factors chosen for stability and rapid convergence of the iteration. To be specific,

$$\text{for } k \geq 0 \quad \begin{cases} \alpha_k^2 = 2/(p_k + q_k + 6\sqrt{p_k q_k}), & \beta_k^2 = p_k q_k \alpha_k^2, \\ \epsilon_k = 1 - 4\alpha_k\beta_k, \quad p_{k+1} = 1 - \epsilon_k, & q_{k+1} = 1 + \epsilon_k, \end{cases} \tag{4.1c}$$

https://doi.org/10.1017/S0334270000010547 Published online by Cambridge University Press

with $p_0 = 1/\ \ H^{-2}\ \ , q_0 = \ \ H^2\ \ $. Either the 1-norm or the 2-norm may be employed.

But it has been shown (see (3.3) and (3.4) in [6]) that the iteration (4.1) is equivalent to the iteration

$$Y_{k+1} = \alpha_k Y_k + \beta_k Y_k^{-1} H^2, \qquad Y_0 = I, \tag{4.2a}$$

$$\alpha_0^2 = p_0 q_0 \beta_0^2, \qquad \beta_0^2 = 2/(p_0 + q_0 + 6\sqrt{p_0 q_0}), \tag{4.2b}$$

with p_0, q_0 as before and

$$\text{for } k \geq 1 \quad \begin{cases} \alpha_k^2 = 2/(p_k + q_k + 6\sqrt{p_k q_k}), & \beta_k^2 = p_k q_k \alpha_k^2, \\ \epsilon_{k-1} = 1 - 4\alpha_{k-1}\beta_{k-1}, \quad p_k = 1 - \epsilon_{k-1}, & q_k = 1 + \epsilon_{k-1}. \end{cases} \tag{4.2c}$$

Under our assumption that H has no eigenvalues on the imaginary axis, H^2 has no zero or negative real eigenvalues and (Y_k) will converge to sqrt(H^2). Since Y_k commutes with H, (4.2a) can be rewritten as

$$Y_{k+1} = \alpha_k Y_k + \beta_k H Y_k^{-1} H, \qquad Y_0 = I, \tag{4.3}$$

where α_k, β_k are as in (4.2b–c).

On premultiplication by J in (4.3), we derive

$$J Y_{k+1} = \alpha_k J Y_k + \beta_k J H Y_k^{-1} J^T J H, \qquad J Y_0 = J,$$

since $J^T J = I$. Let $Z_k = J Y_k$ and $C = JH$. Then we obtain

$$Z_{k+1} = \alpha_k Z_k + \beta_k C Z_k^{-1} C, \qquad Z_0 = J. \tag{4.4}$$

Because H is Hamiltonian, C is symmetric. There exists an orthogonal matrix U and a diagonal matrix D such that $UCU^T = D$. Let $P_k = UZ_kU^T$, so (4.4) becomes

$$P_{k+1} = \alpha_k P_k + \beta_k D P_k^{-1} D, \qquad P_0 = UJU^T. \tag{4.5}$$

Because D is diagonal, (4.5) provides a very simple iteration. Furthermore, we claim that P_k is skew-symmetric. In fact, since J is skew-symmetric, so is P_0 and so also P_k from the recurrence (4.5). Thus the symmetric structure of the Hamiltonian H is exploited in iteration (4.5) to save some computation and storage. Clearly (P_k) converges to $J^T U^T$ sqrt$(H^2)U$.

With M defined by (3.1), we can compute sqrt(M) by the iteration

$$T_{k+1} = \alpha_k T_k + \beta_k T_k^{-1} M, \qquad T_0 = I. \tag{4.6}$$

TABLE 1. Comparison of the Schur method and the square-root method.

	CPU time (seconds)		$\max\{\lvert L\rvert_{ij}\}$	
Example	Schur	sqrt method	Schur	sqrt method
1	0.01	0.01	3.0×10^{-15}	2.3×10^{-13}
2	0.01	0.01	3.3×10^{-13}	3.8×10^{-13}
4 (N=5)	0.09	0.02	9.2×10^{-14}	8.0×10^{-15}
4 (N=10)	0.72	0.08	9.6×10^{-14}	2.0×10^{-14}
4 (N=20)	29.05	0.70	8.5×10^{-13}	6.4×10^{-14}
5	531.63	2.59	3.8×10^{-15}	2.1×10^{-15}
6 ($n, q, r = 11, 1, 1$)	0.04	0.04	5.5×10^{-8}	1.4×10^{-4}
6 ($q = 10^4$)	0.05	0.05	2.6×10^{-2}	2.1×10^{-1}
6 ($n, q, r = 21, 1, 1$)	0.15	0.15	$4.6 \times 10^{+2}$	$1.3 \times 10^{+6}$
6 ($q = 10^4$)	0.15	0.78	$5.4 \times 10^{+9}$	$1.1 \times 10^{+9}$

Let

$$T_k = \begin{bmatrix} T_{11}(k) & T_{12}(k) \\ T_{21}(k) & T_{22}(k) \end{bmatrix}.$$

It is easy to verify that

$$T_{21}(k) = O, \qquad T_{22}(k) = T_{11}^T(k)$$

and that $T_{12}(k)$ is skew-symmetric. So iteration (4.6) can be reduced to

$$T_{11}(k+1) = \alpha_k T_{11}(k) + \beta_k T_{11}^{-1}(k) U, \qquad T_{11}(0) = I, \tag{4.7}$$

$$T_{12}(k+1) = \alpha_k T_{12}(k) + \beta_k T_{11}^{-1}(k)(V - T_{12}(k) T_{11}^{-T}(k) U^T), \quad T_{12}(0) = O. \tag{4.8}$$

In fact (4.7) computes sqrt(U) and (4.8) Y in (3.2).

5. Numerical examples

We now test our square-root method against the Schur method of Laub [5], using a set of benchmark paradigm examples from [5]. MatLab programs were written for the two algorithms. The code hqr5.m (by Richard Y. Chiang) to produce an ordered Complex Schur Form was downloaded from http://www.mathworks.com.

TABLE 2. Estimated condition number of U_{11} or W_{11}.

Example	cond(U_{11}) (Schur)	cond(W_{11}) (sqrt method)
6 ($n, q, r = 11, 1, 1$)	$2.9 \times 10^{+4}$	$8.0 \times 10^{+8}$
6 ($q = 10^4$)	$5.7 \times 10^{+6}$	$5.5 \times 10^{+9}$
6 ($n, q, r = 21, 1, 1$)	$2.4 \times 10^{+9}$	$6.6 \times 10^{+15}$
6 ($q = 10^4$)	$3.5 \times 10^{+11}$	$3.6 \times 10^{+16}$

The algorithm used to compute sqrt(H^2) is described in (4.2). The computations are carried out on an Ultra-1 Sun workstation.

We compare CPU times for the two methods using Examples 1, 2 and 4–6 in [5]. (Example 3 is a discrete-time problem.) Chiang's code did not lend itself to a storage comparison. The results are listed in Table 1, in which

$$L = A^T X^* + X^* A - X^*(BR^{-1}B^T)X^* + Q,$$

where X^* is the solution obtained by applying the algorithms. Clearly $\max\{|L|_{ij}\}$ is a measure of the accuracy of the solution.

Observations

(1) Both methods give a satisfactorily accurate solution to all the problems other than Example 6. The square-root method was comparable or significantly faster than the Schur method except in the rather small problem of Example 1.

(2) Both methods failed to solve Example 6 due to the ill-conditioned nature of U_{11} or W_{11}.

Acknowledgements

The work was completed during a visit of the first author at the Australian National University. He would like to thank the Centre for Mathematics and its Applications for its hospitality and especially Professor Mike Osborne for his kind guidance and encouragement.

References

[1] R. Byers, "Hamiltonian and symplectic algorithms for algebraic Riccati equations", Ph. D. Thesis, Cornell Univ., Ithaca, NY, 1982.

[2] D. Denman and A. N. Beavers, "The matrix sign function and computation in systems", *Appl. Math. Comput.* **2** (1976) 63–94.

[3] F. R. Gantmacher, *The theory of matrices* (Chelsea, New York, 1959).

[4] W. D. Hoskins and D. J. Walton, "A faster method of computing the square root of a matrix", *IEEE Trans. Automat. Control* **23** (1978) 494–495.

[5] A. J. Laub, "A schur method for solving algebraic Riccati equations", *IEEE Trans. Automat. Control* **24** (1979) 913–921.

[6] Linzhang Lu, "On sign function and square root methods for solution of real algebraic Riccati equations", *Numer. Math., J. Chinese Univ.* **1** (1992) 81–90.

[7] Linzhang Lu and Wenwei Lin, "An iterative algorithm for the solution of the discrete-time algebraic Riccati equation", *Lin. Alg. Appl.* **188-189** (1993) 465–488.

[8] C. F. van Loan, "A symplectic method for approximating all eigenvalues of a Hamiltonian matrix", *Lin. Alg. Appl.* **61** (1984) 233–251.

[9] Hongguo Xu and Linzhang Lu, "Properties of a quadratic matrix equation and the solution of the continuous-time algebraic Riccati equation", *Lin. Alg. Appl.* **222** (1995) 127–145.