Electronic Transactions on Numerical Analysis
Volume 1, pp. 33-48, September 1993

ETNA
Kent State University
etna@mcs.kent.edu

A MULTISHIFT ALGORITHM FOR THE NUMERICAL SOLUTION OF ALGEBRAIC RICCATI EQUATIONS *

GREGORY AMMAR†, PETER BENNER‡, AND VOLKER MEHRMANN§

Abstract. We study an algorithm for the numerical solution of algebraic matrix Riccati equations that arise in linear optimal control problems. The algorithm can be considered to be a multishift technique, which uses only orthogonal symplectic similarity transformations to compute a Lagrangian invariant subspace of the associated Hamiltonian matrix. We describe the details of this method and compare it with other numerical methods for the solution of the algebraic Riccati equation.

Key words. algebraic matrix Riccati equation, Hamiltonian matrix, Lagrangian invariant subspace.

AMS subject classifications. 65F15, 15A24, 93B40.

1. Introduction.

We consider the numerical solution of algebraic matrix Riccati equations of the form

$$G + A^T X + XA - XRX = 0, \tag{1}$$

with $A, G, R \in \mathbf{R}^{n,n}$, and where G and R are symmetric positive semidefinite matrices. These equations arise in linear quadratic optimal control problems, differential games, and Kalman filtering problems. In these applications the symmetric positive semidefinite solution X of (1) is often desired; this is called a *stabilizing solution* because the eigenvalues of the resulting closed-loop matrix $A - RX$ are in the open complex left-half plane. The existence and uniqueness of such a solution is guaranteed by certain assumptions on the problem. See, for example, [10, 14].

It is easy to see that the matrix X is a solution of (1) if and only if the columns of $\begin{bmatrix} I_n \\ X \end{bmatrix}$ span an n-dimensional invariant subspace of the *Hamiltonian matrix*

$$H = \begin{bmatrix} A & R \\ G & -A^T \end{bmatrix} \in \mathbf{R}^{2n,2n} \tag{2}$$

where I_n is the $n \times n$ identity matrix. Moreover, it is well known [15, 20] that the unique positive semidefinite solution of (1), when it exists, can be obtained from the *stable* invariant subspace of H; i.e., from the invariant subspace corresponding to the eigenvalues of H with negative real parts. More precisely, we have the following well-known result (see [20]).

THEOREM 1.1. *Assume that the algebraic Riccati equation* (1) *has a unique stabilizing symmetric positive semidefinite solution X. Then the Hamiltonian matrix H given in* (2) *has precisely n eigenvalues with negative real parts. Furthermore, if the columns of $\begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \in \mathbf{R}^{2n,n}$ span the invariant subspace of H corresponding to*

* Received July 9, 1993. Accepted for publication August 10, 1993. Communicated by L. Reichel.
† Department of Mathematical Sciences, Northern Illinois University, DeKalb, IL 60115, USA. Email: ammar@math.niu.edu
‡ Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, D-52056 Aachen, FRG. Present address: Department of Mathematics, 405 Snow Hall, University of Kansas, Lawrence, KS, USA. Email: benner@mercury.math.ukans.edu
§ Fachbereich Mathematik, Technische Universität Chemnitz-Zwickau, Reichenhainerstr. 41, D–O–09009 Chemnitz, FRG. Email: mehrmann@mathematik.tu-chemnitz.de

ETNA
Kent State University
etna@mcs.kent.edu

these eigenvalues, then Q_1 *is invertible and the desired solution of* (1) *is given by* $X := -Q_2Q_1^{-1}$.

In general, the numerical methods for the solution of (1) can be divided in two major classes.

The methods in the first class approach the Riccati equation directly as a nonlinear algebraic equation via fixed point or Newton iteration. The latter is generally attributed to Kleinman [13]. At each step of Newton's method one has to solve a Lyapunov equation, and it can be shown that for an appropriate starting matrix the iteration converges monotonically; see [18]. Unfortunately, it is usually difficult to obtain a starting matrix that guarantees convergence to the desired solution while being close enough to the solution so that the algorithm converges in a reasonable number of iterations. This is the reason why Newton's method is generally most useful in the iterative refinement of solutions obtained from other methods; see [4].

The second class consists of the methods that are based on the computation of the stable invariant subspace of H in (2). These methods include the Schur vector method of Laub [15], the Hamiltonian QR-algorithm of Byers [8], the SR-algorithm of Bunse-Gerstner and Mehrmann [5], the HHDR-algorithm of Bunse-Gerstner and Mehrmann [6] and the matrix sign function method [9].

Unlike earlier attempts based on the computation of eigenvectors of the Hamiltonian matrix H, Laub's method uses the numerically reliable QR-algorithm to compute the desired invariant subspace. This Schur vector method is numerically backwards stable and of complexity $O(n^3)$. However, this algorithm does not respect the Hamiltonian structure of the problem. There has therefore been a significant amount of work devoted to computing the stable invariant subspace of H with a structure-preserving method that could be shown to be *strongly stable*; i.e., the computed solution should be the exact solution of a nearby *Hamiltonian* problem. Both the SR-algorithm and the sign function method respect the Hamiltonian structure, but are not backwards stable. The most promising approach has been the development of a Hamiltonian QR-algorithm.

We now summarize some definitions and basic facts concerning Hamiltonian matrices and QR-type algorithms for computing their invariant subspaces. Let

$$J := \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}.$$

DEFINITION 1.2.
(i) *The matrix* $H \in \mathbf{R}^{2n,2n}$ *is* Hamiltonian *if* $(JH)^T = JH$.
(ii) *The matrix* $S \in \mathbf{R}^{2n,2n}$ *is* symplectic *if* $S^TJS = J$.

Observe that any matrix of the form (2), with R and G symmetric, is a Hamiltonian matrix. Also note that if H is Hamiltonian and S is symplectic, then $S^{-1}HS$ is Hamiltonian.

We will make use of the fact that the n dimensional invariant subspace of H corresponding to the eigenvalues with negative real part is a Lagrangian subspace.

DEFINITION 1.3. *A subspace* $\mathcal{Q}$ *of* $\mathbf{R}^{2n}$ *is called* isotropic *if* $x^TJy = 0$ *for all* $x, y \in \mathcal{Q}$. *A* Lagrangian *subspace is an isotropic subspace of* $\mathbf{R}^{2n}$ *of dimension* n, *or equivalently, a maximal isotropic subspace (i.e., an isotropic subspace that is not contained in a larger isotropic subspace).*

Lagrangian subspaces are important for the control-theoretic Riccati equation because the subspace spanned by the columns of $\begin{bmatrix} I \\ X \end{bmatrix}$ is Lagrangian if and only if

X is symmetric. Consequently, the symmetric solutions of the Riccati equation (1) correspond with the Lagrangian H-invariant subspaces of the form span $\begin{bmatrix} I \\ X \end{bmatrix}$. We will consider an algorithm for the computation of any Lagrangian invariant subspace of a Hamiltonian matrix H.

Structure-preserving methods for computing eigenvalues and invariant subspaces of Hamiltonian matrices are based on the fact that symplectic similarity transformations preserve the Hamiltonian structure and that we can compute the required Lagrangian invariant subspace via a symplectic similarity transformation to a triangular-like form. But symplectic similarity transformations are not necessarily numerically stable, since symplectic matrices can be unbounded in norm and hence transformations with symplectic matrices can cause large roundoff errors. In order to avoid such instabilities, it is desirable to use orthogonal similarity transformations; see, e.g., [26, 12]. There therefore has been considerable effort made toward the development of algorithms for finding the required invariant subspace by performing orthogonal symplectic similarity transformations on the initial Hamiltonian matrix.

Paige and Van Loan [20] first considered the use of orthogonal symplectic transformations on a Hamiltonian matrix. They introduced the Hamiltonian analog of the real Schur form, and gave sufficient conditions for when a Hamiltonian matrix can be transformed by an orthogonal symplectic similarity transformation to Hamiltonian Schur form. This is summarized, and the existence result extended, in the following theorem of Lin [16].

THEOREM 1.4. *Let H be a Hamiltonian matrix whose eigenvalues on the imaginary axis have even algebraic multiplicity. Then there exists an orthogonal symplectic matrix Q such that Q^THQ is in the* Hamiltonian Schur form

$$Q^THQ = \begin{bmatrix} T & W \\ 0 & -T^T \end{bmatrix}, \tag{3}$$

where T is quasi upper triangular and all eigenvalues of T are in the closed complex left-half plane.

Despite the fact that this result has been essentially known now for more than 10 years, no numerically stable algorithm of complexity $O(n^3)$ that operates only with symplectic orthogonal similarities has been found to compute this form.

The only completely satisfactory method for reducing H to Hamiltonian Schur form has been given by Byers [7, 8]. It is a structure preserving, numerically stable QR-like algorithm for the case that the Hamiltonian matrix is in *Hamiltonian Hessenberg form*, i.e.

$$H = \begin{bmatrix} A & R \\ G & -A^T \end{bmatrix} = \begin{bmatrix} \diagdown & \square \\ * & \diagdown \end{bmatrix}, \tag{4}$$

where $A = [a_{ij}] \in \mathbf{R}^{n,n}$ is an upper Hessenberg matrix, $G = \alpha e_n e_n^T$, where e_n is the nth column of I_n, and R is symmetric.

The only obstacle to the use of Byers' Hamiltonian QR-algorithm is that it is not known how to reduce a general Hamiltonian matrix in complexity $O(n^3)$ via orthogonal symplectic similarity transformations to Hamiltonian Hessenberg form. Byers [7, 8] shows that this reduction is possible for the matrix H in (2) if either of the matrices G or R is of rank one. But the reduction for general Hamiltonian matrices remains elusive. In [1], a result is given that indicates why this is such a difficult task. In particular, the first column vector x of an orthogonal symplectic matrix that reduces a Hamiltonian matrix H to Hamiltonian Hessenberg form has to satisfy the set of nonlinear equations

$$x^T J H^{2i-1} x = 0, \quad i = 1, \ldots, n. \tag{5}$$

Actually the same set of equations has to be satisfied even if we use nonorthogonal symplectic transformations, as is shown by Raines and Watkins [22]. Thus, any method for reducing a Hamiltonian matrix to Hamiltonian Hessenberg form must, at least implicitly, solve the nonlinear equations (5).

On the other hand it is observed in [1] that every vector $x \in \mathbf{R}^{2n}$ that is contained in a Lagrangian invariant subspace of H automatically satisfies (5). If we could find such a vector, we might be able to use it to obtain the Lagrangian invariant subspace without transforming the initial matrix to Hamiltonian Schur form. A framework for such a method is suggested in [1]. In this paper we consider an implementation of this method, discuss its advantages and disadvantages, and compare it to other Riccati solvers.

2. A multishift method for the algebraic Riccati equation.

The basic idea of the method proposed in [1] is to compute the invariant subspace corresponding to the stable eigenvalues of the Hamiltonian matrix H, but to do it without iterating to the Hamiltonian Schur form. The key ingredients of this method are the following two structure-preserving methods.

The first is a method due to Van Loan [25] for computing the eigenvalues of a Hamiltonian matrix. This method provides an efficient algorithm for computing the eigenvalues of H, but it cannot be used directly to compute the required invariant subspace.

The second is a reduction procedure to a Hessenberg-like form introduced by Paige and Van Loan [20]. Any Hamiltonian matrix can be reduced to this form in $O(n^3)$ arithmetic operations using orthogonal symplectic similarity transformations. However, this condensed form is not invariant under the Hamiltonian QR-algorithm, so it cannot be used for a structure preserving QR-like algorithm.

This reduction is achieved by performing a sequence of elementary orthogonal symplectic similarity transformations to the initial matrix. These elementary transformations are of two types (see, e.g., [20, 8, 18]). The first is a *symplectic Householder matrix,* which is a matrix of the form

$$P = \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix},$$

where W is a Householder transformation of order n. The second is a *symplectic Givens rotation,* which is a standard Givens rotation of order $2n$ that operates only in coordinate planes j and $n+j$ for some j, $1 \leq j \leq n$.

We now describe the reduction procedure applied to an arbitrary matrix of order $2n$.

Algorithm 2.1 (PVLRED).
Input: $M \in \mathbf{R}^{2n,2n}$

Output: Orthogonal symplectic Q such that $Q^T M Q = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$, where M_{11} is an upper Hessenberg matrix and M_{21} is upper triangular. M will be overwritten with the transformed matrix.

Set $Q := I_{2n}$.
FOR $k = 1, \ldots, n-1$
IF $k \leq n-2$ THEN
Let $\begin{bmatrix} y \\ z \end{bmatrix} := Me_k$ and let P_k be the Householder symplectic matrix $\operatorname{diag}[W_k, W_k]$, where W_k is a Householder transformation of order n that operates in coordinate planes $k+1$ through n and annihilates elements $k+2, \ldots, n$ of z.
Set $M := P_k M P_k,\ Q := QP_k$.
END IF

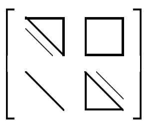
Let G_k be a symplectic Givens rotation such that $M_{n+k+1,k}$ is eliminated with $M_{k+1,k}$ when forming $G_k^T M$.
Set $M := G_k^T M G_k,\ Q := QG_k$

IF $k \leq n-2$ THEN
Let $\begin{bmatrix} y \\ z \end{bmatrix} := Me_k$ and let $\tilde{P}_k$ be the Householder symplectic matrix $\operatorname{diag}[\tilde{W}_k, \tilde{W}_k]$, where $\tilde{W}_k$ is a Householder transformation of order n that operates in coordinate planes $k+1$ through n and annihilates elements $k+2, \ldots, n$ of y.
Set $M := \tilde{P}_k M \tilde{P}_k,\ Q := Q\tilde{P}_k$.
END IF
END FOR
END.

Observe that the transformation Q generated by Algorithm 2.1 does not involve the first axis vector e_1; that is, $Q^T e_1 = e_1$.

If the initial matrix M in Algorithm 2.1 is Hamiltonian, then it is reduced to the following form, which will be called *Paige-Van Loan form*:

(6) $$Q^T MQ =$$



The algorithm of Van Loan [25] for the computation of the eigenvalues of a Hamiltonian matrix has the following form.

ALGORITHM 2.2 (SQRED).

Input: Hamiltonian matrix $H = \begin{bmatrix} A & R \\ G & -A^T \end{bmatrix}$.

Output: Eigenvalues $\lambda_1, \ldots, \lambda_{2n}$ of H.

Step 1: Form

$$N = H^2 = \begin{bmatrix} A^2 + RG & AR - RA^T \\ GA - A^T G & GR + (A^T)^2 \end{bmatrix}. \tag{7}$$

Step 2: Apply Algorithm 2.1 *PVLRED* to N to determine an orthogonal symplectic

ETNA
Kent State University
etna@mcs.kent.edu

matrix Q such that

$$Q^T N Q = \begin{bmatrix} N_1 & N_2 \\ 0 & N_1^T \end{bmatrix} = \begin{bmatrix} \diagdown & \square \\ & \diagdown \end{bmatrix}. \tag{8}$$

Step 3: Determine the eigenvalues $\mu_1, \ldots, \mu_n$ of N_1 with the QR–Algorithm; see, e.g., [12, 23].

Step 4: Set $\lambda_i = \sqrt{\mu_i}$ and $\lambda_{i+n} = -\lambda_i$ for $i = 1, \ldots, n$.

END.

The only disadvantage of this method is that the Hamiltonian matrix has to be squared, which may lead to roundoff errors on the order of the square root of machine precision [25].

Based on these two methods, the basic framework of the method suggested in [1] is the following:

ALGORITHM 2.3 (LISH).

Input: A Hamiltonian matrix $H \in \mathbf{R}^{2n,2n}$ having an n–dimensional Lagrangian invariant subspace $\mathcal{Q}$, the corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$, and a tolerance eps.

Output: A real orthogonal symplectic matrix $Q \in \mathbf{R}^{2n,2n}$ such that the first n columns of Q span the Lagrangian subspace of H corresponding to the other eigenvalues $\lambda_{n+1}, \ldots, \lambda_{2n}$ of H.

Set $Q := I$.

Step 1 (Computation of the first column of the transformation matrix.)

Form

$$x = \alpha(H - \lambda_1 I) \cdots (H - \lambda_n I) e_1, \tag{9}$$

where $\alpha \in \mathbf{R}$ is an arbitrary nonzero scaling factor, and let $Q_1 \in \mathbf{R}^{2n,2n}$ be an orthogonal symplectic matrix such that

$$Q_1^T x = \alpha_1 \, e_1, \;\; \alpha_1 = \pm \|x\|. \tag{10}$$

Such a matrix is constructed in the obvious way, analogous to the construction used in Algorithm 2.1; see [20]. Set

$$H := Q_1^T H Q_1, \;\; Q := Q Q_1 \tag{11}$$

Step 2 (Reduction to Paige/Van Loan form.)

Use Algorithm 2.1 PVLRED to generate an orthogonal symplectic matrix Q_2 such that $Q_2^T H Q_2$ is in the form (6).

Set

$$H := Q_2^T H Q_2, \;\; Q := Q Q_2. \tag{12}$$

Step 3 (Deflation.)

Set $p := 0$.

WHILE $p < n$

  FOR $i = 1, \ldots, n$

    IF $|h_{n+i,i}| <$ eps THEN set $h_{n+i,i} := 0$.

  END FOR

ETNA
Kent State University
etna@mcs.kent.edu

FOR $i = 2, \ldots, n$
  IF $|h_{i,i-1}| <$ eps THEN set $h_{i,i-1} := 0$.
END FOR
Set $\ell := p + 1$, and let p be the largest integer in $\{\ell, \ldots, n\}$ such that $h_{p+1,p} = 0$ and $h_{n+k,k} = 0$ for $k = 1, \ldots, p$. If no such integer exists, then set $p := n$.
IF $p < n$ THEN
  Partition H as

$$\begin{array}{cc} \left[\begin{array}{cccc} A_{11} & A_{12} & R_{11} & R_{12} \\ 0 & A_{22} & R_{21} & R_{22} \\ 0 & 0 & -A_{11}^T & 0 \\ 0 & G_{22} & -A_{12}^T & -A_{22}^T \end{array}\right] & \begin{array}{c} p \\ n-p \\ p \\ n-p \end{array} \end{array} \tag{13}$$

  Set $H_{22} := \begin{bmatrix} A_{22} & R_{22} \\ G_{22} & -A_{22}^T \end{bmatrix}$ and

$$x_2 := \alpha_2 (H_{22} - \lambda_1 I) \cdots (H_{22} - \lambda_n I) e_1. \tag{14}$$

  Let P_1 be an orthogonal symplectic matrix such that $P_1^T x_2 = \pm \|x_2\| e_1$ and determine as in the second step an orthogonal symplectic matrix P_2 that reduces H_{22} to the Paige/Van Loan form via Algorithm 2.1 PVLRED. Set

$$P := P_1 P_2 =: \begin{bmatrix} U & V \\ -V & U \end{bmatrix} \in \mathbf{R}^{2(n-p),2(n-p)} \tag{15}$$

  and

$$\tilde{Q}_p := \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & U & 0 & V \\ 0 & 0 & I & 0 \\ 0 & -V & 0 & U \end{bmatrix}, \; H := \tilde{Q}_p^T H \tilde{Q}_p, \; Q := Q\tilde{Q}_p \tag{16}$$

END IF
END WHILE
END.

In this algorithm only orthogonal symplectic transformations are used, and hence the Hamiltonian structure is preserved. This is done implicitly to avoid a deterioration due to roundoff. Another advantage of this method is the fact that we can allow eigenvalues to be on the imaginary axis (if they appear with even multiplicity), which none of the other methods can.

By analogy with the use of an exact shift in the QR algorithm to isolate a one-dimensional invariant subspace, Algorithm 2.3 can be considered to be a multishift method in the sense that the computed eigenvalues are simultaneously used to isolate the desired invariant subspace. In exact arithmetic, Algorithm 2.3 will transform the Hamiltonian matrix to block upper triangular form to obtain the desired Lagrangian invariant subspace. In practice, however, the (2,1) block of the transformed Hamiltonian matrix can be far from zero, and hence the computed subspace is only an approximation to the required Lagrangian subspace. One reason for this inaccuracy is the fact that the eigenvalues are usually not exact. A second difficulty, in particular in the presence of multiple eigenvalues, is the decision of when to perform a deflation, which is critical for the speed and accuracy of this method. Moreover, roundoff errors can create difficulties even when one is working with a single exact shift, as observed and analyzed by Parlett [21]. Nevertheless, these difficulties can often be surmounted with defect correction techniques, which are described in the next section. In fact, we can use Algorithm 2.3 iteratively as a defect correction method.

There are many variations how this method can be implemented. A detailed analysis of different implementation issues is given in [3]. In all considered cases it was observed that the computed solution is often not satisfactory unless the method is combined with iterative refinement. We will discuss this issue in the next section.

3. Defect correction.

Any numerical solution $\tilde{X}$ of the algebraic Riccati equation

$$0 = G + A^T X + XA - XRX, \tag{17}$$

is usually corrupted from roundoff and other errors. Thus it is in most cases advisable to use iterative refinement to improve the accuracy. A general refinement algorithm for (17) was proposed in [19].

THEOREM 3.1. *Let $X = X^T$ be the positive semidefinite solution of* (17) *and let $\tilde{X}$ be a symmetric approximation to X. Let $P = X - \tilde{X}$, $\tilde{A} = A - G\tilde{X}$ and*

$$\tilde{G} = G + A^T \tilde{X} + \tilde{X}A - \tilde{X}R\tilde{X} \tag{18}$$

the residual when inserting $\tilde{X}$, then P is the stabilizing solution of the Riccati equation

$$0 = \tilde{G} + P\tilde{A} + \tilde{A}^T P - PRP. \tag{19}$$

Proof. See [19, 18]. □

This idea leads to the following defect correction method:

ALGORITHM 3.2 (DCORC).

Input: $A, R, G \in \mathbf{R}^{n,n}$ from (17) and a tolerance $\varepsilon > 0$ for the defect.

Output: An approximation $\tilde{X}$ to the positive semidefinite solution X of (17) and an error estimate $P \in \mathbf{R}^{n,n}$ with $\|P\| \leq \varepsilon$.

Step 1: Compute with any method a stabilizing approximate solution of (17).

Step 2: Set $P := \tilde{X}$, $\tilde{X} := 0$.

Step 3:

WHILE $\|P\| > \varepsilon$

Set

$$\begin{aligned} \tilde{X} &:= \tilde{X} + P \\ \tilde{G} &:= G + A^T \tilde{X} + \tilde{X}A - \tilde{X}R\tilde{X} \\ \tilde{A} &:= A - R\tilde{X} \end{aligned}$$

Compute with any method a stabilizing approximate solution of

$$0 = \tilde{G} + \tilde{A}^T P + P\tilde{A} - PRP \tag{20}$$

END WHILE

END.

Different methods can be used in Step 1 and 3 of this method. In particular, Newton's method (see, e.g., [18]) is used effectively in Step 3, but also the multishift method can be used again in Step 3, since the eigenvalues have not changed for the Hamiltonian matrices. The difficulty with the defect correction method is that the computed residual may be corrupted either from subtractive cancellation or from the fact that the linear system that has to be solved to obtain the solution X is badly conditioned. A particularly bad example for this effect was given by Laub [15]. We will discuss it in Section 4. The cancellation can partially be avoided by computing the residual with higher precision. Another way to avoid this problem in intermediate steps of the defect correction method is the following. Let

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \tag{21}$$

be an orthogonal symplectic such that the span of the first n columns of Q is approximately the stable invariant subspace of the Hamiltonian matrix

$$H = \begin{bmatrix} A & R \\ G & -A^T \end{bmatrix} \tag{22}$$

and let

$$\tilde{X} = -Q_{21}Q_{11}^{-1} \tag{23}$$

be the corresponding approximate solution of (1). Then the residual given in (18) has the form

$$\tilde{G} = G - A^T Q_{21}Q_{11}^{-1} - Q_{11}^{-T}Q_{21}^T A - Q_{11}^{-T}Q_{21}^T R Q_{21}Q_{11}^{-1} \tag{24}$$

Multiplying this equation from the left with Q_{11}^T and from the right with Q_{11}, we obtain the lower left block of the transformed Hamiltonian matrix

$$\hat{H} := Q^T H Q = \begin{bmatrix} \hat{A} & \hat{R} \\ \hat{G} & -\hat{A}^T \end{bmatrix} \tag{25}$$

as

$$\hat{G} = Q_{11}^T \tilde{G} Q_{11} = Q_{11}^T G Q_{11} - Q_{11}^T A Q_{21} - Q_{21}^T A^T Q_{11} - Q_{21}^T R Q_{21}. \tag{26}$$

Now $\hat{G}$ is available from the transformation to triangular-like form and it does not involve the inversion of Q_{11}. Hence it is conceivable that we will obtain better results if we work with a defect correction on the triangular-like form, with residual $\hat{G}$, rather than on the Riccati equation, and postpone the solution of the linear system (which may be ill conditioned) until the defect correction has converged. In combination with the multishift algorithm we then have the following **O**rthogonal **S**ymplectic **M**ultishift method for the solution of the **A**lgebraic **R**iccati **E**quation.

Algorithm 3.3 (OSMARE).

Input: Hamiltonian matrix $H = \begin{bmatrix} A & R \\ G & -A^T \end{bmatrix}$.

Output: Approximate solution X of the algebraic Riccati equation

$$0 = G + A^T X + XA - XRX.$$

Step 1: Compute the eigenvalues of H with Algorithm 2.2 *SQRED*.

ETNA
Kent State University
etna@mcs.kent.edu

Step 2: Set

$$H_0 := H =: \begin{bmatrix} A_0 & R_0 \\ G_0 & -A_0^T \end{bmatrix}, \quad Q := I_{2n}. \tag{27}$$

FOR $i = 1, 2, \ldots$ UNTIL $G_i \approx 0$

Compute with Algorithm 2.3 *LISH* Q_i, H_i, such that

$$H_i = Q_i^T H_{i-1} Q_i = \begin{bmatrix} A_i & R_i \\ G_i & -A_i^T \end{bmatrix}, \tag{28}$$

and set $Q := QQ_i$.

END FOR

Step 3. Let $Q =: \begin{bmatrix} Q_1 & -Q_2 \\ Q_2 & Q_1 \end{bmatrix}$. Solve the linear matrix equation

$$XQ_1 = -Q_2 \tag{29}$$

for example via the QR decomposition or Gaussian elimination with pivoting.

END.

If a subspace has deflated in Algorithm LISH then the iteration in the second step only operates on the subproblem. The cost for this algorithm depends strongly on the number of iteration steps in LISH and the number of defect correction steps. In practice we have observed an increased number of deflation steps with an increase of the matrix dimension. The major difficulty arises from the fact that we do not have a suitable deflation criterion in the deflation procedure of LISH. If we use the usual deflation criterion used in the QR algorithm

$$|\hat{h}_{k+1,k}| < c\mathbf{u}(|\hat{h}_{k,k}| + |\hat{h}_{k+1,k+1}|) \tag{30}$$

where c is a constant and $\mathbf{u}$ is the machine precision, then often a deflation was not recognized. Based on the error estimates for the eigenvalues computed by Algorithm 2.2 [25], we therefore use the less stringent deflation criterion

$$|\hat{h}_{k+1,k}| < c\sqrt{\mathbf{u}}(|\hat{h}_{k,k}| + |\hat{h}_{k+1,k+1}|) \tag{31}$$

but this may lead to inaccurate subspaces.

4. Numerical examples. In this section we describe the results of our comparison. All the described methods were implemented in MATLAB as well as in FORTRAN 77 according to the implementation standards described in [17] using LINPACK [11] and EISPACK [23] subroutines and some routines provided by R. Byers [7]. The codes are documented in [3] and are available from the authors.

The described algorithms were extensively tested and compared with other methods for the algebraic Riccati equation. In this section we describe the results of this comparison. All computations were performed on a Silicon Graphics IRIS Indigo R3000 ($\mathbf{u} \approx 2.22 \cdot 10^{-16}$) at the Institut für Geometrie und Praktische Mathematik in Aachen. Codes for the generation of the test examples are also available from the authors. The implementation of OSMARE is based on elimination via Givens rotations rather than Householder transformations. The reason is that after one iteration step of LISH the elements of the first column vector in (9) become very small in magnitude, and Givens rotations performed much better.

EXAMPLE 4.1. [2]

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}, \quad R = \begin{bmatrix} \epsilon^2 & 0 \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{32}$$

The exact solution of the Riccati equation is

$$X = \begin{bmatrix} \frac{1+\sqrt{1+\epsilon^2}}{\epsilon^2} & \frac{1}{2+\sqrt{1+\epsilon^2}} \\ \frac{1}{2+\sqrt{1+\epsilon^2}} & \frac{1}{4}\left(1 - \frac{\epsilon^2}{\left(2+\sqrt{1+\epsilon^2}\right)^2}\right) \end{bmatrix}. \tag{33}$$

For $\epsilon \rightarrow 0$ the pair (A, G) becomes unstabilizable and the solution X satisfies

$$\lim_{\epsilon \rightarrow 0} x_{11} = \infty.$$

The relative errors in x_{11}, x_{12}, x_{22} in OSMARE are given in the following table. In parentheses we give the number of implicit defect correction steps.

ϵ		rel. error	
1	x_{11}	0.0	
	x_{12}	$3.8 \cdot 10^{-16}$	(0)
	x_{22}	$1.2 \cdot 10^{-16}$	
10^{-2}	x_{11}	$6.6 \cdot 10^{-13}$	
	x_{12}	$1.6 \cdot 10^{-12}$	(1)
	x_{22}	$4.4 \cdot 10^{-16}$	
10^{-4}	x_{11}	$1.2 \cdot 10^{-9}$	
	x_{12}	$7.7 \cdot 10^{-9}$	(1)
	x_{22}	$2.2 \cdot 10^{-16}$	
10^{-6}	x_{11}	$3.0 \cdot 10^{-5}$	
	x_{12}	$1.9 \cdot 10^{-4}$	(1)
	x_{22}	0.0	

The results are in accordance with the condition estimate for the inversion of Q_1, which is $O(1/\epsilon^2)$.

EXAMPLE 4.2. [2]

$$A = \begin{bmatrix} -\epsilon & 1 & 0 & 0 \\ -1 & -\epsilon & 0 & 0 \\ 0 & 0 & \epsilon & 1 \\ 0 & 0 & -1 & \epsilon \end{bmatrix}, \quad G = R = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}. \tag{34}$$

For $\epsilon \rightarrow 0$ a pair of complex conjugate eigenvalues of the Hamiltonian matrix approaches the imaginary axis and the system approaches one which is unstabilizable. The computed eigenvalues in SQRED have real part 0 with respect to the machine precision $\mathbf{u}$ for $\epsilon = 10^{-7}$. We do not know the analytic solution in this case and give therefore the Frobenius norm of the residual. In parentheses we again give the number of implicit defect correction steps.

ϵ	$\|residual\|_F$
1	$2.8 \cdot 10^{-14}$ (1)
10^{-1}	$2.0 \cdot 10^{-15}$ (1)
10^{-2}	$1.4 \cdot 10^{-15}$ (1)
10^{-3}	$1.2 \cdot 10^{-15}$ (1)
10^{-4}	$5.7 \cdot 10^{-14}$ (0)
10^{-5}	$4.1 \cdot 10^{-14}$ (1)
10^{-6}	$6.5 \cdot 10^{-14}$ (3)
10^{-7}	$3.5 \cdot 10^{-15}$ (8)

In both first examples the Schur vector method gave analogous results. The Sign function method gave much worse results in the second example due to eigenvalues approaching the imaginary axis. Observe that the multishift method can deal with eigenvalues on the imaginary axis provided a Lagrangian subspace exists, while the sign function method and Newton's method do not work on such problems and the Schur vector method often has difficulties in determining the correct invariant subspace.

EXAMPLE 4.3. [2]

$$A = \begin{bmatrix} -0.1 & 0 \\ 0 & -0.02 \end{bmatrix}, \quad G = \begin{bmatrix} 100 & 1000 \\ 1000 & 10000 \end{bmatrix},$$
$$R = \begin{bmatrix} 0.1 & 0 \\ 0.001 & 0.01 \end{bmatrix} \begin{bmatrix} 1+\epsilon & 1 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.1 & 0 \\ 0.001 & 0.01 \end{bmatrix}^T.$$

For $\epsilon \to 0$ the elements and the condition of R become increasingly large, and the elements of the solution tend to zero. Again we give the Frobenius norm of the computed residuals

ϵ	$\|residual\|_F$
1	$1.8 \cdot 10^{-12}$ (0)
10^{-1}	$1.1 \cdot 10^{-11}$ (0)
10^{-2}	$1.7 \cdot 10^{-10}$ (0)
10^{-3}	$7.6 \cdot 10^{-8}$ (0)
10^{-4}	$8.3 \cdot 10^{-7}$ (0)
10^{-5}	$4.2 \cdot 10^{-7}$ (1)
10^{-6}	$5.3 \cdot 10^{-7}$ (1)
10^{-7}	$1.1 \cdot 10^{-3}$ (0)

Here the Schur vector method achieved slightly more accurate results but one or two more steps of explicit defect correction with Algorithm 3.2 DCORC improved also the results of OSMARE.

EXAMPLE 4.4. (Laub [15], Example 4)

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 & \dots & & 0 \\ 0 & A_{22} & A_{23} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & 0 & A_{N-2,N-2} & A_{N-2,N-1} & 0 \\ & & & 0 & A_{N-1,N-1} & \begin{matrix} 0 \\ -1 \end{matrix} \\ 0 & \dots & & & 0 \quad 0 & -1 \end{bmatrix} \in \mathbf{R}^{2N-1,2N-1}$$

$$
\begin{aligned}
G &= \mathrm{diag}(1,0,1,0,\ldots,1,0,1)\\
F &= \mathrm{diag}(0,10,0,10,\ldots,0,10,0),
\end{aligned}
$$

where

$$
\begin{aligned}
A_{k,k} &= \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix}, \quad 1 \le k < N,\\
A_{k+1,k} &= \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}, \quad 1 \le k < N-1.
\end{aligned}
$$

Again we give the Frobenius norm of the residual.

n	$\|residual\|_F$	
9	$7.5 \cdot 10^{-14}$	(1)
19	$2.6 \cdot 10^{-13}$	(1)
29	$5.4 \cdot 10^{-13}$	(3)
39	$2.0 \cdot 10^{-12}$	(7)
49	$7.4 \cdot 10^{-12}$	(11)

For large dimensions the number of iterations for OSMARE increases which has the effect that here the Schur vector method and the sign function method are faster for obtaining the same accuracy.

In this case one step of explicit defect correction with Algorithm 3.2 DCORC improved the residuals to $O(10^{-14})$.

EXAMPLE 4.5. (Laub [15], Example 5) The system has the form

$$
A = \begin{bmatrix}
-2 & 1 & 0 & & \ldots & 0 & 1 \\
1 & -2 & 1 & 0 & \ldots & & 0 \\
0 & 1 & -2 & 1 & 0 & \ldots & 0 \\
\vdots & & \ddots & \ddots & \ddots & & \vdots \\
0 & & & & & & 1 \\
1 & 0 & \ldots & & 0 & 1 & -2
\end{bmatrix}, \quad G = R = I_n. \tag{35}
$$

Most eigenvalues of the Hamiltonian matrix have multiplicity 2, hence as expected there will be a number of deflation steps in LISH. Here the abovementioned difficulty with choosing the right deflation criterion occurred and only the relaxed criterion (31) secured convergence.

n	$\|residual\|_F$	
5	$2.1 \cdot 10^{-15}$	(1)
10	$3.4 \cdot 10^{-15}$	(1)
20	$9.6 \cdot 10^{-15}$	(4)
30	$1.8 \cdot 10^{-14}$	(6)
64	$8.7 \cdot 10^{-14}$	(15)

The accuracy is for the given sizes equal to the accuracy obtained from the Schur vector method while for much larger n the accuracy is smaller than that for the Schur vector method due to the inaccurate subspaces obtained from the relaxed deflation criterion.

EXAMPLE 4.6. (Laub [15], Example 6)

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & & & 0 & 1 \\ 0 & \dots & & 0 & 0 \end{bmatrix},$$
$$G = \mathrm{diag}(0,\dots,0,1), \quad F = \mathrm{diag}(1,0,\dots,0).$$

The eigenvalues of the Hamiltonian matrix are the roots of $\lambda^{2n} = (-1)^{n+1}$. It is known [15] that $x_{1n} = 1$. The difficulty in this example lies in the fact that the linear system $XQ_1 = -Q_2$ is extremely ill conditioned and the elements of X become very large. The residual therefore gives no information on the accuracy of the solution. Here we use therefore the accuracy of the computed element x_{1n}.

The condition estimate from the LINPACK procedure DGECO for Q_1 is over 10^9 for $n = 21$. Using $\delta_{rel} = |x_{1n} - 1|$ we obtained the following results

	without def. corr.	with def. corr.
$\|residual\|_F$	$3.0 \cdot 10^{-4}$	244 (1)
δ_{rel}	$2.4 \cdot 10^{-15}$	$1.4 \cdot 10^{-7}$

The result computed without any defect correction is several digits more accurate than the results obtained from the Schur vector and sign function methods.

The decrease in accuracy after defect correction arises from the fact that the solution matrix X has elements of very large magnitude. Even in the implicit defect correction the residual based on (26) is very inaccurate and used in the next step. This shows that it is important to monitor the condition numbers and sizes of the elements of X in order to avoid the situation that the computation of the residual corrupts all further refinement steps. An error analysis of the algorithm is needed in order to better understand the situations where defect correction does not lead to improved solutions.

EXAMPLE 4.7. We also generated random Hamiltonian matrices to compare the different methods. These matrices had the following properties: Let Re_{min}, Re_{max} be the smallest and largest modulus of the real part of an eigenvalue of H

	example no.					
	1	2	3	4	5	6
n	10	10	20	20	30	40
$\|\|H\|\|_F$	39.9	325.6	147.3	96.3	330.3	577.0
Re_{max}	26.44	216.14	102.3	9.309	229.51	403.21
Re_{min}	0.59	0.803	0.367	0.051	0.835	1.098

For the residuals we obtained the following Frobenius norms:

	example no.					
	1	2	3	4	5	6
OSMARE	$7 \cdot 10^{-14}$	$3 \cdot 10^{-13}$	$2 \cdot 10^{-13}$	$3 \cdot 10^{-12}$	$7 \cdot 10^{-13}$	$2 \cdot 10^{-12}$
SIGNF	$1 \cdot 10^{-13}$	$5 \cdot 10^{-13}$	$2 \cdot 10^{-13}$	$3 \cdot 10^{-13}$	$9 \cdot 10^{-13}$	$2 \cdot 10^{-12}$
SCHUR	$1 \cdot 10^{-13}$	$5 \cdot 10^{-13}$	$2 \cdot 10^{-13}$	$4 \cdot 10^{-12}$	$8 \cdot 10^{-13}$	$2 \cdot 10^{-12}$

The results did not differ from those for the sign function and Schur vector method also in all other random test cases, which had dimensions up to $n = 50$.

5. Conclusion. We have outlined the multishift technique first proposed in [1] for the computation of any Lagrangian invariant subspace of a Hamiltonian matrix, and considered its use in solving algebraic matrix Riccati equations that arise in linear optimal control. We have seen that the algorithm can be used iteratively, as a defect correction procedure, to accurately compute solutions of Riccati equations. The procedure has the desirable property that it isolates the required invariant subspace by performing orthogonal symplectic similarity transformations on the initial Hamiltonian matrix. Further refinements and analysis of the algorithm are currently under investigation.

REFERENCES

[1] G.S. Ammar and V. Mehrmann, *On Hamiltonian and symplectic Hessenberg forms*, Lin. Alg. Appl. 149 (1991), pp. 55–72.

[2] W.F. Arnold, III and A.J. Laub, *Generalized eigenproblem algorithms and software for algebraic Riccati equations*, Proc. IEEE 72 (1984), pp. 1746–1754.

[3] P. Benner, *Ein orthogonal symplektischer Multishift Algorithmus zur Lösung der algebraischen Riccatigleichung* , Diplomarbeit RWTH Aachen, Institut f. Geometrie und Praktische Mathematik (March 1993).

[4] A. Bunse–Gerstner, R. Byers and V. Mehrmann, *Numerical methods for algebraic Riccati equations*, Lecture Notes of the Workshop on "The Riccati Equation in Control, Systems and Signal", S. Bittanti (Ed.), Pitagora Editrice Bologna (1989), pp. 107–115.

[5] A. Bunse–Gerstner and V. Mehrmann, *A symplectic QR-like algorithm for the solution of the real algebraic Riccati equation*, IEEE Trans. Autom. Control AC-31, (1986), pp. 1104–1113.

[6] A. Bunse–Gerstner and V. Mehrmann, *The HHDR algorithm and its application to optimal control problems* , R.A.I.R.O. Automatique, 23 (1989), pp. 305–329.

[7] R. Byers, *Hamiltonian and symplectic algorithms for the algebraic Riccati equation*, Ph.D. Thesis, Cornell University (January, 1983).

[8] R. Byers, *A Hamiltonian QR Algorithm*, SIAM J. Sci. Stat. Comp. 7 (1986), pp. 212–229.

[9] R. Byers, *Solving the algebraic Riccati equation with the matrix sign function*, Lin. Alg. Appl. 85 (1987), pp. 267–279.

[10] J.L. Casti, *Linear Dynamical Systems*, Mathematics in Science and Engineering, Academic Press (1987).

[11] J.J. Dongarra, J.R. Bunch, C. Moler and G.W. Stewart, *Linpack User's Guide*, SIAM Philadelphia (1979).

[12] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2nd Ed., The Johns Hopkins University Press, Baltimore, Md (1989).

[13] R. Kleinman, *On an iterative technique for Riccati equation computations*, IEEE Trans. Autom. Control AC–13 (1968), pp. 114–115.

[14] H.W. Knobloch and H. Kwakernaak, *Lineare Kontrolltheorie*, Springer (1985).

[15] A.J. Laub, *A Schur method for solving algebraic Riccati equations*, IEEE Trans. Automatic Control AC–24 (1979), pp. 912–921.

[16] W.W. Lin, *On Schur type decompositions for Hamiltonian and symplectic pencils*, Preprint Institute of Applied Mathematics, National Tsing Hua University, Taiwan (1990).

[17] NAG, *Implementation and Documentation Standards for the Subroutine Library in Control and Systems Theory SLICOT*, Numerical Algorithm Group Publication NP2032, Eindhoven/Oxford (1990).

[18] V. Mehrmann, *The Autonomous Linear Quadratic Control Problem: Theory and Numerical Solution*, Lecture Notes in Control and Information Sciences 163, Springer-Verlag, Berlin (1991).

[19] V. Mehrmann and E. Tan, *Defect correction methods for the solution of algebraic Riccati equations*, IEEE Trans. Automatic Control AC–33 (1988), pp. 695–698.

[20] C. Paige and C.F. Van Loan, *A Schur decomposition for Hamiltonian matrices*, Lin. Alg. Appl. 41 (1981), pp. 11–32.

[21] B.N. Parlett and J. Le, *Forward instability of tridiagonal QR*, SIAM J. Matrix Anal. Appl. 14, No. 1 (1993), pp. 279–316.

[22] A.C. Raines and D.S. Watkins, *A class of Hamiltonian–symplectic methods for solving the algebraic Riccati equation*, technical report, Washington State University, Pullman, Wa

(1992).

[23] B.T. Smith, J.M. Boyle, J.J. Dongarra, B.S. Garbow, Y. Ikebe, V.C. Klema and C.B. Moler, *Matrix Eigensystem Routines–EISPACK Guide*, Lecture Notes in Computer Science 6, Springer, Berlin (1976).

[24] The Mathworks, Inc., *386–Matlab User's Guide*, (1990).

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

[26] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford (1965).