Technische Universität Chemnitz-Zwickau

DFG-Forschergruppe "SPC" · Fakultät für Mathematik

Peter Benner Alan J. Laub Volker Mehrmann

A Collection of Benchmark Examples for the Numerical Solution of Algebraic Riccati Equations II: Discrete-Time Case

Preprint-Reihe der Chemnitzer DFG-Forschergruppe
"Scientific Parallel Computing"

SPC 95_23 **Dezember 1995**

A Collection of Benchmark Examples for the Numerical Solution of Algebraic Riccati Equations II: Discrete-Time Case

Peter Benner* Alan J. Laub† Volker Mehrmann*

Abstract

This is the second part of a collection of benchmark examples for the numerical solution of algebraic Riccati equations. After presenting examples for the continuous-time case in Part I, our concern in this paper is discrete-time algebraic Riccati equations. This collection may serve for testing purposes in the construction of new numerical methods, but may also be used as a reference set for the comparison of methods.

1 Introduction

We present a collection of examples for discrete-time algebraic Riccati equations (DARE) of the form

$$0 = A^TXA - X - (A^TXB + S)(R + B^TXB)^{-1}(B^TXA + S^T) + Q \tag{1}$$

where $A, Q, X \in \mathbb{R}^{n\times n}$, $B, S \in \mathbb{R}^{n\times m}$, and $R = R^T \in \mathbb{R}^{m\times m}$. The matrix $Q = Q^T$ may be given in factored form $Q = C^T\tilde{Q}C$ with $C \in \mathbb{R}^{p\times n}$ and $\tilde{Q} = \tilde{Q}^T \in \mathbb{R}^{p\times p}$.
As it will be described below, (1) can be solved using its relationship to the symplectic pencil defined by

$$L - \lambda M = \begin{bmatrix} \hat{A} & 0 \\ -\hat{Q} & I_n \end{bmatrix} - \lambda \begin{bmatrix} I_n & G \\ 0 & \hat{A}^T \end{bmatrix} \tag{2}$$

where

$$\begin{aligned} \hat{A} &= A - BR^{-1}S^T, \\ G &= BR^{-1}B^T, \\ \hat{Q} &= Q - SR^{-1}S^T = C^T\tilde{Q}C - SR^{-1}S^T. \end{aligned}$$

*Fakultät für Mathematik, Technische Universität Chemnitz–Zwickau, 09107 Chemnitz, FRG. e-mail: `benner@mathematik.tu-chemnitz.de`, `mehrmann@mathematik.tu-chemnitz.de`. These authors have been supported by *Deutsche Forschungsgemeinschaft*, research grant *Me 790/7-1 Singuläre Steuerungsprobleme*.

†Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, e-mail: `laub@ece.ucsb.edu`. This author has been supported by *National Science Foundation Grant ECS-9120643* and *Air Force Office of Scientific Research Grant F49620-94-1-0104DEF*.

If $\hat{A}$ is invertible, this pencil is equivalent to the symplectic matrices

$$\begin{aligned} Z &= M^{-1}L = \begin{bmatrix} \hat{A} + G\hat{A}^{-T}\hat{Q} & -G\hat{A}^{-T} \\ -\hat{A}^{-T}\hat{Q} & \hat{A}^{-T} \end{bmatrix}, \quad (3) \\ \text{or} \qquad \tilde{Z} &= LM^{-1} = \begin{bmatrix} \hat{A} & -\hat{A}G\hat{A}^{-T} \\ -\hat{Q} & \hat{Q}G\hat{A}^{-T} + \hat{A}^{-T} \end{bmatrix}. \quad (4) \end{aligned}$$

The DARE (1) arises, e.g., in (a) stochastic realization problems, and (b) linear-quadratic control problems. In case (a), R is the measurement noise covariance and it is not uncommon for this kind of matrix to be singular. For (b), R is the control weighting matrix and in the discrete-time case, occasionally such a matrix can be singular, too. In these cases, the pencil formulation (2) is not possible. An *extended symplectic pencil* (ESP) can then be formed by

$$\tilde{L} - \lambda\tilde{M} = \begin{bmatrix} A & 0 & B \\ Q & -I_n & S \\ S^T & 0 & R \end{bmatrix} - \lambda \begin{bmatrix} I_n & 0 & 0 \\ 0 & -A^T & 0 \\ 0 & -B^T & 0 \end{bmatrix}. \quad (5)$$

To illustrate a problem where the DARE (1) arises, we consider the discrete-time linear-quadratic control problem (case (b) from above).

Minimize

$$J(x_0, u) = \frac{1}{2}\sum_{k=0}^{\infty}\left(y_k^T\tilde{Q}y_k + 2x_k^TSu_k + u_k^TRu_k\right)\,dt \quad (6)$$

subject to the difference equation

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \quad k = 0, 1, \ldots, \qquad x_0 = \xi, \quad (7) \\ y_k &= Cx_k, \quad k = 0, 1, \ldots. \quad (8) \end{aligned}$$

If, for example, $\tilde{Q} \geq 0$, $R > 0$, (A, B) stabilizable[1], and (A, C) detectable[2], then the solution of the optimal control problem (6)–(8) is given by the feedback law

$$u_k = -(R + B^TXB)^{-1}(A^TXB + S)^Tx_k, \quad k = 0, 1, \ldots,$$

where X is the unique stabilizing, positive semidefinite solution of (1) (see, e.g., [21, 28]). One common approach to solve (1) is to compute the stable invariant subspace of the symplectic matrix Z or the stable deflating subspace of the (extended) symplectic pencil given above, i.e., the invariant/deflating subspace corresponding to the generalized eigenvalues of $L - \lambda M$, $\tilde{L} - \lambda\tilde{M}$, respectively, inside the unit circle (e.g., [18, 21, 24]). If this subspace is spanned by

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \begin{matrix} \}n \\ \}n \end{matrix} \qquad \text{or} \qquad \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} \begin{matrix} \}n \\ \}n \\ \}m \end{matrix}, \qquad \text{respectively,}$$

and U_1 is invertible, then $X = U_2U_1^{-1}$ is the stabilizing solution of (1), i.e., all the eigenvalues of

$$F = A - B(R + B^TXB)^{-1}(A^TXB + S)^T \quad (9)$$

lie inside the unit circle.

[1] (A, B) is *(d-)stabilizable*, if rank $[A - \lambda I,\ B] = n$ for all λ with $|\lambda| \geq 1$.
[2] (A,C) is *(d-)detectable*, if (A^T, C^T) is (d-)stabilizable.

At this point it should be noted that it is possible to transform a continuous-time algebraic Riccati equation (CARE) into a DARE (and vice versa) via a (generalized) Cayley transformation, i.e., the Hamiltonian matrix corresponding to the CARE is transformed into a symplectic matrix/pencil. From this symplectic pencil it is possible to derive the coefficient matrices of a corresponding DARE under certain regularity assumptions; see [22]. In this way, it is possible to obtain DARE examples from the first part of our benchmark collection. We do not use this approach here, though, and restrict ourselves to data arising naturally in a discrete-time setting and/or highlighting some of the properties of discrete-time algebraic Riccati equations.

In the sequel we will use the following notation. Let $A \in \mathbb{R}^{n\times n}$. By $\sigma(A)$ we denote the set of eigenvalues or spectrum of A. The spectral norm of a matrix is given by

$$\|A\| = \sqrt{\max\{|\lambda| \,:\, \lambda \in \sigma(A^TA)\}}$$

and the given matrix condition numbers are based upon the spectral norm,

$$\kappa(A) = \|A\|\|A^{-1}\|.$$

The *Frobenius* norm of a matrix will be denoted by $\|A\|_F$ and is given by

$$\|A\|_F = \sqrt{\sum_{i,j=1}^{n} a_{ij}^2}.$$

All norms and condition numbers given in the sequel were computed by the MATLAB[3] functions `norm` and `cond`.

The examples are grouped in three sections. The first section contains parameter-free examples of fixed dimension while the second has parameter-dependent problems of fixed size. Section 4 contains examples of scalable size.
The coefficient matrices of the presented examples are usually given in the same form as they appear in the literature. Since in most cases $S = 0$, we omit S in all examples where this property holds.
All presented examples may be generated by the FORTRAN 77 subroutine DAREX (see Appendix A) and the MATLAB function `darex` (see Appendix B). Appendix C describes how to obtain the software.
The description of each example contains a table with some of the system properties. This information is summarized in Appendix D. For all parameters needed in the examples there exist default values that are also given in the tables. These default values are chosen in such a way that the collection of examples can be used as a testset for the comparison of methods. The tables contain information for one or more choices of the parameters. Underlined values indicate the default values.
For each example, we provide the condition number $\kappa(\hat{A})$ which shows if the symplectic matrix Z in (3) can be formed safely (though it is still favorable to use the pencil approach and thereby avoiding a matrix inversion unless $\hat{A}^{-1}$ is "easy" to form). The column $|\lambda^C_{max}|$ indicates the absolute value of the closed-loop eigenvalue of largest modulus, i.e.,

$$|\lambda^C_{max}| = \max\{\,|\lambda| \,:\, \lambda \in \sigma(F)\,\}$$

[3] MATLAB is a trademark of The MathWorks, Inc.

with F as in (9). These are the (generalized) eigenvalues of the symplectic matrix (pencils) in (2), (3), and (5) inside the unit circle. Further, we give norms and condition numbers of the stabilizing solution X. For examples without analytical solution available, we computed approximations by the generalized Schur vector method [3, 24]. If possible, these approximations were refined by Newton's method [3, 12, 21] to achieve the highest possible accuracy. We then chose the approximate solution with smallest residual norm and recomputed the solution using the optimal scaling strategy proposed in [11]. This computed solution was then used to determine the properties of the example.
The "right" condition number of algebraic Riccati equations is still an open problem although the problem has been attacked by several papers during recent years, see, e.g., [11, 15, 26, 30]. For simplicity, here we use the condition number proposed in [11]. This condition number measures the sensitivity of the stabilizing DARE solution with respect to first-order perturbations by means of the Fréchet derivative of the DARE at X. In [11] it is shown that, assuming $Q \geq 0$, $R > 0$, the so defined condition number is given by

$$K_{DARE} = \frac{\| [\, Z_1,\, Z_2,\, Z_3 \,] \|}{\|X\|_F},$$

where

$$\begin{aligned} Z_1 &= \|A\|_F P^{-1} \left(I_n \otimes F^T X + (F^T X \otimes I_n) T \right), & (10) \\ Z_2 &= -\|G\|_F P^{-1} \left(\hat{A}^T X (I_n + GX)^{-1} \otimes \hat{A}^T X (I_n + GX)^{-1} \right), & (11) \\ Z_3 &= \|Q\|_F P^{-1}. & (12) \end{aligned}$$

Here, denoting the jth unit vector by e_j, the permutation matrix T is defined by

$$T = \sum_{i,j=1}^{n} e_i e_j^T \otimes e_j e_i^T,$$

and P is the matrix representation of the *Stein* (*discrete Lyapunov*) operator

$$\Omega(Z) = Z - F^T Z F.$$

The computation of K_{DARE} therefore requires the solution of the linear equations (10)–(12). Since Kronecker products are involved, these systems get very large even for small numbers of n. For larger n, an inverse power iteration can be employed to estimate $\| [\, Z_1,\, Z_2,\, Z_3 \,] \|_F$ (see [11]). This approach requires in each iteration step the solution of two Stein equations corresponding to Ω.

2 Parameter-free problems of fixed size

Example 1 [17, Example 2]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	1	2	–	114.99	0.50	21.03	∞	18.85

This is an example of stabilizable-detectable, but uncontrollable-unobservable data. We have the following system matrices:

$$A = \begin{bmatrix} 4 & 3 \\ -\frac{9}{2} & -\frac{7}{2} \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \qquad R = 1, \qquad Q = \begin{bmatrix} 9 & 6 \\ 6 & 4 \end{bmatrix}$$

with stabilizing solution

$$X = \tfrac{1+\sqrt{5}}{2} \begin{bmatrix} 9 & 6 \\ 6 & 4 \end{bmatrix}$$

and closed-loop spectrum $\{ -1/2,\ (3-\sqrt{5})/3 \}$.

Example 2 [17, Example 3], [16, Example 6.15]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	2	2	–	1.05	0.69	5.07×10^{-2}	4.97	4.74

This example illustrates a linear-quadratic control problem as defined by (6)–(8). The coefficient matrices are

$$\begin{aligned} A &= \begin{bmatrix} 0.9512 & 0 \\ 0 & 0.9048 \end{bmatrix}, \qquad B = \begin{bmatrix} 4.877 & 4.877 \\ -1.1895 & 3.569 \end{bmatrix}, \\ R &= \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & 3 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0.005 & 0 \\ 0 & 0.02 \end{bmatrix}. \end{aligned}$$

In [16, 17], solution matrices are given. We omit reproducing them here since they are not derived analytically.

Example 3 [31, Example II]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	1	1	–	5.83	0.00	1.00	1.00	∞

This example was used in [31] to demonstrate a compression technique for the extended pencil (5). The data are given by

$$A = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R = 0.$$

If interpreted in terms of a linear system as in (7)–(8), Q can be written as

$$Q = C^T \tilde{Q} C, \qquad C = [\,0 \ \ 1\,], \qquad \tilde{Q} = 1.$$

The exact stabilizing solution is $X = I_2$, and the closed-loop spectrum is $\{ 0,\ 0 \}$. Due to the singularity of R, the condition number K_{DARE} is not defined here (represented by a value "∞" in the table). This example can be used, e.g., as a first test of any solver to deal with a singular weighting/measurement noise covariance matrix.

Example 4 [13]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	2	2	–	∞	0.69	126.99	2.84×10^3	∞

This is another example with a singular R–matrix. Furthermore, we have a nonzero S–matrix. The coefficients of (1) are given by

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} 9 & 3 \\ 3 & 1 \end{bmatrix}, \\ Q &= \frac{1}{11}\begin{bmatrix} -4 & -4 \\ -4 & 7 \end{bmatrix}, \qquad S = \begin{bmatrix} 3 & 1 \\ -1 & 7 \end{bmatrix}. \end{aligned}$$

Again, the DARE condition number K_{DARE} can not be computed due to the singular R.

Example 5 [14], [22, Example 2].

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	1	2	–	∞	0.38	5.19	114.13	1.88

This example shows one of the major differences between the properties of continuous-time algebraic Riccati equations and their discrete counterparts. Consider the DARE defined by

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad Q = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}, \qquad R = 1.$$

The spectrum of the pencil $L - \lambda M$ in (2) is $\{\,0, \infty, -(3 \pm \sqrt{5})/2\,\}$. The DARE has exactly two solutions,

$$X_1 = \begin{bmatrix} 1 & 2 \\ 2 & 2+\sqrt{5} \end{bmatrix}, \qquad X_2 = \begin{bmatrix} 1 & 2 \\ 2 & 2-\sqrt{5} \end{bmatrix}.$$

but neither of them is negative semidefinite. On the other hand, (A, B) is controllable. In the case of a continuous-time system, this property would assure the existence of a negative semidefinite solution. The stabilizing solution in the control-theoretic sense is the positive definite solution X_1.

Example 6 [1]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
4	2	4	–	1.01	0.94	35.36	3.34	30.58

The data of this example represent a simple control problem for a satellite. The system is given by equations describing the small angle altitude variations about the roll and yaw axes of a satellite in circular orbit. These equations originally form a second-order differential equation. A first-order realization of this model and sampling every 0.1 seconds yields the system matrices

$$A = \begin{bmatrix} 0.998 & 0.067 & 0 & 0 \\ -0.067 & 0.998 & 0 & 0 \\ 0 & 0 & 0.998 & 0.153 \\ 0 & 0 & -0.153 & 0.998 \end{bmatrix}, \qquad B = \begin{bmatrix} 0.0033 & 0.02 \\ 0.1 & -0.0007 \\ 0.04 & 0.0073 \\ -0.0028 & 0.1 \end{bmatrix}.$$

The weighting matrices used in the performance index $J(x_0, u)$ in (6) are given by

$$Q = \tilde{Q} = \begin{bmatrix} 1.87 & 0 & 0 & -0.244 \\ 0 & 0.744 & 0.205 & 0 \\ 0 & 0.205 & 0.589 & 0 \\ -0.244 & 0 & 0 & 1.048 \end{bmatrix}, \qquad R = I_2.$$

Example 7 [19]

n	m	p	parameter	$\kappa(A)$	$\|\lambda^C_{max}\|$	$\|X\|$	$\kappa(X)$	K_{DARE}
4	2	4	–	19.86	0.99	2.06	183.33	790.37

This is a simple example of a control system having slow and fast modes.

$$A = 10^{-3} \times \begin{bmatrix} 984.75 & -79.903 & 0.9054 & -1.0765 \\ 41.588 & 998.99 & -35.855 & 12.684 \\ -546.62 & 44.916 & -329.91 & 193.18 \\ 2662.4 & -100.45 & -924.55 & -263.25 \end{bmatrix},$$

$$B = 10^{-4} \times \begin{bmatrix} 37.112 & 7.361 \\ -870.51 & 0.093411 \\ -11984.0 & -4.1378 \\ -31927.0 & 9.2535 \end{bmatrix}, \qquad R = I_2, \qquad Q = 0.01 I_4.$$

One complex conjugate pair of the computed closed-loop eigenvalues is located on a circle with radius ≈ 0.99 around the origin, i.e., is relatively close to the unit circle. Requiring that this distance should not cause any problems for any DARE solver seems to be reasonable.

Example 8 [20, Example 4.3]

n	m	p	parameter	$\kappa(A)$	$\|\lambda^C_{max}\|$	$\|X\|$	$\kappa(X)$	K_{DARE}
4	4	4	–	378.60	$\approx 1 - 1.8 \times 10^{-5}$	65.77	6.18×10^{11}	5.12×10^4

Here, the coefficient matrices of (1) are constructed as follows. Given

$$A_0 = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 1 & 0.6 & 0 & 0 \\ 0 & 1 & 0.8 & 0 \\ 0 & 0 & 0 & -0.999982 \end{bmatrix}, \qquad Q_0 = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$V = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \iff V^{-1} = \begin{bmatrix} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

we obtain

$$A = V A_0 V^{-1} = \begin{bmatrix} -0.6 & -2.2 & -3.6 & -5.400018 \\ 1 & 0.6 & 0.8 & 3.399982 \\ 0 & 1 & 1.8 & 3.799982 \\ 0 & 0 & 0 & -0.999982 \end{bmatrix}, \qquad B = V I_4 = V,$$

$$Q = V^{-T} Q_0 V^{-1} = \begin{bmatrix} 2 & 1 & 3 & 6 \\ 1 & 2 & 2 & 5 \\ 3 & 2 & 6 & 11 \\ 6 & 5 & 11 & 22 \end{bmatrix}, \qquad R = I_4.$$

A factorization in the control-theoretic sense, $Q = C^T \tilde{Q} C$, is therefore given by $C := V^{-1}$ and $\tilde{Q} := Q_0$.
All the generalized eigenvalues of $L - \lambda M$ are real. The distance of the largest closed-loop eigenvalue to the unit circle is $\approx 1.8 \times 10^{-5}$. The problem is designed so that $\kappa(L+M) \approx 4 \times 10^{11}$. Due to this large condition number and the eigenvalues close to the unit circle, problems with the convergence of the iteration process are to be expected when the DARE is solved via a method based on the sign function iteration (e.g., the methods in [4, 9]). Note that K_{DARE} signals only a very mild ill-conditioning of the DARE.

Example 9 [8, Section 2.7.4]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
5	2	5	–	23.52	0.98	73.90	73.73	100.81

The fifth-order linearized state-space model of a chemical plant presented in [10, 29] is discretized by sampling every 0.5 seconds, yielding a discrete-time linear-quadratic control problem of the form (6)–(7) defined by

$$A = 10^{-4}\times\begin{bmatrix} 9540.70 & 196.43 & 35.97 & 6.73 & 1.90 \\ 4084.90 & 4131.70 & 1608.40 & 446.79 & 119.71 \\ 1221.70 & 2632.60 & 3614.90 & 1593.00 & 1238.30 \\ 411.18 & 1285.80 & 2720.90 & 2144.20 & 4097.60 \\ 13.05 & 58.08 & 187.50 & 361.62 & 9428.00 \end{bmatrix}, \quad B = 10^{-4}\times\begin{bmatrix} 4.34 & -1.22 \\ 266.06 & -104.53 \\ 375.30 & -551.00 \\ 360.76 & -660.00 \\ 46.17 & -91.48 \end{bmatrix}.$$

The weighting matrices in the cost functional (6) are chosen as identities, i.e., we have $Q = \tilde{Q} = I_5$ and $R = I_2$.

If we modify the optimal control problem (6)–(8) by allowing the observation to depend upon the control, we obtain the following problem:
Minimize

$$J(x_0,u) = \frac{1}{2}\sum_{k=0}^{\infty}\left(y_k^T\tilde{Q}y_k + u_k^T\tilde{R}u_k\right)\, dt \tag{13}$$

subject to the difference equation

$$x_{k+1} = Ax_k + Bu_k, \quad k = 0,1,\ldots, \qquad x_0 = \xi, \tag{14}$$

$$y_k = Cx_k + Du_k, \quad k = 0,1,\ldots, \tag{15}$$

then we can rewrite the cost functional (13) as

$$J(x_0,u) = \frac{1}{2}\sum_{k=0}^{\infty}\left(x^TQx + x^TSu + u^TS^Tx + u^TRu\right)\, dt \tag{16}$$

where $Q = C^T\tilde{Q}C$, $R = \tilde{R} + D^T\tilde{Q}D$, and $S = C^T\tilde{Q}D$.
The data of the following example come from such a problem.

Example 10 [7]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
6	2	2	–	∞	0.67	2.53	37.38	3.94

The matrices of the linear system (A, B, C, D) are given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}.$$

With $\tilde{Q} = \tilde{R} = I_2$, we obtain the following coefficient matrices for the DARE: A, B, C, $\tilde{Q}$ are defined above, and

$$Q = C^T \tilde{Q} C = \left[\begin{array}{cccccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right], \qquad R = \left[\begin{array}{cc} 3 & 0 \\ 0 & 1 \end{array}\right], \qquad S = \left[\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0 & 0 \end{array}\right].$$

The system properties are good-natured. The system can easily be transformed to a standard system as in (2). Therefore, this example is helpful for first verifications of any DARE solver based on the extended formulation given in (5) since the results can be compared to those obtained by any other solver based on the formulation by the symplectic pencil (2).

Example 11 [25]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
9	3	2	–	1.58×10^6	0.96	607.66	∞	74.23

This is the data for a 9th-order discrete state-space model of a tubular ammonia reactor. It should be noted that the underlying model includes a disturbance term which is neglected in this context.
The continuous state-space model of this problem was presented as Example 5 in the first part of the benchmark collection [5]. Sampling every 30 seconds yields the following system matrices for the discrete model:

$$A = 10^{-2} \times \left[\begin{array}{rrrrrrrrr} 87.01 & 13.50 & 1.159 & 0.05014 & -3.722 & 0.03484 & 0 & 0.4242 & 0.7249 \\ 7.655 & 89.74 & 1.272 & 0.05504 & -4.016 & 0.03743 & 0 & 0.4530 & 0.7499 \\ -12.72 & 35.75 & 81.70 & 0.1455 & -10.28 & 0.0987 & 0 & 1.185 & 1.872 \\ -36.35 & 63.39 & 7.491 & 79.66 & -27.35 & 0.2653 & 0 & 3.172 & 4.882 \\ -96.00 & 164.59 & -12.89 & -0.5597 & 7.142 & 0.7108 & 0 & 8.452 & 12.59 \\ -66.44 & 11.296 & -8.889 & -0.3854 & 8.447 & 1.36 & 0 & 14.43 & 10.16 \\ -41.02 & 69.30 & -5.471 & -0.2371 & 6.649 & 1.249 & 0.01063 & 9.997 & 6.967 \\ -17.99 & 30.17 & -2.393 & -0.1035 & 6.059 & 2.216 & 0 & 21.39 & 3.554 \\ -34.51 & 58.04 & -4.596 & -0.1989 & 10.56 & 1.986 & 0 & 21.91 & 21.52 \end{array}\right],$$

$$B^T = 10^{-4} \times \left[\begin{array}{rrrrrrrrr} 4.76 & 0.879 & 1.482 & 3.892 & 10.34 & 7.203 & 4.454 & 1.971 & 3.773 \\ -0.5701 & -4.773 & -13.12 & -35.13 & -92.75 & -61.59 & -36.83 & -15.54 & -30.28 \\ -83.68 & -2.73 & 8.876 & 24.80 & 66.80 & 38.34 & 20.29 & 6.937 & 14.69 \end{array}\right].$$

In the discrete model, only the first and fifth state variables are used as outputs, i.e.,

$$C = \left[\begin{array}{ccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

and the weighting matrices are chosen as $\tilde{Q} = 50 I_2$ and $R = I_3$.

3 Parameter-dependent problems of fixed size

Example 12

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
2	1	2	$\varepsilon = 100$	∞	0.00	1.00×10^4	1.00×10^4	2.65
			$\underline{\varepsilon = 10^6}$	∞	0.00	1.00×10^{12}	1.00×10^{12}	2.65

Here, the matrix A has a parameter and the coefficient matrices of the DARE (1) are

$$A = \begin{bmatrix} 0 & \varepsilon \\ 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad R = 1, \qquad Q = I_2.$$

The stabilizing solution is given by

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1+\varepsilon^2 \end{bmatrix}$$

and the closed-loop spectrum is $\{\,0,\,0\,\}$.
For $\varepsilon = 100$, this is Example 2 from [11]. As $\varepsilon \to \infty$, this becomes an example of a DARE which is badly scaled in the sense of [27] due to the fact that $\|A\|_F \gg \|G\|_F$, $\|Q\|_F$. Obviously, the norm (and condition) of the stabilizing solution X grow like ε^2 whereas the DARE condition number K_{DARE} remains constant.

Example 13 [27]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
3	3	3	$\varepsilon = 1$	∞	0.38	9.11	9.11	2.51
			$\underline{\varepsilon = 10^6}$	∞	0.38	9.11×10^6	9.11	2.51

This example is constructed as follows. Let

$$A_0 = \mathrm{diag}(0,\ 1,\ 3), \qquad V = I - \frac{2}{3}vv^T, \qquad v^T = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

Then

$$A = VA_0V, \qquad G = \frac{1}{\varepsilon}I_3, \qquad Q = \varepsilon I_3.$$

A factorization $Q = C^T\tilde{Q}C$ can be obtained by setting $C := V$ and $\tilde{Q} := Q$; a factorization $G = BR^{-1}B^T$ is given by $B = I_3$ and $R = \varepsilon$. This is used in both the FORTRAN 77 and MATLAB implementations if a factored form is required.
As solution we get

$$X = V\,\mathrm{diag}(x_1,\ x_2,\ x_3)\,V$$

where

$$\begin{aligned} x_1 &= \varepsilon, \\ x_2 &= \varepsilon\,\frac{(1+\sqrt{5})}{2}, \\ x_3 &= \varepsilon\,\frac{(9+\sqrt{85})}{2}. \end{aligned}$$

The closed-loop spectrum is given by $\lambda_1 = 0$, $\lambda_2 = \dfrac{(3-\sqrt{5})}{2}$, and $\lambda_3 = \dfrac{(11-\sqrt{85})}{6}$.
For growing ε, the corresponding symplectic pencil (2) becomes more and more badly scaled which leads to a significant loss of accuracy in all DARE solvers based on eigenvalue methods. This demonstrates the need to use an appropriate scaling as proposed in [11].

Example 14 [6, 24]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
4	1	1	$\tau = 2.0$, $D = 1.0$, $K = 2.0$, $r = 0.25$	∞	9.62×10^{-2}	1.05	1.05	6.20
			$\tau = 10^8$, $D = 1.0$, $K = 1.0$, $r = 0.25$	∞	$\approx 1 - \sqrt{5} \times 10^{-8}$	3.09×10^7	3.09×10^7	1.79×10^8
			$\tau = 10^{-6}$, $D = 1.0$, $K = 1.0$, $r = 0.25$	∞	2.0×10^{-7}	1.25	1.25	4.21×10^{12}

The following system describes a very simple process control of a paper machine. The continuous-time model with a time delay is sampled at intervals of length D which yields a singular transition matrix A. The time delay is equal to the length of three sampling intervals. The other parameters defining the system are a first-order time constant τ and the steady-state gain K. The linear system (7)–(8) is then given by

$$A = \begin{bmatrix} 1 - D/\tau & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} KD/\tau \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}.$$

The weighting matrices used in this example are $R = r$ and $\tilde{Q} = 1$. Defining

$$\alpha = 1 - \frac{D}{\tau}, \qquad \beta = \frac{KD}{\tau},$$

it can be shown that the solutions of the DARE (1) are given by

$$X = \operatorname{diag}(x_i,\ 1,\ 1,\ 1)$$

where x_i, $i = 1, 2$, solve the scalar quadratic equation

$$(\alpha^2 - 1)x + 1 - \frac{\alpha^2\beta^2}{r + \beta^2 x} x^2 = 0,$$

whence

$$x_i = \frac{1}{2\beta^2}\left(r(\alpha^2 - 1) + \beta^2 \pm \sqrt{(r(\alpha^2 - 1) + \beta^2)^2 + 4\beta^2 r} \right). \tag{17}$$

The stabilizing positive semidefinite solution of (1) is thus defined by the unique positive solution x_1 of (17) and the closed-loop eigenvalues are

$$\lambda_1 = \frac{\alpha r}{r + \beta^2 x_1} = \frac{(\tau - D)\tau r}{\tau^2 r + (DK)^2 x_1}, \qquad \lambda_2 = \lambda_3 = \lambda_4 = 0.$$

Due to the variety of parameters in this example, it is possible to investigate DAREs with critical properties in many aspects. Since these properties merely rely on α, β, and r, these effects can be produced by keeping K and D constant and varying τ (and r). Since $|\lambda^C_{max}| = \lambda_1$, for $\tau \gg D, K$ the largest closed-loop eigenvalue approaches the unit disk. For $\tau \ll D, K$ the norm and condition of X become large and the DARE becomes ill conditioned with respect to K_{DARE}.

4 Examples of scalable size

Example 15 [24, Example 3]

n	m	p	parameter	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
n	1	n	$n = 10,\ r = 1$	∞	0.00	10.00	10.00	11.01
			$\underline{n = 100,\ r = 1}$	∞	0.00	100.0	100.0	279.75

Consider the DARE defined by

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & & & 0 \\ 0 & & & 0 & 1 \\ 0 & \dots & & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times n}, \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \qquad R = r, \qquad Q = I_n.$$

The stabilizing solution has a very simple form, namely,

$$X = \mathrm{diag}(\,1,\,2,\,\dots,\,n\,).$$

The closed-loop eigenvalues are all zero, that is, the spectrum of the symplectic pencil $L - \lambda M$ in (2) is given by the generalized eigenvalues $\lambda_1 = \ldots = \lambda_n = 0$ and $\lambda_{n+1} = \ldots = \lambda_{2n} = \infty$.
This example can be used to test any DARE solver for growing dimension of the problem. The DARE condition number K_{DARE} increases only slowly and for any order of the DARE, $\lVert X\rVert = \kappa(X) = n$. Note further that the choice of r does not influence the stabilizing solution X but for $r < 1$, the condition number K_{DARE} behaves like $1/r$.

A The FORTRAN 77 subroutine DAREX

This is the prolog of a FORTRAN 77 subroutine for generating all presented examples. The subroutine was documented according to standards for SLICOT[4] [23].

Besides calls to LAPACK and BLAS [2], DAREX calls the subroutines SP2SY and SY2SP which are used to convert symmetric matrices from general storage mode to packed storage mode and vice versa. These subroutines are provided together with *darex.f*. If you have no access to LAPACK and BLAS, please contact the authors.

For some of the examples, DAREX reads the data from data files delivered together with *darex.f*. These are Examples 6–9 and 11. The corresponding data files (in ASCII format) are *DAREX6.DAT, DAREX7.DAT, DAREX8.DAT, DAREX9.DAT*, and *DAREX11.DAT*.

Note that the references given in the prolog of DAREX refer to those given at the end of the prolog and do not correspond to the references of this paper.

```
      SUBROUTINE DAREX(NO, N, M, P, NPAR, DPARAM, A, LDA, B, LDB, C,
     1                 LDC, Q, LDQ, R, LDR, S, LDS, X, LDX, NOTE,
     2                 STORE, WITHC, WITHG, WITHS, RWORK, IERR)
C
C     PURPOSE
C
C     To generate the benchmark examples for the numerical solution of
C     the discrete-time algebraic Riccati equation (DARE)
C
C            T                T                T    -1  T        T
C     0  =  A X A  -  X  -  (A X B + S) (R + B X B)  (B X A + S )  +  Q.
C
C     as presented in [1]. Here, A,Q,X are real N-by-N matrices, B,S are
C     N-by-M, and R is M-by-M. The matrices Q and R are symmetric and Q
C     may be given in factored form
C
C                   T
C     (I)    Q  =  C Q0 C .
C
C     Here, C is P-by-N and Q0 is P-by-P. If R is nonsingular, the DARE
C     can be rewritten equivalently as
C
C                                  -1
C     0  =  X  -  A X (I_n + G X)  A  -  Q
C
C     where I_n is the N-by-N identity matrix and
C
C                   -1  T
C     (II)   G = B R   B .
C
```

[4] Subroutine **LI**brary in **CO**ntrol and Systems **T**heory

```
C     ARGUMENT LIST
C       ARGUMENTS IN
C
C         NO - INTEGER.
C              The number of the benchmark example to generate according
C              to [1].
C
C          N - INTEGER.
C              This integer determines the actual state dimension, i.e.,
C              the order of the matrix A as follows:
C              N is fixed for the examples of Sections 2 and 3 of [1],
C              i.e., currently Examples 1-14.
C              NOTE that N is overwritten for Examples 1-14 and for the
C              other example(s) if N is set by default.
C
C       M, P - INTEGER.
C              M is the number of columns in the matrix B and the order
C              of the matrix R (in control problems, the number of
C              inputs of the system).
C              P is the number of rows in the matrix C from (I) (in
C              control problems, the number of outputs of the system).
C              Currently, M and P are fixed or determined by N for all
C              examples and thus are not referenced on input.
C              NOTE that M and P are overwritten and  M .LE. N  and
C              P .LE. N  for all examples.
C
C       NPAR - INTEGER.
C              Number of input parameters supplied by the user.
C              Examples  1-11 (Section 2 of [1]) have no parameters.
C              Examples 12-13 (Section 3 of [1]) each have one DOUBLE
C              PRECISION parameter which may be supplied in DPARAM(1).
C              Example 14 has 4 DOUBLE PRECISION parameters which may
C              be supplied in DPARAM(1) - DPARAM(4).
C              Example 15 has one INTEGER parameter which determines the
C              size of the problem. This parameter may be supplied in
C              the input argument N. Besides, this example has one
C              DOUBLE PRECISION parameter which may be supplied in
C              DPARAM(1).
C              If the input value of NPAR is less than the number of
C              parameters of the Example NO (according to [1]), the
C              missing parameters are set by default.
C
C     DPARAM - DOUBLE PRECISION array of DIMENSION at least ndp.
C              Double precision parameter vector where ndp is the
C              number of DOUBLE PRECISION parameters of Example NO
C              (according to [1]). For all examples, ndp <= 4. For
C              explanation of the parameters see [1].
```

```
C             DPARAM(1) defines the parameters 'epsilon' for the
C             examples in Section 3 (NO = 12,13), the parameter 'tau'
C             for NO = 14, and the parameter 'r' for NO = 15.
C             For Example 14, DPARAM(2) - DPARAM(4) define in
C             consecutive order 'D', 'K', and 'r'.
C             If NPAR is smaller than the number of used parameters in
C             Example NO (as described in [1]), default values are
C             used and returned in corresponding components of DPARAM.
C             NOTE that those entries of DPARAM are overwritten which
C             are used to generate the example but were not supplied by
C             the user.
C
C       LDA - INTEGER.
C             The leading dimension of array A as declared in the
C             calling program.
C             LDA .GE. N  where N is the order of the matrix A, i.e.,
C             the output value of the integer N.
C
C       LDB - INTEGER.
C             The leading dimension of array B as declared in the
C             calling program.
C             LDB .GE. N  (output value of N).
C
C       LDC - INTEGER.
C             The leading dimension of array C as declared in the
C             calling program.
C             LDC .GE. P  where P is either defined by default or
C             depends upon N. (For all examples, P .LE. N, where N is
C             the output value of the argument N.)
C
C       LDQ - INTEGER.
C             If full storage mode is used for Q, i.e., STORE = 'F'
C             or 'f', then Q is stored like a 2-dimensional array
C             with leading dimension LDQ. If packed symmetric storage
C             mode is used, then LDQ is not referenced.
C             That is, if STORE = 'F' or STORE = 'f', then
C             LDQ .GE. N  if WITHC = .FALSE.
C             LDQ .GE. P  if WITHC =  .TRUE.
C
C       LDR - INTEGER.
C             If full storage mode is used for the array R, i.e.,
C             STORE = 'F' or 'f', then R is stored like a 2-dimensional
C             array with leading dimension LDR. If packed symmetric
C             storage mode is used, then LDR is not referenced.
C             That is, if STORE = 'F' or STORE = 'f', then
C             LDR .GE. M  if WITHG = .FALSE.
C             LDR .GE. N  if WITHG =  .TRUE.
```

```
C
C       LDS - INTEGER.
C             The leading dimension of array S as declared in the
C             calling program.
C             LDS .GE. N  if S is to be returned (see MODE PARAMETER
C             WITHS). Otherwise, LDS is not referenced.
C
C       LDX - INTEGER.
C             The leading dimension of array X as declared in the
C             calling program.
C             LDX .GE. N  if an exact solution is available (Examples
C             1,3,5,12-15). Otherwise, X is not referenced.
C
C      ARGUMENTS OUT
C
C         N - INTEGER.
C             The order of the matrix A.
C
C         M - INTEGER.
C             The number of columns of matrix B and the order of the
C             matrix R.
C
C         P - INTEGER.
C             The number of rows of the matrix C from (I).
C
C    DPARAM - DOUBLE PRECISION array of DIMENSION at least 7.
C             Double precision parameter vector. For explanation of the
C             parameters see [1].
C             DPARAM(1) defines the parameters 'epsilon' for the
C             examples in Section 3 (NO = 12,13), the parameter 'tau'
C             if NO = 14, and the parameter 'r' if NO = 15.
C             For Example 14, DPARAM(2) - DPARAM(4) define in
C             consecutive order 'D', 'K', and 'r'.
C
C         A - DOUBLE PRECISION array of DIMENSION (LDA,N).
C             The leading N by N part of this array contains the
C             coefficient matrix A of the DARE.
C
C         B - DOUBLE PRECISION array of DIMENSION (LDB,M).
C             If WITHG = .FALSE., then array B contains the coefficient
C             matrix B of the DARE.
C             Otherwise, B is used as workspace.
C
C         C - DOUBLE PRECISION array of DIMENSION (LDC,N).
C             If WITHC = .TRUE., then array C contains the matrix C of
C             the factored form (I) of Q.
C             Otherwise, C is used as workspace.
```

```
C
C        Q - DOUBLE PRECISION array of DIMENSION at least qdim.
C            If STORE = 'F' or 'f',           then qdim = LDQ*nq.
C            If STORE = 'U', 'u', 'L' or 'l', then qdim = nq*(nq+1)/2.
C            If WITHC = .FALSE., then nq = N and the array Q
C            contains the coefficient matrix Q of the DARE.
C            If WITHC = .TRUE., then nq = P and the array Q contains
C            the matrix Q0 from (I).
C            The symmetric matrix contained in array Q is stored
C            according to MODE PARAMETER STORE.
C
C        R - DOUBLE PRECISION array of DIMENSION at least rdim.
C            If STORE = 'F' or 'f'           then rdim = LDR*nr.
C            If STORE = 'U', 'u', 'L' or 'l' then rdim = nr*(nr+1)/2.
C            If WITHG = .FALSE., then nr = M and the array R
C            contains the coefficient matrix R of the DARE.
C            If WITHG = .TRUE., then nr = N and the array R contains
C            the matrix G from (II).
C            The symmetric matrix contained in array R is stored
C            according to MODE PARAMETER STORE.
C
C        X - DOUBLE PRECISION array of DIMENSION (LDX,xdim).
C            If an exact solution is available (NO = 1,3,5,12-15),
C            then xdim = N and the leading N-by-N part of this array
C            contains the solution matrix X. Otherwise, X is not
C            referenced.
C
C     NOTE - CHARACTER*70.
C            String containing short information about the chosen
C            example.
C
C     WORK SPACE
C
C    RWORK - DOUBLE PRECISION array of DIMENSION at least N*N.
C
C     MODE PARAMETERS
C
C    STORE - CHARACTER.
C            Specifies the storage mode for arrays Q and R.
C            STORE = 'F' or 'f': Full symmetric matrices are stored in
C                                Q and R, i.e., the leading N-by-N
C                                (M-by-M, P-by-P) parts of these
C                                arrays each contain a symmetric
C                                matrix.
C            STORE = 'L' or 'l': Matrices contained in arrays Q and R
C                                are stored in lower packed mode, that
C                                is, the lower triangle of a k-by-k
```

```
C                                  (k=N,M,P) symmetric matrix is stored
C                                  by columns, i.e., the matrix entry
C                                  Q(i,j) is stored in the array entry
C                                  Q(i+(2*k-j)*(j-1)/2)  for  j <= i.
C              STORE = 'U' or 'u': Matrices contained in arrays Q and R
C                                  are stored in upper packed mode, that
C                                  is, the upper triangle of a k-by-k
C                                  (k=N,M,P) symmetric matrix is stored
C                                  by columns, i.e., the matrix entry
C                                  G(i,j) is stored in the array entry
C                                  G(i+j*(j-1)/2)  for  i <= j.
C              Otherwise, CAREX returns with an error.
C
C      WITHC - LOGICAL.
C              Indicates whether the matrices C, Q0 as in (I) are to be
C              returned as follows.
C              WITHC =  .TRUE., C is returned in array C and Q0 is
C                               returned in array Q.
C              WITHC = .FALSE., the coefficient matrix Q of the DARE is
C                               returned in array Q, whereas C and Q0
C                               are not returned.
C
C      WITHG - LOGICAL.
C              Indicates whether the matrix G in (II) or the matrices B
C              and R are returned as follows.
C              WITHG =  .TRUE., the matrix G from (II) is returned in
C                               array R, whereas the matrices B and R
C                               are not returned.
C              WITHG = .FALSE., the coefficient matrices B and R of the
C                               DARE are returned in arrays B and R.
C
C      WITHS - LOGICAL.
C              Indicates whether the coefficient matrix S of the DARE
C              is returned as follows.
C              WITHS =  .TRUE., the coefficient matrix S of the DARE is
C                               returned in array S.
C              WITHS = .FALSE., the coefficient matrix S of the DARE is
C                               not returned.
C
C     ERROR INDICATOR
C
C       IERR - INTEGER.
C              Unless the routine detects an error (see next section),
C              IERR contains 0 on exit.
C
C     WARNINGS AND ERRORS DETECTED BY THE ROUTINE
C
```

```
C     IERR = 1 : (NO .LT. 1) or (NO .GT. NEX).
C                (NEX = number of available examples.)
C     IERR = 2 : (N .LT. 1) or (N .GT. LDA) or (N .GT. LDB) or
C                or (P .GT. LDC) or (WITHS and N .GT. LDS) or
C                (N .GT. LDX  and solution is available) or
C                ((STORE = 'F' or STORE = 'f') and
C                 ((WITHC .EQ. .FALSE. and N .GT. LDQ) or
C                  (WITHC .EQ.  .TRUE. and P .GT. LDQ)) or
C                 ((WITHG .EQ. .FALSE. and M .GT. LDR) or
C                  (WITHG .EQ.  .TRUE. and N .GT. LDR))).
C     IERR = 3 : MODE PARAMETER STORE had an illegal value on input.
C     IERR = 4 : Data file could not be opened or had wrong format.
C     IERR = 5 : Division by zero.
C     IERR = 6 : G can not be computed as in (II) due to a singular R
C                matrix. This error can only occur if
C                (WITHG .EQ. .TRUE.).
C
C     REFERENCES
C
C     [1] P. BENNER, A.J. LAUB and V. MEHRMANN
C         A Collection of Benchmark Examples for the Numerical Solution
C         of Algebraic Riccati Equations II: Discrete-Time Case.
C         Technical Report SPC 95_23, Fak. f. Mathematik,
C         TU Chemnitz-Zwickau (Germany), December 1995.
C     [2] E. ANDERSON ET AL.
C         LAPACK Users' Guide, second edition.
C         SIAM, Philadelphia, PA (1994).
C
C     CONTRIBUTOR
C
C     Peter Benner and Volker Mehrmann (TU Chemnitz-Zwickau)
C     Alan J. Laub (University of California, Santa Barbara)
C
C     KEYWORDS
C
C     discrete-time, algebraic Riccati equation, Hamiltonian matrix
C
C     REVISIONS
C
C     1995, December 14.
C
C***********************************************************************
```

B The MATLAB function darex

The prolog of the MATLAB function `darex` is listed below. For all listed examples, it is possible to return the matrices A, B, R, Q, and the factors C, $\tilde{Q}$ = `Q0`. $G = BR^{-1}B^T$ can also be returned if R is nonsingular. Otherwise, the output argument `G` will contain an empty matrix. If the solution is not available, the output argument `X` contains an empty matrix, too. Otherwise, X is returned as well as the DARE condition number K_{DARE} computed by the MATLAB function `darecond`.
Note that the references given in the prolog of `darex` refer to those given at the end of the prolog and do not correspond to the references of this paper.

```
function [A,B,Q,R,S,X,parout,G,C,Q0]=darex(index,parin)
%DAREX
%
% Test examples for the discrete-time algebraic Riccati equation (DARE)
%
%                                                      -1
% (I) 0 = DR(X) = A'XA - X - (A'XB + S) (R + B'XB)  (B'XA + S') + Q
%
% Here, A,Q, and X are n-by-n matrices, B and S are n-by-m, and R is
% m-by-m. Q and R are symmetric and X is the required solution matrix.
% One common approach to solve DAREs is to compute a deflating subspace
% of the symplectic pencil
%
%                              -1
%                     ( A - B R  S     0 )     ( I        G        )
% (II)    L - s M := (    -1             ) - s (              -1    )
%                     ( S R  S' - Q   I )     ( 0   (A - B R  S')')
%
%               -1
% where  G = B R  B' is a symmetric n-by-n matrix. Q may also be given
% in factored form, Q = C' Q0 C, where C is a p-by-n and Q0 is a p-by-p
% matrix.
% NOTE that for DAREs, R being a singular matrix is not uncommon. In this
% case, the symplectic pencil cannot be formed as in (II), but a solution
% of the DARE can be computed via a deflating subspace of the extended
% pencil
%
%                        ( A   0   B )     ( I   0   0 )
% (III)    LL - s MM := ( Q  -I   S ) - s ( 0  -A'  0 ) .
%                        ( S'  0   R )     ( 0  -B'  0 )
%
% For examples with singular R-matrix, G can not be computed and is thus
% not returned.
%
% Input:
%  - index: number of example to generate, indices refer to example
```

```
%           numbers in [1].
%  - parin: input parameters (optional, defaults values given in [1])
%           For Example number
%           +  1-11: not referenced ([1], Section 2).
%           + 12-13: parin(1) = real-valued scalar.
%           + 14   : parin(1:4) = [tau, D, K, r], real-valued scalars.
%           + 15   : parin(1) = n = problem size.
%                    parin(2) = r = real-valued scalar.
%
% Output:
%  - A, B, Q, R, S: coefficient matrices of DARE as in (I).
%  - X            : exact solution of DARE (if available), usually the
%                   stabilizing solution.
%                   If an exact solution is not available, the empty matrix
%                   is returned.
%  - parout       : vector with system properties,
%                   parout(1:3) = [n, m, p].
%                   parout(4)   = 2-norm condition number of A.
%                   The following parameters are only returned if an
%                   solution of the DARE is available:
%                   parout(5)   = radius of smallest circle enclosing the
%                                 closed-loop spectrum.
%                   parout(6)   = 2-norm of X.
%                   parout(7)   = 2-norm condition number of X.
%                   parout(8)   = condition number of DARE as defined in [2].
%  - G, C, Q0     : optional output matrices as defined above. NOTE that
%                   G can only be computed if R is nonsingular. Otherwise,
%                   G contains on output the empty matrix.
%
% References:
%
% [1] P.BENNER, A.J. LAUB, V. MEHRMANN: 'A Collection of Benchmark
%     Examples for the Numerical Solution of Algebraic Riccati
%     Equations II: Discrete-Time Case', Tech. Report SPC 95_23,
%     Fak. f. Mathematik, TU Chemnitz-Zwickau (Germany), December 1995.
% [2] T. GUDMUNDSSON, C. KENNEY, A.J. LAUB: 'Scaling of the Discrete-Time
%     Algebraic Riccati Equation to Enhance Stability of the Schur
%     Solution Method', IEEE Transactions on Automatic Control, vol. 37,
%     no. 4, pp. 513-518, 1992.

%  Peter Benner, Volker Mehrmann (TU Chemnitz-Zwickau, Germany),
%  Alan J. Laub (University of California at Santa Barbara)
%  12-14-1995
```

C How to obtain the software

The codes corresponding to this paper may be obtained via anonymous ftp at TU Chemnitz-Zwickau. Proceed as follows.

> \> `ftp ftp.tu-chemnitz.de`
> \> Name: `anonymous`
> \> Password: *your complete e-mail address*
> \> `cd pub/Local/mathematatik/Benner`

Observe the capital "L" in `Local` !
Now get the compressed FORTRAN 77 subroutines *darex.f, sp2sy.f, sy2sp.f*, data files, a sample *Makefile*, and a sample program *example.f* together with an introductory *README* file by

> \> `get darex_f.tar.Z`

or the compressed MATLAB function files *darex.m, darecond.m* and an introductory *README* file by

> \> `get darex_m.tar.Z`

After exiting ftp, extracting the MATLAB codes and data files is achieved by the following commands:

> \> `uncompress darex_m.tar.Z`
> \> `tar xf darex_m.tar`

Analogously, the FORTRAN 77 codes and corresponding data files are obtained by

> \> `uncompress darex_f.tar.Z`
> \> `tar xf darex_f.tar`

In both cases, the command *tar xf* creates a directory containing all required files. For *darex_m.tar.Z*, this directory is called *darex_m* and for *darex_f.tar.Z*, it will be *darex_f*. If any problems occur in obtaining or running the codes, please contact one of the authors.

D Reference table

Table 1 summarizes the properties of all the presented examples. A value "∞" for a condition number means that the corresponding matrix is not invertible with respect to the numerical rank computed by MATLAB. If $K_{DARE} = \infty$, the DARE condition number from [11] is not defined for this example. The column X^* indicates whether an analytical stabilizing solution is available ("+") or not ("–").

no.	n	m	p	default	X^*	$\kappa(A)$	$\lvert\lambda^C_{max}\rvert$	$\lVert X\rVert$	$\kappa(X)$	K_{DARE}
1	2	1	2	–	+	114.99	0.50	21.03	∞	18.85
2	2	2	2	–	–	1.05	0.69	5.07×10^{-2}	4.97	4.74
3	2	1	1	–	+	5.83	0.00	1.00	1.00	∞
4	2	2	2	–	–	∞	0.69	126.99	2.84×10^{3}	∞
5	2	1	2	–	+	∞	0.38	5.19	114.13	1.88
6	4	2	4	–	–	1.01	0.94	35.36	3.34	30.58
7	4	2	4	–	–	19.86	0.99	2.06	183.33	790.37
8	4	4	4	–	–	378.60	$\approx 1 - \frac{1.8}{10^5}$	65.77	6.81×10^{12}	5.12×10^{4}
9	5	2	5	–	–	23.52	0.98	73.90	73.73	100.81
10	6	2	2	–	–	∞	0.67	2.53	37.38	3.94
11	9	3	2	–	–	1.58×10^{6}	0.96	607.66	1.44×10^{24}	74.23
12	2	1	2	$\varepsilon = 10^6$	+	∞	0.00	1.00×10^{12}	1.00×10^{12}	2.65
13	3	3	3	$\varepsilon = 10^6$	+	∞	0.38	9.11×10^{6}	9.11	2.51
14	4	1	1	$\tau = 10^8$, $D = 1.0$ $K = 1.0$, $r = 0.25$	+	∞	$\approx 1 - \frac{\sqrt{5}}{10^8}$	3.09×10^{7}	3.09×10^{7}	1.79×10^{8}
15	n	1	n	$n = 100$, $r = 1$	+	∞	0.00	100.00	100.00	279.75

Table 1

References

[1] G. Ackerson and K. Fu, *On the state estimation in switching environments*, IEEE Trans. Automat. Control, AC-15 (1970), pp. 10–17.

[2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, SIAM, Philadelphia, PA, second ed., 1994.

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

[4] Z. Bai, J. Demmel, and M. Gu, *Inverse free parallel spectral divide and conquer algorithms for nonsymmetric eigenproblems*, Tech. Report LBL-34969, Lawrence Berkeley Laboratory, University of California, Berkeley, CA 94720, 1994.

[5] P. Benner, A. Laub, and V. Mehrmann, *A collection of benchmark examples for the numerical solution of algebraic Riccati equations I: Continuous-time case*, Tech. Report SPC 95_22, Fak. f. Mathematik, TU Chemnitz–Zwickau, 09107 Chemnitz, FRG, 1995.

[6] W. Bialkowski, *Application of steady-state Kalman filters–Theory with field results*, in Proc. Joint Automat. Cont. Conf., Philadelphia, PA, 1978.

[7] E. Davison and S. Wang, *Properties and calculation of transmission zeros of linear multivariable systems*, Automatica, 10 (1974), pp. 643–658.

[8] Z. Gajić and X. Shen, *Parallel Algorithms for Optimal Control of Large Scale Linear Systems*, Springer-Verlag, London, 1993.

[9] J. Gardiner and A. Laub, *A generalization of the matrix-sign-function solution for algebraic Riccati equations*, Internat. J. Control, 44 (1986), pp. 823–832. (see also *Proc. 1985 CDC*, pp. 1233–1235).

[10] K. Gomathi, S. Prabhu, and M. Pai, *A suboptimal controller for minimum sensitivity of closed-loop eigenvalues to parameter variations*, IEEE Trans. Automat. Control, AC-25 (1980), pp. 587–588.

[11] T. Gudmundsson, C. Kenney, and A. Laub, *Scaling of the discrete-time algebraic Riccati equation to enhance stability of the Schur solution method*, IEEE Trans. Automat. Control, AC-37 (1992), pp. 513–518.

[12] G. Hewer, *An iterative technique for the computation of steady state gains for the discrete optimal regulator*, IEEE Trans. Automat. Control, AC-16 (1971), pp. 382–384.

[13] V. Ionescu and M. Weiss, *On computing the stabilizing solution of the discrete-time Riccati equation*, Linear Algebra Appl., 174 (1992), pp. 229–238.

[14] E. Jonckheere, *On the existence of a negative semidefinite antistabilizing solution to the discrete algebraic Riccati equation*, IEEE Trans. Automat. Control, AC-26 (1981), pp. 707–712.

[15] M. Konstantinov, P. Petkov, and N. Christov, *Perturbation analysis of the discrete Riccati equation*, Kybernetica, 29 (1993), pp. 18–29.

[16] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, New York, 1972.

[17] A. Laub, *A Schur method for solving algebraic Riccati equations*, IEEE Trans. Automat. Control, AC-24 (1979), pp. 913–921. (see also *Proc. 1978 CDC (Jan. 1979)*, pp. 60-65).

[18] ——, *Algebraic aspects of generalized eigenvalue problems for solving Riccati equations*, in Computational and Combinatorial Methods in Systems Theory, C. Byrnes and A. Lindquist, eds., Elsevier (North-Holland), 1986, pp. 213–227.

[19] B. Litkouhi, *Sampled-Data Control of Systems with Slow and Fast Modes*, PhD thesis, Michigan State University, 1983.

[20] L. Lu and W. Lin, *An iterative algorithm for the solution of the discrete time algebraic Riccati equation*, Linear Algebra Appl., 188/189 (1993), pp. 465–488.

[21] V. Mehrmann, *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*, no. 163 in Lecture Notes in Control and Information Sciences, Springer-Verlag, Heidelberg, July 1991.

[22] ——, *A step towards a unified treatment of continuous and discrete time control problems*, Linear Algebra Appl., to appear (1996). (see also: *Preprint SPC 94_20, Fak. f. Mathematik, TU Chemnitz-Zwickau, Chemnitz, FRG*).

[23] Numerical Algorithms Group, *Implementation and documentation standards for the subroutine library in control and systems theory SLICOT*, Publication NP2032, Numerical Algorithms Group, Eindhoven/Oxford, 1990.

[24] T. Pappas, A. Laub, and N. Sandell, *On the numerical solution of the discrete-time algebraic Riccati equation*, IEEE Trans. Automat. Control, AC-25 (1980), pp. 631–641.

[25] L. Patnaik, N. Viswanadham, and I. Sarma, *Computer control algorithms for a tubular ammonia reactor*, IEEE Trans. Automat. Control, AC-25 (1980), pp. 642–651.

[26] P. Petkov, N. Christov, and M. Konstantinov, *Numerical properties of the generalized Schur approach for solving the discrete matrix Riccati equation*, in Proc. 18th Spring Conference of the Union of Bulgarian Mathematicians, Albena, 1989, pp. 452–457.

[27] ——, *A posteriori error analysis of the generalized Schur approach for solving the discrete matrix Riccati equation*, preprint, Department of Automatics, Higher Institute of Mechanical and Electrical Engineering, 1756 Sofia, Bulgaria, 1989.

[28] A. Sage, *Optimum Systems Control*, Prentice-Hall, Englewood Cliffs, NJ, 1966.

[29] V. Shankar and K. Ramar, *Pole assignment with minimum eigenvalue sensitivity to parameter variations*, Internat. J. Control, 23 (1976), pp. 493–504.

[30] J. Sun, *Backward error of the discrete-time algebraic Riccati equation*, preprint, Dept. Computing Science, Umeå University, Umeå, Sweden, 1995.

[31] P. Van Dooren, *A generalized eigenvalue approach for solving Riccati equations*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 121–135.

Other titles in the SPC series:

95_1 T. Apel, G. Lube. Anisotropic mesh refinement in stabilized Galerkin methods Januar 1995.

95_2 M. Meisel, A. Meyer. Implementierung eines parallelen vorkonditionierten Schur-Komplement CG-Verfahrens in das Programmpaket FEAP. Januar 1995.

95_3 S. V. Nepomnyaschikh. Optimal multilevel extension operators. January 1995

95_4 M. Meyer. Grafik-Ausgabe vom Parallelrechner für 3D-Gebiete. Januar 1995

95_5 T. Apel, G. Haase, A. Meyer, M. Pester. Parallel solution of finite element equation systems: efficient inter-processor communication. Februar 1995

95_6 U. Groh. Ein technologisches Konzept zur Erzeugung adaptiver hierarchischer Netze für FEM-Schemata. Mai 1995

95_7 M. Bollhöfer, C. He, V. Mehrmann. Modified block Jacobi preconditioners for the conjugate gradient method. Part I: The positive definit case. January 1995

95_8 P. Kunkel, V. Mehrmann, W. Rath, J. Weickert. GELDA: A Software Package for the Solution of General Linear Differential Algebraic Equation. February 1995

95_9 H. Matthes. A DD preconditioner for the clamped plate problem. February 1995

95_10 G. Kunert. Ein Residuenfehlerschätzer für anisotrope Tetraedernetze und Dreiecksnetze in der Finite-Elemente-Methode. März 1995

95_11 M. Bollhöfer. Algebraic Domain Decomposition. March 1995

95_12 B. Nkemzi. Partielle Fourierdekomposition für das lineare Elastizitätsproblem in rotationssymmetrischen Gebieten. März 1995

95_13 A. Meyer, D. Michael. Some remarks on the simulation of elasto-plastic problems on parallel computers. March 1995

95_14 B. Heinrich, S. Nicaise, B. Weber. Elliptic interface problems in axisymmetric domains. Part I: Singular functions of non-tensorial type. April 1995

95_15 B. Heinrich, B. Lang, B. Weber. Parallel computation of Fourier-finite-element approximations and some experiments. May 1995

95_16 W. Rath. Canonical forms for linear descriptor systems with variable coefficients. May 1995

95_17 C. He, A. J. Laub, V. Mehrmann. Placing plenty of poles is pretty preposterous. May 1995

95_18 J. J. Hench, C. He, V. Kučera, V. Mehrmann. Dampening controllers via a Riccati equation approach. May 1995

95_19 M. Meisel, A. Meyer. Kommunikationstechnologien beim parallelen vorkonditionierten Schur-Komplement CG-Verfahren. Juni 1995

95_20 G. Haase, T. Hommel, A. Meyer and M. Pester. Bibliotheken zur Entwicklung paralleler Algorithmen. Juni 1995.

95_21 A. Vogel. Solvers for Lamé equations with Poisson ratio near 0.5. June 1995.

95_22 P. Benner, A. J. Laub, V. Mehrmann. A collection of benchmark examples for the numerical solution of algebraic Riccati equations I: Continuous-time case. October 1995.

95_23 P. Benner, A. J. Laub, V. Mehrmann. A collection of benchmark examples for the numerical solution of algebraic Riccati equations II: Discrete-time case. December 1995.

95_24 P. Benner, R. Byers. Newton's method with exact line search for solving the algebraic Riccati equation. October 1995.

95_25 P. Kunkel, V. Mehrmann. Local and Global Invariants of Linear Differential-Algebraic Equations and their Relation. July 1995.

95_26 C. Israel. NETGEN69 - Ein hierarchischer paralleler Netzgenerator. August 1995.

95_27 M. Jung. Parallelization of multi-grid methods based on domain decomposition ideas. November 1995.

95_28 P. Benner, H. Faßbender. A restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem. October 1995.

95_29 G. Windisch. Exact discretizations of two-point boundary value problems. October 1995.

95_30 S. V. Nepomnyaschikh. Domain decomposition and multilevel techniques for preconditioning operators. November 1995.

95_31 H. Matthes. Parallel preconditioners for plate problems. November 1995.

95_32 V. Mehrmann, H. Xu. An analysis of the pole placement problem. I. The single input case. November 1995.

95_33 Th. Apel. SPC-PM Po 3D — User's manual. December 1995.

95_34 Th. Apel, F. Milde, M. Theß. SPC-PM Po 3D — Programmer's manual. December 1995.

95_35 S. A. Ivanov, V. G. Korneev. On the preconditioning in the domain decomposition technique for the p-version finite element method. Part I. December 1995.

95_36 S. A. Ivanov, V. G. Korneev. On the preconditioning in the domain decomposition technique for the p-version finite element method. Part II. December 1995.

95_37 V. Mehrmann, N. K. Nichols. Mixed output feedback for descriptor systems. December 1995.

Some papers can be accessed via anonymous ftp from server `ftp.tu-chemnitz.de`,
directory `pub/Local/mathematik/SPC`. (Note the capital L in Local!)
The complete list of current and former preprints is available via
`http://www.tu-chemnitz.de/~pester/sfb/spc95pr.html`.