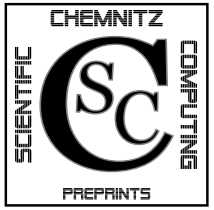
TECHNISCHE UNIVERSITÄT CHEMNITZ

Peter Benner Heike Faßbender

On the solution of the rational matrix equation $X = Q + LX^{-1}L^T$

CSC/06-02



Chemnitz Scientific Computing Preprints

Chemnitz Scientific Computing Preprints – ISSN 1864-0087

Impressum:

Chemnitz Scientific Computing Preprints — ISSN 1864-0087

(1995–2005: Preprintreihe des Chemnitzer SFB393)

Herausgeber:
Professuren für
Numerische und Angewandte Mathematik
an der Fakultät für Mathematik
der Technischen Universität Chemnitz

Postanschrift:
TU Chemnitz, Fakultät für Mathematik
09107 Chemnitz
Sitz:
Reichenhainer Str. 41, 09126 Chemnitz

`http://www.tu-chemnitz.de/mathematik/csc/`

The complete list of SFB393 preprints is available via
`http://www.tu-chemnitz.de/sfb393/preprints.html`.

Some titles of the former SFB393 preprint series:

04-01 A. Meyer, F. Rabold, M. Scherzer. Efficient Finite Element Simulation of Crack Propagation. February 2004.

04-02 S. Grosman. The robustness of the hierarchical a posteriori error estimator for reaction-diffusion equation on anisotropic meshes. March 2004.

04-03 A. Bucher, A. Meyer, U.-J. Görke, R. Kreißig. Entwicklung von adaptiven Algorithmen für nichtlineare FEM. April 2004.

04-04 A. Meyer, R. Unger. Projection methods for contact problems in elasticity. April 2004.

04-05 T. Eibner, J. M. Melenk. A local error analysis of the boundary concentrated FEM. May 2004.

04-06 H. Harbrecht, U. Kähler, R. Schneider. Wavelet Galerkin BEM on unstructured meshes. May 2004.

04-07 M. Randrianarivony, G. Brunnett. Necessary and sufficient conditions for the regularity of a planar Coons map. May 2004.

04-08 P. Benner, E. S. Quintana-Ortí, G. Quintana-Ortí. Solving Linear Matrix Equations via Rational Iterative Schemes. October 2004.

04-09 C. Pester. Hamiltonian eigenvalue symmetry for quadratic operator eigenvalue problems. October 2004.

04-10 T. Eibner, J. M. Melenk. An adaptive strategy for hp-FEM based on testing for analyticity. November 2004.

04-11 B. Heinrich, B. Jung. The Fourier-finite-element method with Nitsche-mortaring. November 2004.

04-12 A. Meyer, C. Pester. The Laplace and the linear elasticity problems near polyhedral corners and associated eigenvalue problems. December 2004.

04-13 M. Jung, T. D. Todorov. On the Convergence Factor in Multilevel Methods for Solving 3D Elasticity Problems. December 2004.

05-01 C. Pester. A residual a posteriori error estimator for the eigenvalue problem for the Laplace-Beltrami operator. January 2005.

05-02 J. Badía, P. Benner, R. Mayo, E. Quintana-Ortí, G. Quintana-Ortí, J. Saak. Parallel Order Reduction via Balanced Truncation for Optimal Cooling of Steel Profiles. February 2005.

05-03 C. Pester. CoCoS – Computation of Corner Singularities. April 2005.

05-04 A. Meyer, P. Nestler. Mindlin-Reissner-Platte: Einige Elemente, Fehlerschätzer und Ergebnisse. April 2005.

05-05 P. Benner, J. Saak. Linear-Quadratic Regulator Design for Optimal Cooling of Steel Profiles. April 2005.

05-06 A. Meyer. A New Efficient Preconditioner for Crack Growth Problems. April 2005.

TECHNISCHE UNIVERSITÄT CHEMNITZ

Chemnitz Scientific Computing Preprints

Peter Benner Heike Faßbender

On the solution of the rational matrix equation $X = Q + LX^{-1}L^T$

CSC/06-02

CSC/06-02 ISSN 1864-0087 September 2006

Contents

Peter Benner
TU Chemnitz
Fakultät für Mathematik
D-09107 Chemnitz

benner@mathematik.tu-chemnitz.de

Heike Faßbender
TU Braunschweig
Institut *Computational Mathematics*
38106 Braunschweig

h.fassbender@tu-bs.de

[24] A.N. Malyshev. Parallel algorithm for solving some spectral problems of linear algebra. *Linear Algebra Appl.*, 188/189:489–520, 1993.

[25] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*. Number 163 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Heidelberg, July 1991.

[26] B. Meini. Efficient computation of the extreme solutions of $X+A^*X^{-1}A=Q$ and $X-A^*X^{-1}A=Q$. *Math. Comp.* 71(239):1189–1204, 2001.

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

[28] M. Reurings. *Symmetric Matrix Equations*. PhD Thesis, Amsterdam, ISBN 90-9016681-5, NUGI 918, 2003.

[29] V. Sima. *Algorithms for Linear-Quadratic Optimization*, volume 200 of *Pure and Applied Mathematics*. Marcel Dekker, Inc., New York, NY, 1996.

[30] M. Slowik, P. Benner, and V. Sima. Evaluation of the Linear Matrix Equation Solvers in SLICOT. *SLICOT Working Note*, 2004–1, 2004. Available from `www.icm.tu-bs.de/niconet/`.

[31] X. Sun and E.S. Quintana-Orti. Spectral divison methods for block generalized Schur decompositions. *Math. Comp.*, 73:1827–1847, 2004.

[32] D.S. Watkins and L. Elsner. Theory of decomposition and bulge chasing algorithms for the generalized eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 15:943–967, 1994.

[33] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Upper Saddle River, NJ, 1995.

On the solution of the rational matrix equation $X = Q + LX^{-1}L^T$

Peter Benner* Heike Faßbender†

September 30, 2006

Abstract: We study numerical methods for finding the maximal symmetric positive definite solution of the nonlinear matrix equation $X = Q + LX^{-1}L^T$, where Q is symmetric positive definite and L is nonsingular. Such equations arise for instance in the analysis of stationary Gaussian reciprocal processes over a finite interval. Its unique largest positive definite solution coincides with the unique positive definite solution of a related discrete-time algebraic Riccati equation (DARE). We discuss how to use the butterfly SZ algorithm to solve the DARE. This approach is compared to several fixed point type iterative methods suggested in the literature.

1 Introduction

The nonlinear matrix equation

$$X = f(X) \quad \text{with} \quad f(X) = Q + LX^{-1}L^T, \tag{1}$$

where $Q = Q^T \in \mathbb{R}^{n\times n}$ is positive definite and $L \in \mathbb{R}^{n\times n}$ is nonsingular, arises in the analysis of stationary Gaussian reciprocal processes over a finite interval. The solution of certain 1-D stochastic boundary-value systems are reciprocal processes. For instance, the steady-state distribution of the temperature along a heated ring or beam subjected to random loads along its length can be modeled in terms of such reciprocal processes. A different example is a ship surveillance problem: given a Gauss-Markov state-space model of the ship's trajectory, it is desired to assign a probability distribution not only to the initial state, but also

*Technische Universität Chemnitz, Fakultät für Mathematik, 09107 Chemnitz, Germany, `benner@mathematik.tu-chemnitz.de`

†Technische Universität Braunschweig, Institut *Computational Mathematics*, 38106 Braunschweig, Germany, `h.fassbender@tu-bs.de`

to the final state, corresponding to some predictive information about the ship's destination. This has the effect of modeling the trajectory as a reciprocal process. For references to these examples see, e.g., [22].

The problem considered here is to find the (unique) largest positive definite symmetric solution X_+ of (1). This equation has been considered, e.g., in [12, 15, 18, 23, 26, 28]. In [12], the set of Hermitian solutions of (1) is characterized in terms of the spectral factors of the matrix Laurent polynomial $\mathcal{L}(z) = Q + Lz - L^T z^{-1}$. These factors are related to the Lagrangian deflating subspace of the matrix pencil

$$G - \lambda H = \begin{bmatrix} L^T & 0 \\ -Q & I \end{bmatrix} - \lambda \begin{bmatrix} 0 & I \\ L & 0 \end{bmatrix}. \tag{2}$$

In particular, one can conclude from the results in [12, Section 2] that this matrix pencil does not have any eigenvalues on the unit circle and that the spectral radius $\rho(X_+^{-1}L^T)$ is less than 1 as $\left[\begin{smallmatrix} I \\ X_+ \end{smallmatrix}\right]$ spans the stable Lagrangian deflating subspace of $G - \lambda H$. Alternatively, one could rewrite (1) as the discrete Lyapunov equation $X_+ - (X_+^{-1}L^T)^T X_+ (X_+^{-1}L^T) = Q$. As Q and X_+ are positive definite, we get $\rho(X_+^{-1}L^T) < 1$ from the discrete version of the Lyapunov stability theorem (see, e.g., [20, p. 451]). Moreover, it is shown in [12], that the unique largest positive definite solution of (1) coincides with the unique positive definite solution of a related Riccati equation. For this, it is noted in [12] that if X solves (1), then it also obeys the iterated equation

$$X = f(f(X)) = Q + F(R^{-1} + X^{-1})^{-1} F^T$$

with $F = LL^{-T}$ and $R = L^T Q^{-1} L = R^T$ positive definite. Using the Sherman-Morrison-Woodbury formula to derive an expression for $(R^{-1}+X^{-1})^{-1}$, we obtain

$$\mathcal{DR}(X) = Q + FXF^T - FX(X+R)^{-1}XF^T - X, \tag{3}$$

a discrete-time algebraic Riccati equation (DARE). Because (F, I) is controllable and (F, Q) is observable, a unique stabilizing positive definite solution X_* exists [19, Theorem 13.1.3]. This unique solution coincides with that solution of (1) one is interested in. DAREs appear not only in the context presented, but also in numerous procedures for analysis, synthesis, and design of control and estimation systems with H_2 or H_∞ performance criteria, as well as in other branches of applied mathematics and engineering, see, e.g., [1, 2, 3, 19, 33].

In [12] essentially three ideas for solving (1) have been proposed. The straightforward one is a basic iterative algorithm that converges to the desired positive definite solution X_+ of (1). Essentially, the algorithm interprets equation (1) as a fixed point equation and iterates $X_{i+1} = f(X_i)$; see Section 2.1 for more details.

The second idea is to compute the desired solution from the stable Lagrangian deflating subspace of $G - \lambda H$. If we can compute $Y_1, Y_2 \in \mathbb{R}^{n\times n}$ such that the

[11] H. Faßbender. *Symplectic Methods for the Symplectic Eigenproblem*. Kluwer Academic/Plenum Publishers, New York, 2000.

[12] A. Ferrante and B.B. Levy. Hermitian Solutions of the Equation $X = Q + NX^{-1}N^\star$. *Linear Algebra Appl.*, 247:359–373, 1996.

[13] J.D. Gardiner, A.J. Laub, J.J. Amato and C.B. Moler. Solution of the Sylvester matrix equation $AXB^T + CXD^T = E$. *ACM Trans. Math. Software* 18:223–231, 1992.

[14] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[15] C.-H. Guo and P. Lancaster. Iterative solution of two matrix equations. *Math. Comp.* 68(228):1589–1603, 1999.

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

[17] R. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press 1994.

[18] I.G. Ivanov, V.I. Hasanov and F. Uhlig. Improved methods and starting values to solve the matrix equations $X \pm A^* X^{-1} A = I$ iteratively. *Math. Comp.* 74(249):263–278, 2004.

[19] P. Lancaster and L. Rodman. *The Algebraic Riccati Equation*. Oxford University Press, Oxford, 1995.

[20] P. Lancaster and M. Tismenetsky. *The theory of matrices*. Second edition, Academic Press, Orlando, FL, 1985.

[21] A.J. Laub. Algebraic aspects of generalized eigenvalue problems for solving Riccati equations. In C.I. Byrnes and A. Lindquist, editors, *Computational and Combinatorial Methods in Systems Theory*, pages 213–227. Elsevier (North-Holland), 1986.

[22] B.C. Levy, R. Frezza and A.J. Kerner. Modeling and estimation of discrete-time Gaussian reciprocal processes. *IEEE Trans. Automat. Control* AC-90:1013–1023, 1990.

[23] W.-W. Lin and S.-F. Xu. Convergence analysis of structure-preserving doubling algorithms for Riccati-type matrix equations. *SIAM J. Matrix Anal. Appl.*, 28:26–39, 2006.

can be used to improve the accuracy of a computed solution. Several examples comparing the iterative methods with the SZ approach show that none of the methods discussed is superior. Usually, the doubling-type algorithm computes the approximate solution very fast, but due to the back transformation step, the accuracy can deteriorate significantly. On the other hand, the fixed point iteration is often very slow. The SZ approach needs a predictable computing time which is most often less than that of the fixed point iteration when a comparable accuracy is requested, but is much higher than for the doubling algorithm. The accuracy of the SZ approach is not always the best compared to the other methods, but in none of the examples tested it fails as opposed to the iterative methods.

References

[1] C.D. Ahlbrandt and A.C. Peterson. *Discrete Hamiltonian Systems: Difference Equations, Continued Fractions, and Riccati Equations*. Kluwer Academic Publishers, Dordrecht, NL, 1998.

[2] B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.

[3] B.D.O. Anderson and B. Vongpanitlerd. *Network Analysis and Synthesis. A Modern Systems Approach*. Prentice-Hall, Englewood Cliffs, NJ, 1972.

[4] A.C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, Philadelphia, PA, 2005.

[5] Z. Bai, J. Demmel and M. Gu. An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. *Numer. Math.*, 76:279–208, 1997.

[6] A. Y. Barraud. A numerical algorithm to solve $A^TXA - X = Q$. *IEEE Trans. Automat. Control*, AC-22:883–885, 1977.

[7] P. Benner. *Contributions to the Numerical Solution of Algebraic Riccati Equations and Related Eigenvalue Problems*. Logos-Verlag, Berlin, 1997.

[8] P. Benner and H. Faßbender. The symplectic eigenvalue problem, the butterfly form, the SR algorithm, and the Lanczos method. *Linear Algebra Appl.*, 275/276:19–47, 1998.

[9] P. Benner, H. Faßbender, and D.S. Watkins. SR and SZ algorithms for the symplectic (butterfly) eigenproblem. *Linear Algebra Appl.*, 287:41–76, 1999.

[10] E.K.-W. Chu, H.-Y. Fan, W.-W. Lin, and C.-S. Wang. A structure-preserving doubling algorithm for periodic discrete-time algebraic Riccati equations. *Int. J. Control*, 77:767–788, 2004.

columns of $\left[\begin{smallmatrix} Y_1 \\ Y_2 \end{smallmatrix}\right]$ span the desired deflating subspace of $G - \lambda H$, then $X_\diamond = -Y_2 Y_1^{-1}$ is the desired solution of (1).

The third idea is to compute the desired solution via the unique solution X_* of the DARE. The solution X_* can be found by direct application of Newton's method for DAREs [15, 16, 19, 25]. However, comparison with the basic fixed point iteration is not favorable [15, Section 5]. Therefore, this approach of solving the DARE is not considered here. Instead we will compute its solution via the stable deflating subspace of an associated matrix pencil. As R is positive definite, we can define

$$M - \lambda N = \begin{bmatrix} F^T & 0 \\ Q & I \end{bmatrix} - \lambda \begin{bmatrix} I & -R^{-1} \\ 0 & F \end{bmatrix}. \tag{4}$$

As (F, I) is controllable, (F, Q) is observable, and Q and R^{-1} are positive definite, $M - \lambda N$ has no eigenvalues on the unit circle; see, e.g., [19]. It is then easily seen that $M - \lambda N$ has precisely n eigenvalues in the open unit circle and n outside. Moreover, the Riccati solution X_* can be given in terms of the deflating subspace of $M - \lambda N$ corresponding to the n eigenvalues $\lambda_1, \ldots, \lambda_n$ inside the unit circle using the relation

$$\begin{bmatrix} F^T & 0 \\ Q & I \end{bmatrix} \begin{bmatrix} I \\ -X \end{bmatrix} = \begin{bmatrix} I & -R^{-1} \\ 0 & F \end{bmatrix} \begin{bmatrix} I \\ -X \end{bmatrix} \Lambda,$$

where $\Lambda \in \mathbb{R}^{n \times n}$ with the spectrum $\sigma(\Lambda) = \{\lambda_1, \ldots, \lambda_n\}$. Therefore, if we can compute $Y_1, Y_2 \in \mathbb{R}^{n \times n}$ such that the columns of $\left[\begin{smallmatrix} Y_1 \\ Y_2 \end{smallmatrix}\right]$ span the desired deflating subspace of $M - \lambda N$, then $X_* = -Y_2 Y_1^{-1}$ is the desired solution of the DARE (3). See, e.g., [19, 21, 25], and the references therein.

Hence, two of the ideas stated in [12] how to solve (1) can be interpreted as the numerical computation of a deflating subspace of a matrix pencil $A - \lambda B$. This is usually carried out by an iterative procedure like the QZ algorithm. Applying the numerically backward stable QZ algorithm to a matrix pencil results in a general $2n \times 2n$ matrix pencil in generalized Schur form from which the eigenvalues and deflating subspaces can be read off.

Both matrix pencils to be considered here ($G - \lambda H$ and $M - \lambda N$) have a symplectic spectrum, that is, their eigenvalues appear in reciprocal pairs λ, λ^{-1}. They have exactly n eigenvalues inside the unit circle, and n outside. Sorting the eigenvalues in the generalized Schur form such that the eigenvalues inside the unit circle are contained in the upper left $n \times n$ block, the desired deflating subspace can easily be read off and the solution $X_\diamond$, resp. X_* can be computed. (This method results in the popular generalized Schur vector method for solving DAREs [27].) Due to roundoff errors unavoidable in finite-precision arithmetic, the computed eigenvalues will in general not come in pairs $\{\lambda, \lambda^{-1}\}$, although the exact eigenvalues have this property. Even worse, small perturbations may cause eigenvalues close to the unit circle to cross the unit circle such that the number of true and computed eigenvalues inside the open unit disk may differ. Moreover,

the application of the QZ algorithm to a $2n \times 2n$ matrix pencil is computationally quite expensive. The usual initial reduction to Hessenberg-triangular form requires about $70n^3$ flops plus $24n^3$ for accumulating the Z matrix; each iteration step requires about $88n^2$ flops for the transformations and $136n^2$ flops for accumulating Z; see, e.g., [29]. An estimated $40n^3$ flops are necessary for ordering the generalized Schur form. This results in a total cost of roughly $415n^3$ flops, employing standard assumptions about convergence of the QZ iteration (see, e.g., [14, Section 7.7]).

The use of the QZ algorithm is prohibitive here not only due to the fact that it does not preserve the symplectic spectra, but also due to the costly computation. More efficient methods have been proposed which make use of the following observation: $M - \lambda N$ of the form (4) is a symplectic matrix pencil. A symplectic matrix pencil $M - \lambda N, M, N \in \mathbb{R}^{2n\times 2n}$, is defined by the property

$$MJM^T = NJN^T,$$

where

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

and I_n is the $n \times n$ identity matrix. The nonzero eigenvalues of a symplectic matrix pencil occur in reciprocal pairs: if λ is an eigenvalue of $M - \lambda N$ with left eigenvector x, then $\overline{\lambda^{-1}}$ is an eigenvalue of $M - \lambda N$ with right eigenvector $(Jx)^H$. Hence, as we are dealing with real symplectic pencils, the finite generalized eigenvalues always occur in pairs if they are real or purely imaginary or in quadruples otherwise. Although $G - \lambda H$ as in (2) is not a symplectic matrix pencil, it can be transformed into a very special symplectic pencil $\widehat{G} - \lambda\widehat{H}$ as noted in [23]. This symplectic pencil $\widehat{G} - \lambda\widehat{H}$ allows the use of a doubling algorithm to compute the solution $X_\diamond$. These methods originate from the fixed point iteration derived from the DARE. Instead of generating the usual sequence $\{X_k\}$, doubling algorithms generate $\{X_{2^k}\}$. This class of methods attracted much interest in the 1970s and 80s, see [29] and the references therein. After having been abandoned for the past decade, they have recently been revived by a series of papers, e.g. [10, 23]. To be more specific, define

$$\mathcal{N}(\widehat{G}, \widehat{H}) = \{[G_\star, H_\star] : G_\star, H_\star \in \mathbb{R}^{2n\times 2n}, \operatorname{rank}[G_\star, H_\star] = 2n, [G_\star, H_\star] \begin{bmatrix} \widehat{H} \\ -\widehat{G} \end{bmatrix} = 0\}.$$

Since $\operatorname{rank}\begin{bmatrix} \widehat{H} \\ -\widehat{G} \end{bmatrix} \leq 2n$, it follows that $\mathcal{N}(\widehat{G}, \widehat{H}) \neq \emptyset$. For any given $[G_\star, H_\star] \in \mathcal{N}(\widehat{G}, \widehat{H})$, define

$$\breve{G} = G_\star\widehat{G}, \qquad \breve{H} = H_\star\widehat{H}.$$

The transformation

$$\widehat{G} - \lambda\widehat{H} \to \breve{G} - \lambda\breve{H}$$

becomes compared to the residual *tol* obtained by the SZ algorithm. Hence, the SZ algorithm may require more arithmetic operations, but usually it generates more accurate solutions.

Example 2. In [15], the following example is considered:

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

The solution X_+ is given by

$$X_+ \approx \begin{bmatrix} 51.7993723118 & 16.0998802679 \\ 16.0998802679 & 62.2516164469 \end{bmatrix}$$

Slow convergence for the fixed point iteration was already observed in [15], after 400 iteration steps one obtains the residual norm

$$\frac{||X_{400} - Q - LX_{400}^{-1}L^T||_F}{||X_{400}||_F} = 3.78 \cdot 10^{-10},$$

and the error

$$||X_+ - X_{400}||_F = 1.64 \cdot 10^{-8},$$

since $\rho(X_+^{-1}L^T) = 0.9719$. The doubling iteration yields after 8 iterations

$$\frac{||X_\diamond - Q - LX_\diamond^{-1}L^T||_F}{||X_\diamond||_F} = 6.35 \cdot 10^{-13}$$

and

$$||X_+ - X_\diamond||_F = 7.77 \cdot 10^{-11},$$

while the SZ algorithm obtains

$$\frac{||X_* - Q - LX_*^{-1}L^T||_F}{||X_*||_F} = 1.79 \cdot 10^{-13}$$

and

$$||X_+ - X_*||_F = 6.98 \cdot 10^{-11}.$$

Hence, the doubling iteration outperforms the SZ algorithm here, but the SZ algorithm is slightly more accurate.

6 Conclusions

We have discussed several algorithms for a rational matrix equation that arises in the analysis of stationary Gaussian reciprocal processes. In particular, we have described the application of the SZ algorithm for symplectic pencils to solve this equation. Moreover, we have derived a defect correction equation that

	fixed point iteration			doubling iteration	
n	av	$it > 58$	$it > 150$	av	$\sharp(R_{SDA} > tol$
5	50.33	30	20	5.15	23
6	59.76	44	13	5.39	31
7	53.79	30	21	5.05	28
8	55.56	35	22	5.06	47
9	53.60	29	20	4.97	48
10	52.28	27	22	4.87	56
11	49.17	23	13	4.70	56
12	48.97	24	24	4.60	66
13	48.40	23	22	4.55	70
14	50.35	24	19	4.60	68
15	45.62	15	21	4.41	72
16	46.64	17	22	4.42	75
17	46.89	16	21	4.23	84
18	45.56	15	16	4.15	84
19	42.77	14	15	4.03	81
20	43.40	10	27	3.97	88
30	37.22	1	0	3.49	96
40	35.77	0	0	3.23	96
50	33.00	0	0	2.93	98
60	29.73	0	0	2.82	100

Table 2: Second set of test examples

100 examples of size $n = 5, 6, 7, \ldots, 20$ and $n = 30, 40, 50, \ldots, 60$ were generated and solved as described above. The matrices Q generated for these tests had a small norm

$$1.6 < ||Q||_2 < 405,$$

but a fairly large condition number, we allowed for

$$1 < \kappa_2(Q) < 10^{13}.$$

The fixed point iteration performed much better for these examples, but the number of iterations necessary for convergence seems to be unpredictable. The doubling iteration performs better than before, less iterations were needed for convergence. But while the iteration is run until the residual is less than $n \cdot ||Q||_F \cdot eps$, it is clearly seen here that this does not imply the same accuracy for the solution $X_\diamond$ of (1). The larger n is chosen, the worse the residual

$$R_{SDA} = \frac{||X_\diamond - Q - LX_\diamond^{-1}L^T||_F}{||X_\diamond||_F}$$

is called a doubling transformation. An important feature of this kind of transformation is that it is structure-preserving [7], eigenspace-preserving [5, 7, 24], and eigenvalue-squaring. In [23], an appropriate doubling transformation for the symplectic pencil $\widehat{G} - \lambda\widehat{H}$ is given. The resulting algorithm has very nice numerical behavior, with a quadratic converge rate, low computational cost and good numerical stability. Essentially the same algorithm was proposed in [26] using a different motivation. See Section 2.2 for more details.

Here we propose to compute the desired solution X_* via an approximate solution of the DARE (3) by the (butterfly) SZ algorithm applied to the corresponding symplectic pencil [8, 9, 11]. This algorithm is a fast, reliable and structure-preserving algorithm for computing the stable deflating subspace of the symplectic matrix pencil $M - \lambda N$ (4) associated with the DARE. The matrix pencil $M - \lambda N$ is first reduced to the so called symplectic butterfly form, which is determined by only $4n-1$ parameters. By exploiting this special reduced form and the symplecticity, the SZ algorithm is fast and efficient; in each iteration step only $O(n)$ arithmetic operations are required instead of $O(n^2)$ arithmetic operations for a QZ step. We thus save a significant amount of work. Of course, the accumulation of the Z matrix requires $O(n^2)$ arithmetic operations as in the QZ step. Moreover, by forcing the symplectic structure, the abovementioned problems of the QZ algorithm are avoided. See Section 3 for more details.

Any approximate solution $\widetilde{X}$ computed, e.g., with one of the methods described above, can be improved via defect correction. This is considered in Section 4. Finally, in Section 5 we compare the different algorithms for solving (1) discussed here.

2 Iterative Algorithms for (1)

2.1 The Fixed Point Iteration

As suggested in [12], the equation (1) can be solved directly by turning it into a fixed point iteration

$$X_{i+1} = f(X_i) = Q + LX_i^{-1}L^T \tag{5}$$

with initial condition $X_0 = Q$. In [12], it is shown that the sequence $\{X_i\}$ converges to the unique positive definite solution X_+ of (1). This convergence is robust as for any positive ϵ there exists a neighborhood Υ of X_+ such that for any initial condition $X_0 \in \Upsilon$, the sequence generated by (5) remains in a ball of radius ϵ centered in X_+ and converges to X_+. Moreover, the sequence generated by (5) converges to X_+ for any positive definite initial condition X_0 as well as for any initial condition such that $X_0 \leq -LQ^{-1}L^T$. The convergence rate is related to the spectral radius $\rho(X_+^{-1}L^T)$. The convergence is linear, but, if $\rho(X_+^{-1}L^T)$ is close to 1, the convergence may be very slow. See also [15, Section 2].

An inverse free variant of the fixed point iteration is possible. However, the algorithm is not always convergent, [15, last paragraph, Section 3].

Our implementation of the fixed point iteration first computes the Cholesky decomposition $X_i = C_i C_i^T$, next the linear system $L = C_i B_i$ is solved (that is, $B_i = LC_i^{-1}$)) and finally $X_{i+1} = Q + B_i B_i^T$ is computed. The total flop count for one iteration step is therefore $\frac{10}{3}n^3$ flops, as the first step involves about $\frac{n^3}{3}$, the second one $2n^3$ flops and the last one n^3 flops.

In many applications, rather than the solutions of matrix equations themselves, their factors (such as Cholesky or full-rank factors) are needed; see, e.g., [4, 29]. Therefore, it is desirable to use methods that compute such a factor directly without ever forming the solution explicitly. Such a method can also easily be derived based on the fixed point iteration (1). As all iterates are positive definite, it is natural here to use their Cholesky factors. Assuming we have a Cholesky factorization $X_i = Y_i Y_i^T$; then the Cholesky factor of

$$X_{i+1} = Q + LX_i^{-1}L^T = CC^T + L(Y_iY_i^T)^{-1}L^T = \left[C,\ LY_i^{-T}\right]\left[C,\ LY_i^{-T}\right]^T$$

can be obtained from the leading $n \times n$ submatrix of the L-factor of the LQ factorization of

$$\left[C,\ LY_i^{-T}\right] = \left[\begin{array}{cc} \diagdown & \square \end{array}\right]. \tag{6}$$

Note that the Q-factor is not needed as it cancels:

$$X_{i+1} = Y_{i+1}Y_{i+1}^T = L_iQ_iQ_i^TL_i^T = \left[\hat{L}_i,\ 0\right]\left[\hat{L}_i,\ 0\right]^T = \hat{L}_i\hat{L}_i^T.$$

An LQ factorization for the specially structured matrix in (6) is implemented in the SLICOT[1] subroutine `MB04JD`. Employing this, the factorized fixed point iteration yielding the sequence Y_i of Cholesky factors of X_i requires $3n^3$ flops per iteration and is thus slightly cheaper than the fixed point iteration itself. Additionally, $\frac{1}{3}n^3$ flops for the initial Cholesky factorization of Q are needed.

2.2 The Doubling Algorithm

As already observed in [12], the solution X of (1),

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

satisfies

$$G\begin{bmatrix} I \\ X \end{bmatrix} = H\begin{bmatrix} I \\ X \end{bmatrix}W \tag{7}$$

for some matrix $W \in \mathbb{R}^{n\times n}$ where

$$G = \begin{bmatrix} L^T & 0 \\ -Q & I \end{bmatrix}, \qquad H = \begin{bmatrix} 0 & I \\ L & 0 \end{bmatrix}.$$

[1] See `http://www.slicot.org`.

	fixed point iteration			doubling iteration
n	av	$it > 58$	$it > 150$	av
5	77.60	74	8	6.01
6	78.47	71	11	6.06
7	79.00	76	12	6.08
8	75.27	69	9	5.97
9	85.87	78	13	6.17
10	89.70	84	14	6.34
11	84.59	83	19	6.30
12	87.86	88	24	6.35
13	87.03	90	23	6.38
14	92.90	87	10	6.35
15	93.79	85	22	6.58
16	89.72	89	29	6.56
17	95.87	92	22	6.59
18	92.27	87	19	6.53
19	102.92	94	21	6.65
20	99.27	92	29	6.69
30	101.00	95	26	6.74
40	113.44	99	37	7.10
50	110.00	97	50	7.21
60	118.87	99	62	7.40
70	119.06	100	64	7.45
80	114.77	97	53	7.46
90	110.94	100	65	7.60
100	113.76	97	59	7.67

Table 1: First set of test examples

same (or better) accuracy than the solution X_* computed via the SZ algorithm. Therefore, for these examples, the doubling algorithm is certainly the most efficient algorithm. The matrices Q generated for these tests had a fairly small condition number

$$1 < \kappa_2(Q) < 10^5,$$

and a small norm

$$0.3 < ||Q||_2 < 1.$$

In order to generate a different set of test matrices, Q and L were constructed as follows (using MATLAB notation as before)

```
Q = triu(rand(n));
Q = Q'*Q;
L = rand(n);
```

to which the fixed point iteration is run. That is, the fixed point iteration was stopped as soon as

$$\frac{||X_{i+1} - X_i||_F}{||X_{i+1}||_F} = \frac{||X_i - Q - LX_i^{-1}L^T||_F}{||X_{i+1}||_F} < tol.$$

Hence, the fixed point iteration is run to the same accuracy as the one obtained by the SZ approach. As the fixed point approach requires about $\frac{10}{3}n^3$ arithmetic operations per iteration step, while the SZ approach requires $\frac{586}{3}n^3$ arithmetic operations, the SZ approach is cheaper if more than 58 iteration are needed in the fixed point iteration.

For the first set of examples Q and L were constructed as follows (using MATLAB notation)

```
Q = qr(rand(n));
Q = Q'*diag(rand(n,1))*Q;
L = rand(n);
```

100 examples of size $n = 5, 6, 7, \ldots, 20$ and $n = 30, 40, 50, \ldots, 100$ were generated and solved as described above. The fixed point iteration was never run for more than 150 steps. Table 1 reports how many examples of each size needed more than 58 iteration steps as well as how many examples of each size needed more than 150 iteration steps; here *it* denotes the number of iteration steps. Moreover, an average number *av* of iterations is determined, where only those examples of each size were counted which needed less than 150 iteration steps to converge. It can be clearly seen, that the larger n is chosen, the more iteration steps are required for the fixed point equation. Starting with $n = 40$ almost all examples needed more than 58 iteration steps. Hence the SZ approach is cheaper than the fixed point approach. But even for smaller n, most examples needed more than 58 iterations, the average number of iterations needed clearly exceeds 58 for all n. Hence, overall, it is cheaper to use the SZ approach.

The accuracy of the residual (17) achieved by the SZ approach was in general of the order of 10^{-12} for smaller n and 10^{-8} for larger n. But, as nonorthogonal transformation have to be used, occasionally, the accuracy can deteriorate to 10^{-3}. In that case, defect correction as described in Section 4 or the fixed point iteration with starting matrix $X_0 = X_*$ can be used to increase the accuracy of the computed solution.

Next the doubling algorithm was used to solve the same set of examples. Its iteration solves the equation (9), the desired solution $X_\diamond$ is obtained from the computed solution via (11). The iteration was run until the residuum was less than $n \cdot ||Q||_F \cdot eps$, where *eps* is MATLAB's machine epsilon. This does not imply the same accuracy for the solution $X_\diamond$ of (1). Due to the back substitution (11), the final solution $X_\diamond$ may have a larger residual error. For these examples, only about 7 iterations where needed to determine an $X_\diamond$ which has about the

Hence, the desired solution X can be computed via an appropriate deflating subspace of $G - \lambda H$. This could be done by employing the QZ algorithm. But the following idea suggested in [23] achieved a much faster algorithm.

Assume that X is the unique symmetric positive definite solution of (1). Then it satisfies (7) with $W = X^{-1}L^T$. Let

$$\widehat{L} = LQ^{-1}L, \quad \widehat{Q} = Q + LQ^{-1}L^T, \quad \widehat{P} = L^TQ^{-1}L,$$

and

$$\widehat{X} = X + \widehat{P}.$$

Then it follows that

$$\widehat{G}\begin{bmatrix} I \\ \widehat{X} \end{bmatrix} = \widehat{H}\begin{bmatrix} I \\ \widehat{X} \end{bmatrix} W^2, \tag{8}$$

where

$$\widehat{G} = \begin{bmatrix} \widehat{L}^T & 0 \\ \widehat{Q} + \widehat{P} & -I \end{bmatrix}, \qquad \widehat{H} = \begin{bmatrix} 0 & I \\ \widehat{L} & 0 \end{bmatrix}.$$

The pencil $\widehat{G} - \lambda\widehat{H}$ is symplectic as $\widehat{G}J\widehat{G}^T = \widehat{H}J\widehat{H}^T$. (As G and H are not symplectic themselves, the butterfly SZ algorithm described in the next section can not be employed directly in order to computed the desired deflating subspace of $\widehat{G} - \lambda\widehat{H}$.) It is easy to see that $\widehat{X}$ satisfies (8) if and only if the equation

$$\widehat{X} = (\widehat{Q} + \widehat{P}) - \widehat{L}\widehat{X}^{-1}\widehat{L}^T \tag{9}$$

has a symmetric positive definite solution $\widehat{X}$.

In [23], it is suggest to use a doubling algorithm to compute the solution $\widehat{X}$ of (8). An appropriate doubling transformation for the symplectic pencil (8) is given. Applying this special doubling transformation repeatedly the following structure-preserving doubling algorithm (SDA) arises:

$$\begin{aligned} &\text{for } i = 0, 1, 2, \ldots \\ &\qquad L_{i+1} = L_i(Q_i - P_i)^{-T}L_i \\ &\qquad Q_{i+1} = Q_i - L_i(Q_i - P_i)^{-1}L_i^T \\ &\qquad P_{i+1} = P_i + L_i^T(Q_i - P_i)^{-1}L_i \\ &\text{until convergence} \end{aligned} \tag{10}$$

with

$$L_0 = \widehat{L}, \quad Q_0 = \widehat{Q} + \widehat{P}, \quad P_0 = 0.$$

As the matrix $Q_i - P_i$ is positive definite for all i [23], the iterations above are all well defined. The sequence Q_{i+1} will converge to $\widehat{X}$. Thus, the unique symmetric positive definite solution to (1) can be obtained by computing

$$X_\diamond = \widehat{X} - \widehat{P}. \tag{11}$$

Essentially the same algorithm was proposed in [26] using a different motivation.

Both papers [23, 26] point out that this algorithm has very nice numerical behavior, with a quadratic converge rate, low computational cost and good numerical stability. The convergence statement proven in [23] requires that the spectral radius $\rho(\widehat{X}^{-1}\widehat{L}^T)$ is strictly less than 1. Though this is not shown in [23], it can be proved.

Lemma 2.1 *With the notation introduced above, we have*

$$\rho(\widehat{X}^{-1}\widehat{L}^T) < 1.$$

Proof. As

$$\widehat{X}^{-1}\widehat{L}^T = (X+\widehat{P})^{-1}LQ^{-1}L = (X+L^TQ^{-1}L)^{-1}LQ^{-1}L$$

we have with (7) ($Q = X - LW, W = X^{-1}L^T$)

$$\begin{aligned}
\widehat{L}^{-T}\widehat{X} &= L^{-1}QL^{-1}(X+L^TQ^{-1}L)\\
&= L^{-1}QL^{-1}(XL^{-1}QL^{-1}+L^TQ^{-1}LL^{-1}QL^{-1})LQ^{-1}L\\
&= L^{-1}QL^{-1}(W^{-T}QL^{-1}+L^TL^{-1})LQ^{-1}L\\
&= L^{-1}QL^{-1}(W^{-T}(X-LW)L^{-1}+L^TL^{-1})LQ^{-1}L\\
&= L^{-1}QL^{-1}(W^{-T}XL^{-1}-W^{-T}LWL^{-1}+L^TL^{-1})LQ^{-1}L\\
&= L^{-1}QL^{-1}((W^{-T})^2-XL^{-1}LX^{-1}L^TL^{-1}+L^TL^{-1})LQ^{-1}L\\
&= L^{-1}QL^{-1}((W^{-T})^2-L^TL^{-1}+L^TL^{-1})LQ^{-1}L\\
&= L^{-1}QL^{-1}(W^{-T})^2LQ^{-1}L.
\end{aligned}$$

Therefore,

$$\widehat{X}^{-1}\widehat{L}^T = L^{-1}QL^{-1}(W^T)^2LQ^{-1}L,$$

and as $\rho(X^{-1}L^T) < 1$,

$$\rho(\widehat{X}^{-1}\widehat{L}^T) = \rho(W^2) = \rho\left((X^{-1}L^T)^2\right) < 1. \qquad \square$$

This algorithm requires about $8n^3$ arithmetic operations per iteration step when implemented as follows: first a Cholesky decomposition of $Q_i - P_i = C_i^TC_i$ is computed ($\frac{1}{3}n^3$ arithmetic operations), then $L_i^TC_i^{-1}$ and $C_i^{-T}L_i^T$ are computed (both steps require $2n^3$ arithmetic operations), finally $L_{i+1}, Q_{i+1}, P_{i+1}$ are computed using these products ($4n^3$ arithmetic operations if the symmetry of Q_{i+1} and P_{i+1} is exploited). Hence, one iteration step requires $\frac{25}{3}n^3$ arithmetic operations.

Despite the fact that a factorized version of the doubling iteration for DAREs has been around for about 30 years, see [29] and the references therein, the SDA (10) for (1) can not easily be rewritten to work on a Cholesky factor of Q_i due to the minus sign in the definition of the Q_i's.

would be roughly 12 times that for the basic fixed point iteration. But a more efficient algorithm which makes use of the special structure of (16) can be easily devised: first, note that (16) looks very similar to a Stein (discrete Lyapunov) equation. The only difference is the sign in front of $\widetilde{E}$. With this observation and a careful inspection of the Bartels-Stewart type algorithm for Stein equations suggested in [6] and implemented in the SLICOT Basic Control Toolbox[2] function `slstei` (see also [30]), equation (16) can be solved efficiently with this algorithm when only a few signs are changed. This method requires less than 10 times the cost for one fixed point iteration.

5 Numerical Experiments

Numerical experiments were performed in order to compare the three different approaches for solving (1) discussed here. All algorithms were implemented in MATLAB Version 7.2.0.232 (R2006a) and run on an Intel Pentium M processor.

In particular, we implemented

- the fixed point iteration as described in Section 2.1 which requires $\frac{10}{3}n^3$ arithmetic operations per iteration,
- the doubling algorithm as described in Section 2.2 which requires $\frac{25}{3}n^3$ arithmetic operations per iteration,
- the SZ algorithm as described in Section 3 which requires $\frac{586}{3}n^3$ arithmetic operations.

Slow convergence of the fixed point iteration has been observed in, e.g., [12, 15]. The convergence rate depends on the spectral radius $\rho(X_+L^{-T})$. One iteration of the doubling algorithm costs as much as 2.5 iterations of the fixed point iteration. In [23], no numerical examples are presented, in [26] only one example is given (see Example 2 below) in which the doubling algorithm is much faster than the fixed point iteration. Our numerical experiments confirm that this is so in general. The SZ algorithm costs as much as 59 iterations of the fixed point iteration and as much as 23 iterations of the doubling algorithm.

Example 1. First, the fixed point equation approach as described in Section 2.1 was compared to the SZ approach as described in Section 3. For this, each example was first solved via the SZ approach. The so computed solution X_* was used to determined the tolerance tol

$$tol = \frac{||X_* - Q - LX_*^{-1}L^T||_F}{||X_*||_F} \tag{17}$$

[2] See `http://www.slicot.org`.

Assume that $||E|| < 1/||\widetilde{X}^{-1}||$. Then we have $||E\widetilde{X}^{-1}|| < 1$. Using the Neumann series [14, Lemma 2.3.3] yields

$$\begin{aligned}\widetilde{X} &= E + Q + L\widetilde{X}^{-1}(I + E\widetilde{X}^{-1} + (E\widetilde{X}^{-1})^2 + \ldots)L^T \\ &= E + Q + L\widetilde{X}^{-1}L^T + L\widetilde{X}^{-1}E\widetilde{X}^{-1}L^T + L\widetilde{X}^{-1}(E\widetilde{X}^{-1})^2L^T + \ldots \\ &= E + Q + L\widetilde{X}^{-1}L^T + L\widetilde{X}^{-1}E\widetilde{X}^{-1}L^T + L\widetilde{X}^{-1}E\widetilde{X}^{-1}E\widetilde{X}^{-1}L^T + \ldots \\ &= E + Q + L\widetilde{X}^{-1}L^T + \widetilde{L}E\widetilde{L}^T + \widetilde{L}E\widetilde{X}^{-1}E\widetilde{L}^T + \ldots\end{aligned}$$

where

$$\widetilde{L} = L\widetilde{X}^{-1}.$$

With the residual

$$R(\widetilde{X}) = \widetilde{X} - Q - L\widetilde{X}^{-1}L^T,$$

we thus have $R(\widetilde{X}) \approx E + \widetilde{L}E\widetilde{L}^T$. By dropping terms of order $O(||E||^2)$, we obtain the defect correction equation

$$R(\widetilde{X}) = \widetilde{E} + \widetilde{L}\widetilde{E}\widetilde{L}^T. \tag{16}$$

Hence, the approximate solution $\widetilde{X}$ can be improved by solving (16) for $\widetilde{E}$. The improved $\widehat{X}$ is then given by $\widehat{X} = \widetilde{X} - \widetilde{E}$.

Lemma 4.1 *Equation (16) has a unique solution if* $\rho(\widetilde{L}) = \rho(L\widetilde{X}^{-1}) < 1$.

Proof. Note that (16) is equivalent to the linear system of equations

$$\left(I_{n^2} + \widetilde{L}^T \otimes \widetilde{L}^T\right) \mathrm{vec}(\widetilde{E}) = \mathrm{vec}(R(\widetilde{X})),$$

where $\otimes$ denotes the Kronecker product and $\mathrm{vec}(A) = [a_{11}, \ldots, a_{n1}, a_{12}, \ldots, a_{n2}, \ldots, a_{1n}, \ldots, a_{nn}]^T$ is the vector that consists of the columns of $A = [a_{ij}]_{i,j=1}^n$ stacked on top of each other from left to right [17, Section 4.2]. As $\rho(\widetilde{L}) < 1$, the assertion follows from $\sigma(I_{n^2} + \widetilde{L}^T \otimes \widetilde{L}^T) = 1 + \sigma(\widetilde{L}^2)$. □

Note that Lemma 4.1 also follows from a more general existence result for linear matrix equations given in [28, Proposition 3.1].

In [15], essentially the same defect correction was derived by applying Newton's method to (1). Written in the notation used here, the defect correction equation derived in [15] reads

$$\widetilde{X} - Q + \widetilde{L}\widetilde{X}\widetilde{L}^T = E + \widetilde{L}E\widetilde{L}^T + 2\widetilde{L}L^T.$$

It is easy to see that this is equivalent to (16). In [15], it is suggested to solve the defect correction equation with a general Sylvester equation solver as in [13]. In that case, the computational work for solving the defect correction equation

3 The butterfly SZ algorithm

As shown in [12], instead of solving the equation (1) one can solve the related DARE (3),

$$\mathcal{DR}(X) = Q + FXF^T - FX(X+R)^{-1}XF^T - X.$$

One approach to solve this equation is via computing the stable deflating subspace of the matrix pencil from (4), i.e.,

$$M - \lambda N = \begin{bmatrix} F^T & 0 \\ Q & I \end{bmatrix} - \lambda \begin{bmatrix} I & -R^{-1} \\ 0 & F \end{bmatrix}.$$

Here we propose to use the butterfly SZ algorithm for computing the deflating subspace of $M - \lambda N$. The butterfly SZ algorithm [9, 11] is a fast, reliable and efficient algorithm especially designed for solving the symplectic eigenproblem for a symplectic matrix pencil $\widetilde{M} - \lambda\widetilde{N}$ in which both matrices are symplectic; that is $\widetilde{M}J\widetilde{M}^T = \widetilde{N}J\widetilde{N}^T = J$. The above symplectic matrix pencil

$$\begin{bmatrix} F^T & 0 \\ Q & I \end{bmatrix} - \lambda \begin{bmatrix} I & -R^{-1} \\ 0 & F \end{bmatrix} = \begin{bmatrix} L^{-1}L^T & 0 \\ Q & I \end{bmatrix} - \lambda \begin{bmatrix} I & -L^{-1}QL^{-T} \\ 0 & LL^{-T} \end{bmatrix}$$

can be rewritten (after premultiplying by $\left[\begin{smallmatrix} L & 0 \\ 0 & L^{-1} \end{smallmatrix}\right]$) as

$$\widetilde{M} - \lambda\widetilde{N} = \begin{bmatrix} L^T & 0 \\ L^{-1}Q & L^{-1} \end{bmatrix} - \lambda \begin{bmatrix} L & -QL^{-T} \\ 0 & L^{-T} \end{bmatrix}, \tag{12}$$

where both matrices $\widetilde{M} = \widetilde{N}^T$ are symplectic. In [9, 11] it is shown that for the symplectic matrix pencil $\widetilde{M} - \lambda\widetilde{N}$ there exist numerous symplectic matrices Z and nonsingular matrices S which reduce $\widetilde{M} - \lambda\widetilde{N}$ to a symplectic butterfly pencil $A - \lambda B$:

$$S(\widetilde{M} - \lambda\widetilde{N})Z = A - \lambda B = \begin{bmatrix} C & D \\ 0 & C^{-1} \end{bmatrix} - \lambda \begin{bmatrix} 0 & -I \\ I & T \end{bmatrix}, \tag{13}$$

where C and D are diagonal matrices, and T is a symmetric tridiagonal matrix. (More generally, not only the symplectic matrix pencil in (12), but any symplectic matrix pencil $\widetilde{M} - \lambda\widetilde{N}$ with symplectic matrices $\widetilde{M}, \widetilde{N}$ can be reduced to a symplectic butterfly pencil). This form is determined by just $4n - 1$ parameters. The symplectic matrix pencil $A - \lambda B$ is called a symplectic butterfly pencil. If T is an unreduced tridiagonal matrix, then the butterfly pencil is called unreduced. If any of the $n - 1$ subdiagonal elements of T are zero, the problem can be split into at least two problems of smaller dimension, but with the same symplectic butterfly structure.

Once the reduction to a symplectic butterfly pencil is achieved, the SZ algorithm is a suitable tool for computing the eigenvalues/deflating subspaces of

the symplectic pencil $A-\lambda B$ [9, 11]. The SZ algorithm preserves the symplectic butterfly form in its iterations. It is the analogue of the SR algorithm (see [8, 11]) for the generalized eigenproblem, just as the QZ algorithm is the analogue of the QR algorithm for the generalized eigenproblem. Both are instances of the GZ algorithm [32].

Each iteration step begins with an unreduced butterfly pencil $A-\lambda B$. Choose a spectral transformation function q and compute a symplectic matrix $\breve{Z}$ such that

$$\breve{Z}^{-1}q(A^{-1}B)e_1 = \alpha e_1$$

for some scalar α. Then transform the pencil to

$$\widetilde{A} - \lambda\widetilde{B} = (A - \lambda B)\breve{Z}.$$

This introduces a bulge into the matrices $\widetilde{A}$ and $\widetilde{B}$. Now transform the pencil to

$$\widehat{A} - \lambda\widehat{B} = S^{-1}(\widetilde{A} - \lambda\widetilde{B})\widetilde{Z},$$

where $\widehat{A} - \lambda\widehat{B}$ is again of symplectic butterfly form. S and $\widetilde{Z}$ are symplectic, and $\widetilde{Z}e_1 = e_1$. This concludes the iteration. Under certain assumptions, it can be shown that the butterfly SZ algorithm converges cubically. The needed assumptions are technically involved and follow from the GZ convergence theory developed in [32]. The convergence theorem says roughly that if the eigenvalues are separated, and the shifts converge, and the condition numbers of the accumulated transformation matrices remain bounded, then the SZ algorithm converges. For a detailed discussion of the butterfly SZ algorithm see [9, 11].

Hence, in order to compute an approximate solution of the DARE (3) by the butterfly SZ algorithm, first the symplectic matrix pencil $\widetilde{M} - \lambda\widetilde{N}$ as in (12) has to be formed, then the symplectic matrix pencil $A - \lambda B$ as in (13) is computed. That is, symplectic matrices Z_0 and S_0 are computed such that

$$A - \lambda B := S_0^{-1}\widetilde{M}Z_0 - \lambda S_0^{-1}\widetilde{N}Z_0$$

is a symplectic butterfly pencil. Using the butterfly SZ algorithm, symplectic matrices Z_1 and S_1 are computed such that

$$S_1^{-1}AZ_1 - \lambda S_1^{-1}BZ_1$$

is a symplectic butterfly pencil and the symmetric tridiagonal matrix T in the lower right block of $S_1^{-1}BZ_1$ is reduced to quasi-diagonal form with 1×1 and 2×2 blocks on the diagonal. The eigenproblem decouples into a number of simple 2×2 or 4×4 generalized symplectic eigenproblems. Solving these subproblems, finally symplectic matrices Z_2, S_2 are computed such that

$$\begin{aligned}
\breve{A} &= S_2^{-1}S_1^{-1}AZ_1Z_2 = \begin{bmatrix} \phi_{11} & \phi_{12} \\ 0 & \phi_{22} \end{bmatrix}, \\
\breve{B} &= S_2^{-1}S_1^{-1}BZ_1Z_2 = \begin{bmatrix} \psi_{11} & \psi_{12} \\ 0 & \psi_{22} \end{bmatrix},
\end{aligned}$$

where the eigenvalues of the matrix pencil $\phi_{11} - \lambda\psi_{11}$ are precisely the n stable generalized eigenvalues. Let $Z = Z_0Z_1Z_2$. Partitioning Z conformably,

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}, \tag{14}$$

the Riccati solution X_* is found by solving a system of linear equations:

$$X_* = -Z_{21}Z_{11}^{-1}. \tag{15}$$

This algorithm requires about $195n^3$ arithmetic operations in order to compute the solution of the Riccati equation (and is therefore cheaper than the QZ algorithm which requires about $422n^3$ arithmetic operations). The cost of the different steps of the approach described above are given as follows. The computation of $L^{-1}Q$ and L^{-1} using an LU decomposition of L requires about $\frac{14}{3}n^3$ arithmetic operations. A careful flop count reveals that the initial reduction of $\widetilde{M}-\lambda\widetilde{N}$ to butterfly form $A-\lambda B$ requires about $75n^3$ arithmetic operations. For computing Z_0, an additional $28n^3$ arithmetic operations are needed. The butterfly SZ algorithm requires about $O(n^2)$ arithmetic operations for the computation of $\breve{A} - \lambda\breve{B}$ and additional $85n^3$ arithmetic operations for the computation of Z (this estimate is based on the assumption that $\frac{2}{3}$ iterations per eigenvalue are necessary as observed in [9]). The solution of the final linear system requires $\frac{14}{3}n^3$ arithmetic operations. Hence, the entire algorithm described above requires about $\frac{586}{3}n^3$ arithmetic operations.

However, it should be noted that in the SZ algorithm non-orthogonal equivalence transformations have to be used. These are not as numerically stable as the orthogonal transformations used by the QZ algorithm. Therefore, the approximate DARE solution computed by the SZ algorithm is sometimes less accurate than the one obtained from using the QZ algorithm. A possibility to improve the computed solution is defect correction is discussed in the next section.

4 Defect Correction

Any approximate solution $\widetilde{X}$ computed, e.g., with one of the methods described above, can be improved via defect correction. Let

$$\widetilde{X} = X + E$$

where X is the exact solution of (1), $X = Q + LX^{-1}L^T$. Then

$$\begin{aligned}
\widetilde{X} &= E + Q + LX^{-1}L^T \\
&= E + Q + L(\widetilde{X} - E)^{-1}L^T \\
&= E + Q + L((I - E\widetilde{X}^{-1})\widetilde{X})^{-1}L^T \\
&= E + Q + L\widetilde{X}^{-1}(I - E\widetilde{X}^{-1})^{-1}L^T.
\end{aligned}$$