Accelerating Newton's Method for Discrete-Time Algebraic Riccati Equations

Peter Benner
Fachbereich 3/Mathematik und Informatik
Universität Bremen
28334 Bremen (Germany)
benner@math.uni-bremen.de

Keywords: Newton's method, discrete-time algebraic Riccati equation, line search, Armijo rule.

Abstract

This paper studies Newton's method for solving discrete-time algebraic Riccati equations. We will modify this method by using line searches along the Newton directions. Numerical experiments show that the new method almost always saves some iterations compared to Newton's method. Iteration steps of the line search method are shown to be only slightly more expensive than standard Newton steps. Hence we conclude that the new method is usually faster than Newton's method and often even outperforms standard methods based on invariant subspace computations.

1 Introduction

We consider the numerical solution of the *generalized discrete–time algebraic Riccati equation* (DARE)

$$0 = Q + A^T XA - E^T XE - (A^T XB + S)K(X) =: \mathcal{R}(X), \tag{1}$$

where $K(X) := (R+B^T XB)^{-1}(B^T XA+S^T)$, $A, E, Q \in \mathbb{R}^{n\times n}$, $B, S \in \mathbb{R}^{n\times m}$, $R \in \mathbb{R}^{m\times m}$, and $X \in \mathbb{R}^{n\times n}$ is the sought-after solution matrix.

We will use the notation $M > 0$ ($M \geq 0$) for a positive (semi-)definite square matrix M. An arbitrary matrix norm is denoted by $\|M\|$ while $\|M\|_F = \mathrm{trace}\left(M^T M\right)^{1/2}$ is the Frobenius norm.

The following assumptions will be used in this paper:

(i) E is nonsingular.

(ii) $(E^{-1}A, E^{-1}B)$ is d-stabilizable.

(iii) $Q = Q^T$, $R = R^T$.

(iv) A stabilizing solution X_d (i.e., $E - \lambda(A - BK(X_d))$ has all its eigenvalues in the open unit disk), of (1) exists, is unique, and $R + B^T X_d B > 0$.

Note that we do not assume $Q \geq 0$, $R \geq 0$, or d-detectability. Sufficient conditions for (iv) to hold can be found, e.g., in [9]. The DARE (1) is a well-defined matrix equation as long as $R + B^T X_d B$ is nonsingular. Note that this does not require R to be nonsingular.

The methods presented here will be given in terms of the equation (1). By inverting E, (1) can be reduced to the case $E = I$. However, this introduces unnecessary rounding errors and, if E is ill-conditioned, even numerical instabilities. Therefore, the algorithm derived here avoids inverting E.

2 Newton's Method for the DARE

Newton's method for iteratively solving DAREs was proposed in [7] and extended to the generalized equation as given in (1) in [2]. A discussion of the convergence properties can be found in [10, 9]. The resulting algorithm can be formulated as follows.

Algorithm 1 ***[Newton's Method for the DARE]***
Input: *The coefficient matrices A, B, E, Q, R, S of (1); a stabilizing starting guess X_0.*
Output: *X_{k+1}, an approximate solution of (1); N_k, an estimate for $X_d - X_{k+1}$.*

FOR $k = 0, 1, 2, \ldots$ "until convergence"
1. $A_k \leftarrow A - BK(X_k)$.
2. Solve for N_k in the Stein equation
$$A_k^T N_k A_k - E^T N_k E = -\mathcal{R}(X_k). \tag{2}$$
3. $X_{k+1} \leftarrow X_k + N_k$.

END FOR

We have the following result for Algorithm 1 [7, 10, 9].

Theorem 1 *If the assumptions (i)–(iv) hold, and X_0 is stabilizing, then the iterates of Algorithm 1 satisfy:*

a) All iterates X_k are stabilizing.

b) $X_d \leq \ldots \leq X_{k+1} \leq X_k \leq \ldots \leq X_1$.

c) $\lim_{k\to\infty} X_k = X_d$.

d) There exists a constant $\gamma > 0$ such that for all $k \geq 1$, $\|X_{k+1} - X_d\| \leq \gamma \|X_k - X_d\|^2$, i.e., the X_k converge globally quadratic to X_d.

Proofs for the results collected in Theorem 1 in case $E = I$ and R nonsingular are given in [9]. Although not mentioned in [9], the proofs also hold without the assumption of R being nonsingular. The above theorem is then a trivial corollary of the results in [9] using the equivalence of (1) to the standard DARE with $E = I_n$; see [3].

The computational cost for Algorithm 1 mainly depends upon the cost for the numerical solution of the Stein equation (2). A detailed flop count reveals that for $n \approx m$, solving the DARE (1) by the Schur vector method as proposed in [2, 11], the computational cost is equivalent to 5–6 Newton steps. In case $E = I$, the solution of (2) becomes significantly cheaper while the cost for the Schur vector method remains essentially the same. Hence, in this case, Newton's method is competitive if it requires at most 15 or 16 iterations.

The main difficulty is to find a "good" stabilizing initial guess X_0. There exist stabilization procedures for discrete-time linear systems (see, e.g., [1, 8]). But these may give large initial errors $\|X_d - X_0\|$.

Despite the ultimate rapid convergence indicated by Theorem 1 d), the iteration may initially converge slowly. This can be due to a large initial error $\|X_d - X_0\|$ or a disastrous first Newton step resulting in a large error $\|X_d - X_1\|$. In both cases, usually many iterations are required to find the region of rapid convergence. For these reasons, Newton's method is usually not used by itself to solve DAREs. However, when it is used for iterative refinement of an approximate solution obtained by some other method, it is often able to squeeze out the maximum possible accuracy [10].

From the point of view of optimization theory, the Newton step gives a search direction along which the next residual may be minimized. A disastrous first step may then be considered as a too long step, whereas the initial slow but monotonic convergence suggests that one could take longer steps in that direction. These observations lead to the line search method discussed below.

3 Step Size Control by Line Search

The idea used here is to replace Step 3 of Algorithm 1 by $X_{k+1} = X_k + t_k N_k$, where t_k is a real scalar controlling the "length" of the step tN_k. Ideally, we would choose t_k such that it minimizes $\|\mathcal{R}(X_k + tN_k)\|_F$ in every step. Unfortunately, the dependence of the expression $\|\mathcal{R}(X_k + tN_k)\|_F$ upon t is not polynomial like in the continuous-time case [5]. Thus, it will in general be very difficult to solve the minimization problem

$$\min_t \|\mathcal{R}(X_k + tN_k)\|_F^2 = \operatorname{trace}\left(\mathcal{R}(X_k + tN_k)^2\right) \qquad (3)$$

exactly. The minimization problem (3) requires to minimize a rational function. As line searches should not significantly increase the cost of one iteration step, it will therefore not be reasonable to try to solve (3) exactly. Our approach here is to use a Taylor approximation of the non-polynomial part $(R + B^T(X_k + tN_k)B)^{-1}$.

We will use the notation $R_k := R + B^T X_k B$ for $k = 0, 1, \ldots$. Throughout this section we will assume R_k to be nonsingular for all k. We will see that under certain circumstances, it can be proved that this assumption holds. We can then further define $G_k := BR_k^{-1}B^T$.

After some tedious calculations we obtain (see [3])

$$\begin{aligned} \mathcal{R}(X_k + tN_k) = (1-t)\mathcal{R}(X_k) \ - \\ - \ t^2 A_k^T N_k (I + tG_k N_k)^{-1} G_k N_k A_k. \end{aligned} \qquad (4)$$

Now we replace $g_k(t) := (I + tG_kN_k)^{-1}$ by its Taylor series at 0,

$$T_k(t) \ := \ I - tG_kN_k + t^2(G_kN_k)^2 + O(t^3). \qquad (5)$$

The standard result for convergence of the Neumann series yields a sufficient condition for pointwise convergence of the Taylor series $T_k(t)$ [3].

Lemma 1 *If for any submultiplicative norm $\|\cdot\|$, $t \in \mathbb{R}$ satisfies*

$$|t| < \|G_kN_k\|^{-1}, \qquad (6)$$

then the Taylor series $T_k(t)$ converges to $g_k(t)$.

We also obtain a sufficient condition for nonsingularity of the R_k's [3].

Lemma 2 *If t_k satisfies (6) where $\|\cdot\|$ is any submultiplicative norm, and R_k is nonsingular, then $R_{k+1} = R + B^T(X_k + t_kN_k)B$ is nonsingular.*

Thus, we can expect our approach to give reasonable results if condition (6) is satisfied. Close to the solution X_d we can expect N_k to be small and therefore, the assumption (6) will then allow for large t. In the following we will see that the t_k are chosen from $[0, 2]$ such that (6) is satisfied for all t_k as long as

$$\|G_kN_k\|_F < 1/2. \qquad (7)$$

To simplify notation, we introduce

$$V_k := \big(A - BK(X_k)\big)^T N_k G_k N_k \big(A - BK(X_k)\big). \qquad (8)$$

Assuming convergence of the Taylor series, from (4) and (8) we obtain as an expression for the residual of the DARE

$$\mathcal{R}(X_k + tN_k) = (1-t)\mathcal{R}(X_k) - t^2 V_k + O(t^3). \qquad (9)$$

Now let $f_k(t) := \|(1-t)\mathcal{R}(X_k) - t^2V_k\|_F^2$. We suggest instead of minimizing $\|\mathcal{R}(X_k + tN_k)\|_F$ to solve the minimization problem

$$\min_t \|f_k(t)\|_F. \qquad (10)$$

For the minimizing t_k we then have

$$\|\mathcal{R}(X_k + t_kN_k)\|_F^2 \leq \|f_k(t_k)\|_F + O\left(\|t_kG_kN_k\|_F^3\right). \qquad (11)$$

As $f_k(t)$ is a polynomial of degree at most four having the same structure as in the continuous-time case [3, 5], we will not discuss the details of solving (10) here. We have the following result yielding the search interval for the step sizes [3, 5].

Proposition 1 *The function f_k has a local minimum at some value in $[0, 2]$. Moreover, t_k can be chosen such that $f_k(t_k) = \min_{t\in[0,2]} f_k(t)$.*

The proof of Proposition 1 in [3, 5] also shows that

$$f_k(t_k) < f_k(0) = \|\mathcal{R}(X_k)\|_F^2 \qquad (12)$$

unless $\mathcal{R}(X_k) = 0$, i.e., unless X_k is a solution of the DARE (1).

Choosing t_k from the interval $[0, 2]$, the convergence of the Taylor series $T_k(t)$ to $g_k(t)$ is guaranteed if (7) is satisfied.

The additional cost for the line search procedure implied by computing t_k via (10) comes mainly from forming V_k. Exploiting the symmetry of V_k and using some intermediate computations required for forming A_k and $\mathcal{R}(X_k)$, this amounts to $5n^2m$ flops for $n = m$ [3]. If $m \ll n$, the additional cost is negligible compared to the $O(n^3)$ flops required for solving (2). Even if $m = n$, the additional cost does raise the cost of an iteration step only by $\approx 5\%$, using the flop counts for Algorithm 1 derived in the last section.

4 Convergence

We can expect the approximate line search procedure derived in the last section to yield reasonable results only if (6) holds. Numerous tests with randomly generated data show that often with large $\|G_kN_k\|_F$ (and hence, large $\|V_k\|_F$), the computed step sizes are small and no (or only very limited) progress is made towards a solution of (1). An explanation for this behaviour can be given as follows. If the computed V_k is large relative to $\|\mathcal{R}(X_k)\|_F$, then f_k behaves essentially like t_k^4. The computed local minimum is therefore close to zero. But as in that case, the convergence criterion (7) is usually not satisfied, the computed local minimizer can be far from an exact minimizer of $\|\mathcal{R}(X_k + tN_k)\|_F$. Moreover, a very small step size will not move the iterate far and so we obtain similar behaviour in the next step.

Therefore we suggest a modified approach which removes the possible drawbacks of approximate line search.

The *Armijo Rule* (see, e.g, [6, Section 6.1])

$$\|\mathcal{R}(X_k + t_kN_k)\|_F \le \sqrt{1-\alpha t_k}\,\|\mathcal{R}(X_k)\|_F, \qquad (13)$$

where α is a small constant, guarantees that in each step, the residual decreases "sufficiently". If the t_k are bounded from below by some constant $t_L > 0$ and $t_k \le 2$ for all k as suggested by Proposition 1, then by choosing $\alpha < 1/2$, there exists a constant $\gamma < 1$ such that $\sqrt{1-\alpha t_k} < \gamma$ for all k. Hence,

$$\|\mathcal{R}(X_k + t_kN_k)\|_F < \gamma^k \|\mathcal{R}(X_0)\|_F$$

such that $\lim_{k\to\infty} \mathcal{R}(X_k) = 0$.

If neither the approximate minimizer obtained from (10) nor a Newton step satisfy the Armijo rule, then the step size is computed via a standard backtracking approach as suggested in [6] and worked out for the problem considered here in [3].

If very small step sizes are chosen, the constant $\sqrt{1-\alpha t_k}$ will be very close to one such that very small decreases of the residual norms will be sufficient to satisfy (13). This may lead to a stagnation of the algorithm. On the other hand, performing a Newton step may increase the residual substantially, but subsequent Newton steps often decrease the residual much faster then the tiny steps chosen by approximate line search or backtracking. In this case, it is also advisable to "restart" the algorithm by performing a Newton step. As a criterion to escape stagnation we use

$$\|\mathcal{R}(X_k + t_kN_k)\|_F < tol_S \|\mathcal{R}(X_{k-k_B})\|_F. \qquad (14)$$

Collecting all the above considerations leads to the following line search procedure.

1. *Compute t_k by (10).*
2. *$r_1 := \|\mathcal{R}(X_k + t_kN_k)\|_F$.*
3. *$r_2 := \|\mathcal{R}(X_k + N_k)\|_F$. If $r_1 > r_2$, set $t_k := 1$.*
4. *WHILE $\|\mathcal{R}(X_k + t_kN_k)\|_F > \sqrt{1-\alpha t_k}\,\|\mathcal{R}(X_k)\|_F$*
   choose new $\hat{t}_k$ by backtracking
   END WHILE
5. *If no t_k satisfying (13) can be found, or if (14) is not satisfied, perform a Newton step.*

The above procedure uses all the advantages provided by Newton's method, the approximate line search, and backtracking. Eventually, the above procedure will always take Newton steps and therefore will ultimately converge quadratically to the solution. Once the region of rapid convergence has been reached, it is unlikely that any line search will improve the convergence significantly. Therefore, we can include another criterion which signals that the region of rapid convergence has been reached, e.g.,

$$\|\mathcal{R}(X_k + N_k)\|_F < tol_R \|\mathcal{R}(X_k)\|_F. \qquad (15)$$

The required solution of the DARE is stabilizing. Therefore, in order to have a complete convergence theory of our line search algorithm, we also need that the iterates converge to the *stabilizing* solution X_d. This can be ensured by choosing all iterates X_k to be stabilizing. Unfortunately, so far there is no proof that starting with a stabilizing initial guess X_0, all iterates obtained by the line search procedure proposed here are also stabilizing. During all our numerical tests, all iterates were stabilizing. This topic requires further investigations.

5 Numerical Examples

Newton's method (Algorithm 1) and Newton's method with line search based on the above considerations were implemented as MATLAB[1] functions using $\alpha = 0.2$, $tol_R = 10^{-4}$, $k_B = 2$, $tol_S = 0.9$. All computations were done under MATLAB Version 4.2c on Hewlett Packard Apollo series 700 computers using IEEE double precision arithmetic with machine precision $\varepsilon \approx 2.22 \times 10^{-16}$.

Example 1 We tested both algorithms for all examples of the benchmark collection [4] (except Example 4 for which the assumptions (i)–(iv) do not hold). For each example we tried the stabilization methods from [8] and [1] in order to obtain a stabilizing initial guess. For d-stable systems, we also tried $X_0 = 0$.

As observed in Section 2, Newton's method used as a direct solver for DAREs (1) with $E = I$ is competitive with the Schur vector approach if it requires up to about 15 iterations. For all benchmark examples, $E = I$. Using the default settings given in [4] we made the following observations:

- in the cases where $X_0 = 0$ was a stabilizing initial guess, Newton's method and line search never required more than eight iterations and hence both methods were at least twice as fast as the Schur vector method;
- computing X_0 by Kleinman's stabilization procedure, Newton's method required more than 15 iterations only twice;
- computing X_0 by the procedure given in [1] Newton's method required more than 15 iterations once;
- the maximum number of iterations required using line searches is 15 for Example 8 with X_0 computed by the method from [1]. For all other examples, line search is faster than the Schur vector method. The maximum savings in iteration steps was obtained for Example 14 and X_0 computed by Kleinman's method [8]; here Newton's method requires 28 iterations while line search converges after only 9 iterations.

Note that for none of the benchmark examples, it was necessary to compute t_k by backtracking. For a complete account of the test results see [3].

6 Conclusions

The numerical experiments performed so far suggest that line search is an efficient tool to overcome some of the possible convergence problems of Newton's method for solving the DARE. When used for iterative refinement, line search will usually not improve Newton's method substantially unless the problem is very ill-conditioned. If used as an iterative method to solve the DARE, it is often much faster than Newton's method and the Schur vector method. Particularly for d-stable systems, starting the method from $X_0 = 0$, the numerical experiments suggest that Newton's method with line search is currently the fastest and most reliable method to solve DAREs.

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

References

[1] E.S. Armstrong and G.T. Rublein. A stabilization algorithm for linear discrete constant systems. *IEEE Trans. Automat. Control*, AC–21:629–631, 1976.

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

[3] P. Benner. *Contributions to the Numerical Solution of Algebraic Riccati Equations and Related Eigenvalue Problems.* Logos–Verlag, Berlin, Germany, 1997. *Also:* Dissertation, Fakultät für Mathematik, TU Chemnitz–Zwickau, 1997.

[4] P. Benner, A.J. Laub, and 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, 09107 Chemnitz, 1995.

[5] P. Benner and R. Byers. An exact line search method for solving generalized continuous-time algebraic Riccati equations. *IEEE Trans. Automat. Control*, 43:101–107, 1998.

[6] J. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall, Englewood Cliffs, NJ, 1983.

[7] 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.

[8] D.L. Kleinman. Stabilizing a discrete, constant, linear system with application to iterative methods for solving the Riccati equation. *IEEE Trans. Automat. Control*, AC-19:252–254, 1974.

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

[10] 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.

[11] 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.