Fast validated computation for solutions of discrete-time algebraic Riccati equations

Shinya Miyajima

Faculty of Science and Engineering, Iwate University

17th International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics

September 29, 2016, miyajima@iwate-u.ac.jp

Discrete-time algebraic Riccati equation

$$F(X) := A^H X A - X - A^H X B(R + B^H X B)^{-1} B^H X A + Q = 0,$$

$A, Q \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times m}$, $R \in \mathbb{C}^{m \times m}$: given, $Q^H = Q$, $R^H = R$,

$X \in \mathbb{C}^{n \times n}$: unknown, $m \leq n$

appears in: discrete-time LQ-optimal control problems, etc.

Stabilizing sol.: $F(X) = 0$, $X^H = X$ and $\rho(A_X) < 1$, where

$$A_X := A - B(R + B^H X B)^{-1} B^H X A$$

The stabilizing sol. is required in the practical applications.

Purpose

Numerically computing X^ε s.t. $|\tilde{X} - X^*| \le X^\varepsilon$, where $X^* :=$ sol., $\tilde{X} :=$ numerical sol., $|M| := (|M_{ij}|)$, $M \le N \Leftrightarrow M_{ij} \le M_{ij}, \forall i, j$.

$\Rightarrow$ If X^ε_{ij} is small, $\tilde{X}_{ij}$ is reliable. Moreover, $X^* \in \langle \tilde{X}, X^\varepsilon \rangle$.

Previous work (pioneering work)

Luther-Otten-Traczinski (1998) $\supseteq$ Luther-Otten (1999)

- The special structure of the DARE is skillfully exploited.
- $\mathcal{O}(n^6)$ ops. per iteration

Our contribution: 2 algorithms are proposed

- $\mathcal{O}(n^3)$ ops. per iteration
- The stabilizability and uniqueness can moreover be verified.

Alg.1: applicable when eigenvector matrix of $A_{\tilde{X}}$ is well-conditioned

Alg.2: less reliable, but not breaking down when ill-conditioned

Outline of Algorithm 1 (1/2)

1. Derivation of $G(Y)$ s.t. $F(X) = 0 \Leftrightarrow G(Y) = 0$
2. Enclosing Y^* s.t. $G(Y^*) = 0$

Outline of Algorithm 1 (2/2)

3. Computation of X^{ε} utilizing the enclosure
4. Checking eigenvalues of A_X, $\forall X \in \langle \tilde{X}, X^{\varepsilon} \rangle$

Derivation of $G(Y)$ (1/2)

Let $A_{\tilde{X}}^H V \approx V\Lambda$ (Λ: diagonal), V be nonsingular, and $W \approx V^{-1}$.

Define $E := \Lambda - WA_{\tilde{X}}^H V$, $Z := I_n - WV$, $Y := V^{-1}XV^{-H}$,

$\tilde{Y} := V^{-1}\tilde{X}V^{-H}$, $S_Y := R + (V^H B)^H Y V^H B$.

$F(X) = 0 \Leftrightarrow WF(X)W^H = 0$ and $WF(X)W^H = G(Y)$, where

Derivation of $G(Y)$ (2/2)

$$G(Y) := WA^H VYV^H AW^H - WVY(WV)^H + WQW^H \\ -(V^H AW^H)^H YV^H BS_Y^{-1}(V^H B)^H YV^H AW^H.$$

We can find its advantage when we consider

$$G'_{\tilde{Y}}(H) = WA_{\tilde{X}}^H VH(WA_{\tilde{X}}^H V)^H - WVH(WV)^H \\ = (\Lambda - E)H(\Lambda - E)^H - (I_n - Z)H(I_n - Z)^H,$$

whereas $F'_{\tilde{X}}(H) = A_{\tilde{X}}^H HA_{\tilde{X}} - H$.

Enclosing $\boldsymbol{Y}^*$ where $\boldsymbol{G(Y^*) = 0}$

We compute $\boldsymbol{Y}$ s.t. $\boldsymbol{Y} \ni Y^*$. If $G'_{\tilde{Y}}(H)$ is invertible, $Y^* = N(Y^*)$, where $N(Y) := Y - (G'_{\tilde{Y}})^{-1}(G(Y))$.

$\Rightarrow$ We verify the invertibility and

$\{N(Y) : Y \in \langle \tilde{Y}, Y^r \rangle\} \subseteq \mathrm{int}(\langle \tilde{Y}, Y^r \rangle)$ for given $Y^r > 0$.

If these are true, $Y^* \in \mathrm{int}(\langle \tilde{Y}, Y^r \rangle)$.

$\Rightarrow Y^* = N(Y^*) \in \{N(Y) : Y \in \langle \tilde{Y}, Y^r \rangle\}$

$\Rightarrow \boldsymbol{Y}$ can be computed s.t. $\{N(Y) : Y \in \langle \tilde{Y}, Y^r \rangle\} \subseteq \boldsymbol{Y}$.

How to verify the invertibility?

$\mathrm{vec}(G'_{\tilde{Y}}(H)) = \mathsf{P}\mathrm{vec}(H)$, where

$\mathsf{P} := \overline{(\Lambda - E)} \otimes (\Lambda - E) - \overline{(I_n - Z)} \otimes (I_n - Z)$.

$\Rightarrow$ If P is nonsingular, $G'_{\tilde{Y}}(H)$ is invertible.

$\Rightarrow$ We verify the nonsingularity of $\mathsf{P} \in \mathbb{C}^{n^2 \times n^2}$.

We can exploit the special structures of $\Lambda - E$ and $I_n - Z$.

$\Rightarrow$ The verification is possible with $\mathcal{O}(n^3)$ ops..

Verifying $\{N(Y) : Y \in \langle \tilde{Y}, Y^r \rangle\} \subseteq \mathrm{int}(\langle \tilde{Y}, Y^r \rangle)$

We compute Y^ε s.t. $\{N(Y) : Y \in \langle \tilde{Y}, Y^r \rangle\} \subseteq \langle \tilde{Y}, Y^\varepsilon \rangle$

and verify $Y^\varepsilon < Y^r$. Y^ε can be obtained by the following idea:

$N(Y) = Y - (G'_{\tilde{Y}})^{-1}(G(Y)) \Leftrightarrow G'_{\tilde{Y}}(N(Y)) = G'_{\tilde{Y}}(Y) - G(Y)$

$\Rightarrow \{N(Y) : Y \in \langle \tilde{Y}, Y^r \rangle\}$ is the set of all sol. of

$(\Lambda - E)N_Y(\Lambda - E)^H - (I_n - Z)N_Y(I_n - Z)^H = G'_{\tilde{Y}}(Y) - G(Y)$,

$N_Y \in \mathbb{C}^{n\times n}$: unknown, $Y \in \langle \tilde{Y}, Y^r \rangle$: parameter

$\Rightarrow$ We enclose the sol. set (possible with $\mathcal{O}(n^3)$ ops.).

Computation of $\boldsymbol{X}^{\varepsilon}$ utilizing $\boldsymbol{Y}$

We compute $\boldsymbol{X}$ s.t. $\{VYV^H : Y \in \boldsymbol{Y}\} \subseteq \boldsymbol{X}$.

Since $X^* = VY^*V^H$, $X^* \in \boldsymbol{X}$. Therefore, $X^{\varepsilon} = \mathrm{rad}(\boldsymbol{X})$.

Verification of the stabilizability

From [Corollary 2, Ionescu-Weiss (1992)], we can prove

Corollary 1 If $F(X^*) = 0$ and $\rho(A_{X^*}) < 1$, then ${X^*}^H = X^*$.

$\Rightarrow$ If $X^* \in \langle \tilde{X}, X^{\varepsilon} \rangle$, showing $\rho(A_X) < 1, \forall X \in \langle \tilde{X}, X^{\varepsilon} \rangle$ is sufficient

$\Rightarrow$ We enclose all the eigenvalues of $V^{-1}A_X^H V$, $\forall X \in \langle \tilde{X}, X^{\varepsilon} \rangle$.

Verification of the (local) uniqueness

Theorem 2 (Ionescu-Weiss (1992)) The stabilizing sol. is (globally) unique if it exists.

Assume $X^* \in \langle \tilde{X}, X^\varepsilon \rangle$ and $\rho(A_X) < 1$, $\forall X \in \langle \tilde{X}, X^\varepsilon \rangle$.

If $\langle \tilde{X}, X^\varepsilon \rangle$ contains 2 sol.s, both of them must be stabilizing, which contradicts Theorem 2.

$\Rightarrow$ Showing $\rho(A_X) < 1$, $\forall X$ is sufficient. $\Rightarrow$ Nothing to do

Algorithm 2

We can write $X = \tilde{X} + U$. Then, $F(X) = 0 \Leftrightarrow U = P(U)$, where

$$P(U) := A_{\tilde{X}}^H U(I_n - B(R + B^H(\tilde{X} + U)B)^{-1}B^H U)A_{\tilde{X}} + F(\tilde{X}).$$

$\Rightarrow$ We compute $\boldsymbol{P}$ s.t. $\boldsymbol{P} \supseteq \{P(U) : U \in \boldsymbol{U}\}$ for given $\boldsymbol{U}$.

If $\boldsymbol{P} \subseteq \mathrm{int}(\boldsymbol{U})$, $\mathrm{int}(\boldsymbol{U}) \ni U^*$ s.t. $U^* = P(U^*)$.

Then, $U^* = P(U^*) \in \{P(U) : U \in \boldsymbol{U}\} \subseteq \boldsymbol{P} \Rightarrow X^* \in \tilde{X} + \boldsymbol{P}$.

Numerical results

Intel Core 2.60GHz CPU, 8.00GB RAM, MATLAB R2012a with Intel MKL and IEEE 754 double precision

`LOT1`: 1st algorithm by Luther-Otten-Traczinski (1998) [existence]
`LOT2`: 2nd algorithm by Luther-Otten-Traczinski (1998) [existence]
`M1`: Alg.1 [existence, stabilizability and uniqueness]
`M2`: `M1` with accurate evaluation of $F(\tilde{X})$ [similar to `M1`]
`M3`: Alg.2 [similar to `M1`]
`M4`: `M3` with accurate evaluation of $F(\tilde{X})$ [similar to `M1`]

maximum radius $:= \max_{i,j} X^{\varepsilon}_{ij}$, where $\langle \tilde{X}, X^{\varepsilon} \rangle \ni X^*$

Example 1

$$A = (M - \delta_t K)^{-1} M,\ B = \delta_t (M - \delta_t K)^{-1} F,\ Q = C^T C,\ R = I_m,$$

$$M = \frac{1}{6n} \begin{bmatrix} 4 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 4 \end{bmatrix}, \quad K = -\alpha n \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix},$$

$\delta_t = 0.01$, $F = \texttt{rand}(n,m)$, $C = \texttt{rand}(m,n)$, $\alpha = 0.002$, $m = 10$

We generated 100 problems and took the median of the maximum radii and computing times.

Maximum radii

n	LOT1	LOT2	M1	M2	M3	M4
10	2.9e–10	2.9e–10	1.2e–8	1.7e–11	failed	failed
20	1.2e–7	1.2e–7	2.7e–5	2.3e–11	failed	failed
30	1.2e–6	1.4e–6	5.5e–4	2.4e–11	failed	failed
40	failed	failed	6.5e–3	2.5e–11	failed	failed
50	failed	failed	4.7e–2	3.6e–11	failed	failed
60	failed	failed	1.5e–1	5.3e–11	failed	failed
100	failed	failed	failed	5.9e–11	failed	failed
200	failed	failed	failed	2.3e–10	failed	failed
300	failed	failed	failed	4.2e–9	failed	failed

Computing times (sec)

n	LOT1	LOT2	M1	M2	M3	M4
10	4.2e–1	4.4e–1	2.2e–2	2.6e–1	failed	failed
20	1.5e+0	1.5e+0	3.1e–2	7.4e–1	failed	failed
30	6.3e+0	6.3e+0	4.9e–2	1.6e+0	failed	failed
40	failed	failed	6.9e–2	2.8e+0	failed	failed
50	failed	failed	1.1e–1	4.9e+0	failed	failed
60	failed	failed	1.9e–1	7.9e+0	failed	failed
100	failed	failed	failed	4.2e+1	failed	failed
200	failed	failed	failed	1.8e+2	failed	failed
300	failed	failed	failed	5.1e+2	failed	failed

Number of failed problems (within 100)

n	existence						stabilizability and uniqueness			
	LOT1	LOT2	M1	M2	M3	M4	M1	M2	M3	M4
10	0	0	0	0	100	100	0	0	100	100
20	0	0	0	0	100	100	100	0	100	100
30	96	93	0	0	100	100	100	0	100	100
40	100	100	0	0	100	100	100	1	100	100
50	100	100	3	0	100	100	100	27	100	100
60	100	100	46	0	100	100	100	92	100	100
100	100	100	100	0	100	100	100	100	100	100
200	100	100	100	0	100	100	100	100	100	100
300	100	100	100	0	100	100	100	100	100	100

Example 2

$m = 4$, $A = A_0/\|A_0\|_1$ and A_0 is obtained from the centered finite difference discretization of

$$L(u) = \Delta u - 10y\frac{\partial u}{\partial x} - 10\frac{\partial u}{\partial y} - 100u,$$

on $[0,1] \times [0,1]$ with homogeneous Dirichlet boundary conditions.

Then $n = n_0^2$ (n_0: number of inner grid points in each direction)

$R = I_m$, `B = rand(n,m); C = rand(m,n); Q = C'*C;`

We generated 100 problems and took the median of the maximum radii and computing times.

Maximum radii

n_0	LOT1	LOT2	M1	M2	M3	M4
3	2.7e–11	2.7e–11	6.1e–9	1.4e–13	failed	failed
4	6.4e–10	6.4e–10	5.1e–7	6.0e–13	failed	failed
5	7.8e–9	7.8e–9	2.9e–4	2.5e–11	failed	failed
6	7.4e–8	7.4e–8	5.7e–4	2.2e–11	failed	failed
7	4.9e–7	4.7e–7	8.8e–3	7.4e–11	failed	failed
8	failed	failed	8.8e–2	2.1e–10	failed	failed
11	failed	failed	2.0e+0	2.3e–9	failed	failed
12	failed	failed	failed	8.4e–9	failed	failed
20	failed	failed	failed	2.1e–7	failed	failed

Computing times (sec)

n_0	LOT1	LOT2	M1	M2	M3	M4
3	4.1e–1	4.2e–1	2.6e–2	1.6e–1	failed	failed
4	9.2e–1	9.4e–1	2.7e–2	4.0e–1	failed	failed
5	4.0e+0	4.1e+0	3.4e–2	9.1e–1	failed	failed
6	1.7e+1	1.7e+1	4.8e–2	1.8e+0	failed	failed
7	6.2e+1	6.1e+1	6.4e–2	3.4e+0	failed	failed
8	failed	failed	7.2e–2	6.0e+0	failed	failed
11	failed	failed	2.3e–1	2.3e+1	failed	failed
12	failed	failed	failed	3.5e+1	failed	failed
20	failed	failed	failed	4.2e+2	failed	failed

Number of failed problems (within 100)

n_0	existence						stabilizability and uniqueness			
	LOT1	LOT2	M1	M2	M3	M4	M1	M2	M3	M4
3	0	0	0	0	100	100	0	0	100	100
4	0	0	0	0	100	100	22	0	100	100
5	0	0	0	0	100	100	100	0	100	100
6	0	0	0	0	100	100	100	0	100	100
7	7	2	0	0	100	100	100	1	100	100
8	100	100	0	0	100	100	100	6	100	100
11	100	100	99	0	100	100	100	100	100	100
12	100	100	100	0	100	100	100	100	100	100
20	100	100	100	0	100	100	100	100	100	100

Example 3

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & & & 0 \\ 0 & & & 0 & 1 \\ 0 & \cdots & & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad Q = I_n, \quad R = 1,$$

$m = 1$

Stabilizing sol. $X^* = \mathrm{diag}(1, \ldots, n)$

Then, $A_{X^*} = A \Rightarrow V$ becomes singular or ill-conditioned.

Maximum radii

n	LOT1	LOT2	M1	M2	M3	M4
10	2.7e–15	1.4e–14	failed	failed	1.6e–14	2.7e–15
20	5.3e–15	6.0e–14	failed	failed	6.8e–14	5.3e–15
30	5.3e–15	1.3e–13	failed	failed	1.3e–13	5.3e–15
40	1.1e–14	2.6e–13	failed	failed	2.7e–13	1.1e–14
50	1.1e–14	4.0e–13	failed	failed	4.1e–13	1.1e–14
60	failed	failed	failed	failed	5.5e–13	1.1e–14
100	failed	failed	failed	failed	1.6e–12	2.1e–14
200	failed	failed	failed	failed	6.5e–12	4.3e–14
300	failed	failed	failed	failed	1.5e–11	8.5e–14

Computing times (sec)

n	LOT1	LOT2	M1	M2	M3	M4
10	6.1e–1	6.4e–1	failed	failed	4.6e–2	1.2e–1
20	3.3e+0	3.3e+0	failed	failed	8.3e–2	3.4e–1
30	2.0e+1	1.9e+1	failed	failed	1.2e–1	6.9e–1
40	7.3e+1	7.4e+1	failed	failed	1.7e–1	1.2e+0
50	2.3e+2	2.3e+2	failed	failed	2.3e–1	1.8e+0
60	failed	failed	failed	failed	3.4e–1	2.6e+0
100	failed	failed	failed	failed	8.3e–1	7.2e+0
200	failed	failed	failed	failed	5.0e+0	3.4e+1
300	failed	failed	failed	failed	2.2e+1	1.1e+2