Approximate solutions of Lyapunov and Riccati equations

Mark Opmeer

UNIVERSITY OF BATH

The approximation problem

Infinite-dimensional optimal control problem

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad y(t) = Cx(t).$$

$$\langle Qx_0, x_0 \rangle = \inf_u \int_0^\infty \|u(t)\|^2 + \|y(t)\|^2 \, dt.$$

$$A^*Q + QA + C^*C - QBB^*Q = 0.$$

Finite-dimensional approximation on n-dimensional space

$$\dot{x}_n(t) = A_n x_n(t) + B_n u_n(t), \quad x_n(0) = x_{n,0}, \quad y_n(t) = C_n x_n(t).$$

$$A_n^* Q_n + Q_n A + C_n^* C_n - Q_n B_n B_n^* Q_n = 0.$$

- Note that $\operatorname{rank}(Q_n) \leq n$.
- So fundamental question: how well can Q be approximated by operators of rank $\leq n$?
- i.e., how does $\sigma_n(Q)$ behave?

Special case $B = 0$: the Lyapunov equation

Case $C = I$ and $A = A^* \ll 0$

$$A^*Q + QA + C^*C = 0 \iff Q = \frac{-1}{2}A^{-1}.$$

- $\sigma_n(Q) = \frac{-1}{2\sigma_n(A)}$.
- So for the canonical example of the Dirichlet Laplacian in d spatial dimensions: $\sigma_n(Q) \sim \frac{1}{n^{2/d}}$.

Ober (1987)

On state space $\ell^2(\mathbb{N})$

$$C_k = \frac{\sigma_k}{k}, \qquad A_{ij} = \frac{-C_i C_j}{\sigma_i + \sigma_j},$$

Then $Q = \mathrm{diag}(\sigma_k)$.

- So anything can happen to $\sigma_n(Q)$ even when
  - $A = A^* < 0$ bounded,
  - C bounded,
  - The output space one-dimensional.

- Penzl (2000), *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*
- Antoulas, Sorensen, Zhou (2002) *On the decay rate of the Hankel singular values and related issues*
- Grasedyck (2004), *Existence of a low rank or H-matrix approximant to the solution of a Sylvester equation*
- Baker, Embree, Sabino (2015), *Fast singular value decay for Lyapunov solutions with nonnormal coefficients*

- Curtain, Sasane (2001), *Compactness and nuclearity of the Hankel operator and internal stability of infinite-dimensional state linear systems*
- Opmeer (2010), *Decay of Hankel singular values of analytic control systems*
- Grubisic, Kressner (2014), *On the eigenvalue decay of solutions to operator Lyapunov equations*
- Opmeer (2015), *Decay of singular values of the Gramians of infinite-dimensional systems*

Upper bound for singular values

- A generates an exponentially stable analytic strongly continuous semigroup.
- C is strictly less than half as unbounded as A (in an interpolation space sense) and the output space is finite-dimensional.

Then there exist $M, \vartheta > 0$ such that $\sigma_n(Q) \leq M\mathrm{e}^{-\vartheta\sqrt{n}}$.

- B is strictly less than half as unbounded as A (in an interpolation space sense).

Then the same estimate holds true for the solution of the algebraic Riccati equation.

$$Q = \int_0^\infty \mathrm{e}^{A^*t} C^* C \mathrm{e}^{At}\, dt.$$

Proof uses ideas from the (Banach-space valued generalization of) **Weideman, Trefethen; *The exponentially convergent trapezoidal rule*, SIAM Review, 2014**

Computational method: ADI

ADI: $S_n^* S_n \approx Q$; shift parameters α_i with $\mathrm{Re}(\alpha_i) > 0$

1: $V_1 = (\alpha_1 I - A^*)^{-1} C^*$
2: $S_1 = \sqrt{2\,\mathrm{Re}(\alpha_1)} \cdot V_1^*$
3: **for** $i = 2, 3, \ldots, n$ **do**
4: $\quad V_i = V_{i-1} - (\alpha_i + \overline{\alpha_{i-1}}) \cdot (\alpha_i I - A^*)^{-1} V_{i-1}$
5: $\quad S_i = [\, S_{i-1}^*\,,\ \sqrt{2\,\mathrm{Re}(\alpha_i)} \cdot V_i\,]^*$
6: **end for**

- The application of $(\alpha_i I - A^*)^{-1}$ is really solving an (shifted) elliptic boundary value problem.

- Wachspress (1988)
- Li and White (2002)
- Opmeer, Reis and Wollner (2013)

ADI does not always converge (to the correct limit)

$A = -1, C = \sqrt{2} \implies Q = 1.$ $\alpha_k = \frac{1}{8k^2-1} \implies Q_n \to 1 - \frac{4}{\pi^2}.$

- ADI converges provided that $\sum_{j=1}^{\infty} \frac{\mathrm{Re}(\alpha_j)}{1+|\alpha_j|^2} = \infty.$
- Convergence happens in the Schatten class S_p (with $1 \leq p \leq \infty$) provided that $Q \in S_p$.

What ADI really is

S_k is a matrix representation of $P_k \Psi$ where

- $\Psi : \mathscr{X} \to L^2(0, \infty; \mathscr{Y})$ is the initial condition to output map,
- P_k is the orthogonal projection onto

  $$\mathscr{K}_k(\alpha) := \mathrm{span}\{t \mapsto \mathrm{e}^{-\alpha_1 t}, \ldots, t \mapsto \mathrm{e}^{-\alpha_k t}\}$$

  (in case $\alpha_i \neq \alpha_j$ for $i \neq j$).
- the basis is $\mathscr{K}_k$ is the (orthonormal) Takenaka–Malmquist basis.

Therefore $Q_n = \Psi^* P_n \Psi$.

Solution of the Riccati equation

For stable systems: $Q = \Psi^*(I + \mathbb{F}^*\mathbb{F})^{-1}\Psi$, where

- $\Psi : \mathscr{X} \to L^2(0, \infty; \mathscr{Y})$ is the initial condition to output map,
- $\mathbb{F} : L^2(0, \infty; \mathscr{Y}) \to L^2(0, \infty; \mathscr{Y})$ is the input-to-output map.

ADI approximation for Riccati equations (Massoudi, Opmeer and Reis SIMAX 2016)

Approximate Ψ by $P_k\Psi$ as before and $\mathbb{F}$ by $P_k\mathbb{F}|_{\mathscr{K}_k}$.

Connection to projected optimal control problem

$$Q_n \leftrightarrow \inf_{u \in \mathscr{K}_n(\alpha)} \|P_n y\|_{L^2}^2 + \|u\|_{L^2}^2.$$

- Same algorithm as: Lin and Simoncini *A new subspace iteration method for the algebraic Riccati equation*, 2014.
- See also Peter Benner et al. *The RADI algorithm for solving large-scale algebraic Riccati equations*, 2015 preprint.

ADI also generalizes to Lur'e equations (Massoudi, Opmeer and Reis 2014 preprint).

Example

$\Omega := [0,1] \times [0,1]$ be the unit square with boundary $\partial\Omega := \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$, where $\Gamma_1 := \{0\} \times [0,1]$, $\Gamma_2 := [0,1] \times \{0\}$, $\Gamma_3 := [0,1] \times \{1\}$, and $\Gamma_4 := \{1\} \times [0,1]$.
We consider the two-dimensional convection-diffusion equation

$$\frac{\partial x}{\partial t}(\xi,t) = \Delta x(\xi,t) + b^\top \nabla x(\xi,t), \quad (\xi,t) \in \Omega \times \mathbb{R}_{\geq 0},$$

with Robin boundary conditions

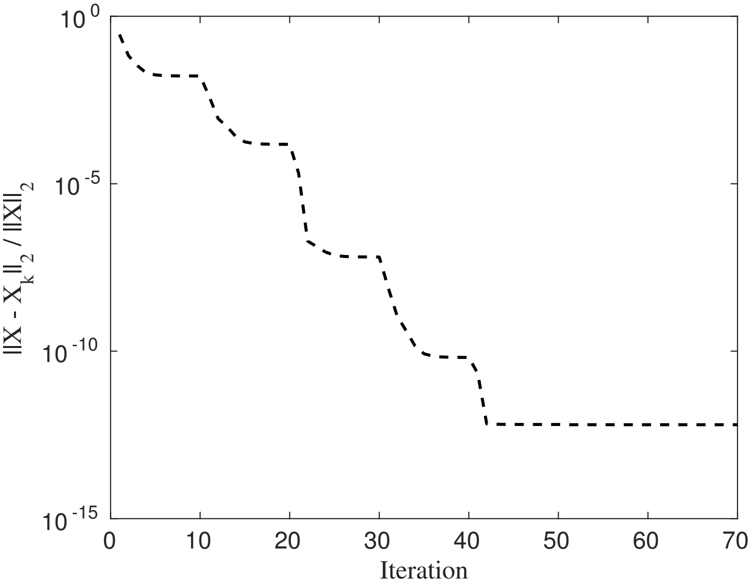
$$\begin{aligned} u(t) &= \nu(\xi)^\top \nabla x(\xi,t) + a x(\xi,t), \qquad (\xi,t) \in (\Gamma_1 \cup \Gamma_2) \times \mathbb{R}_{\geq 0}, \\ 0 &= \nu(\xi)^\top \nabla x(\xi,t) + a x(\xi,t), \qquad (\xi,t) \in (\Gamma_3 \cup \Gamma_4) \times \mathbb{R}_{\geq 0}. \end{aligned}$$

and two-dimensional output

$$y(t) = \begin{bmatrix} \int_{\Gamma_1} x(\xi,t) d\sigma_\xi \\ \int_{\Gamma_3} x(\xi,t) d\sigma_\xi \end{bmatrix},$$

where σ_ξ denotes the surface measure and $\nu(\xi)$ denotes the outward normal.

Shifted elliptic PDEs in ADI solved using piecewise linear finite elements with $N = 3600$



10^{0}
10^{-5}
10^{-10}
10^{-15}
0
10
20
30
40
50
60
70
Iteration
$\|X - X_k\|_2 / \|X\|_2$

Euler–Bernoulli beam with Kelvin-Voigt and viscous damping

$$\rho w_{tt}(r,t) + C_v w_t(r,t) + \left[C_d I_b w_{rrt}(r,t) + EI_b w_{rr}(r,t)\right]_{rr} = 0,$$

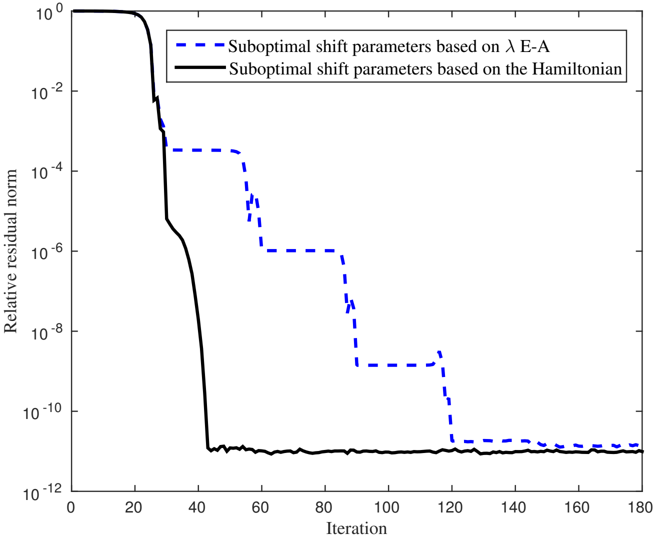
with boundary conditions and controls

$$\begin{aligned} w(0,t) &= 0, \\ w_r(0,t) &= 0, \\ \left[C_d I_b w_{rrt}(r,t) + EI_b w_{rr}(r,t)\right]_{r=R} &= u_1(t), \\ \left[C_d I_b w_{rrrt}(r,t) + EI_b w_{rrr}(r,t)\right]_{r=R} &= u_2(t). \end{aligned}$$

Two dimensional boundary observation at the free end $(r = R)$

$$y(t) = \begin{bmatrix} w_t(R,t) \\ w_{rt}(R,t) \end{bmatrix}.$$

Shifted elliptic PDEs in ADI solved using cubic B-splines with $N = 96$



The matlab “care” function gives a warning and is less accurate than ADI.

Positive Real example

We consider a convection-diffusion equation on the unit square $\Omega := [0, 1] \times [0, 1]$, namely

$$\frac{\partial x}{\partial t}(\xi, t) = k\Delta x(\xi, t) + b^\top \nabla x(\xi, t), \quad (\xi, t) \in \Omega \times \mathbb{R}_{\geq 0}.$$

The input is a scalar function formed by the Robin boundary condition

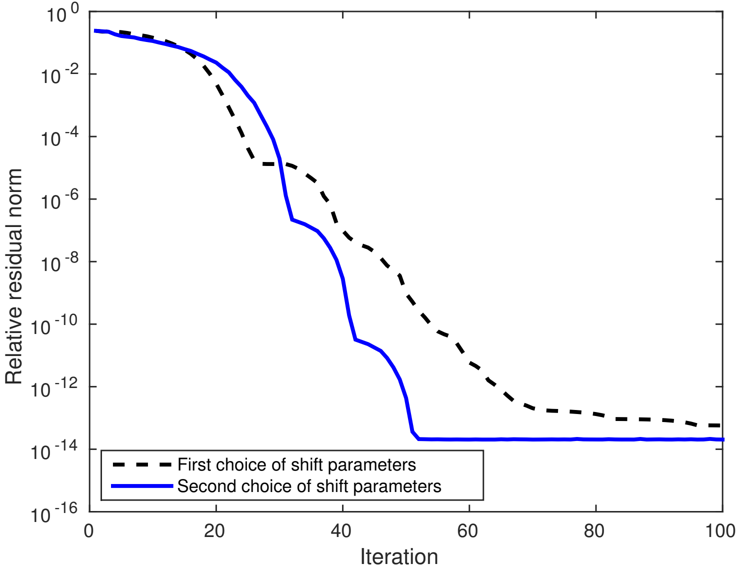
$$u(t) = \nu(\xi)^\top \nabla x(\xi, t) + \alpha x(\xi, t), \quad (\xi, t) \in \partial\Omega \times \mathbb{R}_{\geq 0},$$

and the output consists of the integral of Dirichlet boundary values, i.e.

$$y(t) = \int_{\partial\Omega} x(\xi, t) d\sigma_\xi,$$

where $\partial\Omega$ denotes the boundary of Ω, σ_ξ denotes the surface measure, and $\nu(\xi)$ denotes the outward unit normal.

Shifted elliptic PDEs in ADI solved using piecewise linear finite elements with $N = 4900$



There is no built-in-matlab function to solve this Lur'e equation.