Lecture notes

Federico Poloni

1 Introduction

Riccati-type matrix equations are a family of matrix equations that appears very frequently in literature and applications, especially in systems theory. One of the reasons why they are so ubiquitous is that they are equivalent to certain invariant subspace problems; this equivalence connects them to a larger part of numerical linear algebra, and opens up avenues for many solution algorithms.

Many books (and even more articles) have been written on these equations; among them, we recall the classical monography by Lancaster and Rodman [56], a review book edited by Bittanti, Laub and Willems [22], various treatises which consider them from different points of view such as [1, 4, 5, 19, 24, 34, 50, 64], and recently also a book devoted specifically to doubling [49].

This vast theory can be presented from different angles; in this exposition, we aim to present a selection of topics which differs from that of the other books and treatises, and focuses on computational linear algebra. In addition to other more common algorithm, we try to introduce doubling algorithms with a direct approach, explaining in particular that they arise as 'doubling variants' of other more basic iterations, and detailing how they are related to the subspace iteration, to ADI, to cyclic reduction and to Schur complements. We do not treat algorithms and equations with the greatest generality possible, to reduce technicalities.

1.1 Example: heating a long corridor

The main application behind these equations is so-called *control theory*, a popular field in engineering. The idea behind this is altering the behavior of certain linear dynamical systems using an additional input.

Consider a very long and thin corridor with a radiator at one end and an open window at the other. We can model it as the segment $[0, 1]$; the endpoint 1 (window) is at constant temperature $0°\mathsf{C}$, and the endpoint 0 (radiator) is at a temperature $u(t)$ that we can choose by turning a valve.

We discretize the space: $x(t)$ is the vector of temperatures at equi-spaced points $h, 2h, \dots, (n-1)h$ (the temperatures at 0 and $(n+1)h = 1$ are prescribed). We get

$$\frac{d}{dt}x(t) = Ax(t) + bu(t),$$

with $A = \alpha h^2 \operatorname{tridiag}(1, -2, 1)$ and $b = \alpha h^2 e_1$.

This is a first example of a linear control system: we are given a target temperature x_F at each point of the room, and we want to choose $u(t)$ so that we reach it.

If left to itself at temperature $u(t) = 0$, this system is stable: all the eigenvalues of A are in the left half-plane, so the temperature will (slowly) converge to zero.

This dynamical system is in continuous time, but we can also formulate a variant in discrete time; it takes the form

$$x_{k+1} = Ax_k + bu_k.$$

Some formulas are simpler to state and prove in discrete time, but modelling often naturally produces continuous-time linear systems; so we see a litle bit of both here.

In the following, we use the notation $A \succ B$ (resp. $A \succeq B$) to mean that $A - B$ is positive definite (resp. semidefinite) (Loewner order). We use $\rho(M)$ to denote the spectral radius of M, the symbol LHP $= \{z \in \mathbb{C} : \mathrm{Re}(z) < 0\}$ to denote the (open) left half-plane, and RHP for the (open) right half-plane. We use the notation $\Lambda(M)$ to denote the spectrum of M, i.e., the set of its eigenvalues. We use M^* to denote the conjugate transpose, and $M^\top$ to denote the transpose without conjugation, which appears when combining vectorizations and Kronecker products with the identity $\mathrm{vec}(MXN) = (N^\top \otimes M)\,\mathrm{vec}(X)$ [41, Sections 1.3.6–1.3.7].

2 Stein equations

The simplest matrix equation that we consider is the *Stein equation* (or *discrete-time Lyapunov equation*).

$$X - A^*XA = Q, \quad Q = Q^* \succeq 0, \tag{1}$$

for $A, X, Q \in \mathbb{C}^{n\times n}$. Here X is the unknown, while A and Q are given.

2.1 Relation to dynamical systems

Consider the discrete-time constant-coefficient linear dynamical system

$$x_{k+1} = Ax_k, \quad x_0 \in \mathbb{C}^n. \tag{2}$$

We know that $x_k = A^k x_0$, and that $\lim_{k\to\infty} x_k = 0$ if and only if $\rho(A) < 1$. One of the classical uses for the Stein equation is proving that the dynamical system is stable without having to compute an eigendecomposition (at the time of Lyapunov, no computers existed!). Indeed, the following result holds.

Lemma 1. *Let $Q \succ 0$, and suppose that the Stein equation (1) has solution $X \succ 0$. Then, $\rho(A) < 1$.*

Proof. Let $Av = v\lambda$ be an eigenpair of A. Then, we have

$$v^*Qv = v^*(X - A^*XA)v = (1 - \overline{\lambda}\lambda)v^*Xv.$$

Hence

$$1 - \overline{\lambda}\lambda = \frac{v^*Qv}{v^*Xv} < 0,$$

i.e., $|\lambda|^2 < 1$. □

Hence, if we wish to check that A has eigenvalues in the unit circle, we choose any $Q \succ 0$, compute X by solving the vectorized linear system (4), and check that $X \succ 0$. All these operations can be done by hand, without a computer. If A has rational entries, we never have to deal with irrationals.

Note that this method is guaranteed to succeed: whenever $Q \succ 0$ and $\rho(A) < 1$, we get $X \succ 0$, thanks to the expression (6), which is a sum of positive definite matrices.

Lyapunov's argument actually was slightly different, without involving eigenvalues at all. Suppose we have found a Stein equation with $X \succ 0$, $Q \succ 0$. Let us consider the energy function $V(x) := x^*Xx$. Then,

$$V(x_k) - V(x_{k+1}) = x_k^*(X - A^*XA)x_k = x_k^*Qx_k > 0,$$

i.e., Q is decreasing over the trajectories of (2). So $V(x_k)$ is bounded, and so is the sequence of vectors $\{x_k\}$, since $X \succ 0$. By strengthening this argument slightly, we can prove that x_k actually converges to 0: suppose by contradiction that $\|x_k\| > \varepsilon$ for all k; then $V(x_k) - V(x_{k+1}) > \lambda_{\min}(Q)\varepsilon^2$, which is impossible since $V(x) \geq 0$.

2.2 Vectorization

This is an 'easy' equation, from the linear algebra point of view, because it is *linear*: the $n \times n$ matrices form a vector space, and the left-hand side is a linear operator. We can write out this equation explicitly as a linear system using *vectorization* and *Kronecker products*.

We define the map $\mathrm{vec}\colon \mathbb{C}^{n\times n} \to \mathbb{C}^{nn}$ as

$$X = \left[\begin{array}{c|c|c|c} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{array}\right] \in \mathbb{C}^{n\times n}, \quad \mathrm{vec}\, X := \left[\begin{array}{c} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \\ \hline x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \\ \hline \vdots \\ \hline \\ x_{1n} \\ x_{2n} \\ \vdots \\ x_{nn} \end{array}\right].$$

In plain words, $\mathrm{vec}\, X$ is the vector obtained by stacking the columns of the matrix X one on top of the other.

With vectorization, the elements of X are listed in *column-major* order: the leftmost index is the one that 'changes more often'. This choice matches the memory layout in which matrices are stored in memory in most languages, like Matlab, Fortran, Python/Numpy (C/C++ prefer row-major instead), and it is often the most convenient one to work with.

[Matlab example]

There is an identity that gives explicitly the matrix associated to the linear operator $X \mapsto BXA$:

$$\mathrm{vec}(BXA) = (A^\top \otimes B)\,\mathrm{vec}(X), \tag{3}$$

where $A^\top \otimes B$ is the $n^2 \times n^2$ matrix

$$\begin{bmatrix} a_{11}B & a_{21}B & \dots & a_{n1}B \\ a_{12}B & a_{22}B & \dots & a_{n2}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n}B & a_{2n}B & \dots & a_{nn}B \end{bmatrix}$$

In our case, we set $B = A^*$, so the Stein equation in vectorized form becomes

$$(I_{n^2} - \underbrace{A^\top \otimes A^*}_{:=M})\,\mathrm{vec}(X) = \mathrm{vec}(Q). \tag{4}$$

Note that one of the two symbols is a transpose-conjugate $*$ and one is a plain conjugate $\top$. This is not a mistake: the identity (3) is one of the few places in linear algebra when one encounters a transpose that does not become a conjugate transpose when dealing with complex matrices.

Using more properties of the Kronecker product, we can tell explicitly what the eigenvalues of M (and hence of the system matrix $I-M$) are. Let $A = UTU^*$ be a Schur factorization of A, with

$$\operatorname{diag}(T) = (\lambda_1, \dots, \lambda_n) = \Lambda(A).$$

Then, we can factor M as

$$M = A^\top \otimes A^* = (\bar{U} \otimes U)(T^\top \otimes T^*)(U^\top \otimes U^*). \tag{5}$$

This is (up to transposition) a Schur factorization of M, since $T^\top \otimes T^*$ is lower triangular and $(\bar{U} \otimes U)$ is orthogonal with inverse $(U^\top \otimes U^*)$; in particular, it reveals that the n^2 eigenvalues of M are the diagonal entries

$$\lambda_1\overline{\lambda_1}, \lambda_1\overline{\lambda_2}, \dots, \lambda_1\overline{\lambda_n}, \lambda_2\overline{\lambda_1}, \lambda_2\overline{\lambda_2}, \dots, \lambda_n\overline{\lambda_n}.$$

In particular, the Stein equation is solvable whenever A does not have a pair of eigenvalues λ_i, λ_j such that $\lambda_i\overline{\lambda_j} = 1$. An important case when this happens is when $\rho(A) < 1$.

Exercise 2. *Show that the Kronecker product satisfies* $(B \otimes A)(D \otimes C) = BD \otimes AC$*, when the dimensions are compatible, and that* $(B \otimes A)^\top = B^\top \otimes A^\top$*. Use these properties to verify the steps that lead to* (5)*.*

We conclude this section with another useful formula: when $\rho(A) < 1$, then also $\rho(M) < 1$, and we can use the Neumann series $(I-M)^{-1} = I + M + M^2 + M^3 + \dots$ to solve the system:

$$\operatorname{vec}(X) = \operatorname{vec}(Q) + (A^\top \otimes A^*)\operatorname{vec}(Q) + (A^\top \otimes A^*)^2 \operatorname{vec}(Q) + \dots,$$

or, in non-vectorized form,

$$X = Q + A^*QA + (A^*)^2QA^2 + \cdots = \sum_{k=0}^{\infty} (A^*)^k Q A^k. \tag{6}$$

This infinite sum converges if $\rho(A) < 1$. This form also makes it apparent that the solution X must be symmetric; this follows also from transposing (1) (even without assuming $\rho(A) < 1$).

2.3 Controllability

There is another important application in linear control theory that results in Lyapunov equations: that of *controllability*. Suppose that, at each step k, we are allowed to modify the behavior of our dynamical system by adding a certain multiple of a vector $b \in \mathbb{C}^n$. In engineering, this is usually a very practical matter: the dynamical system describes the evolution of a certain quantity (e.g., the temperature in a room, the position of a drone), and this multiple of b is a *control*: e.g., a radiator that changes the temperature, or an engine

attached to a propeller that makes the drone go up or down. Sometimes, there is more than one control, resulting in a matrix $B \in \mathbb{C}^{n\times m}$ rather than a vector, but the differences are minor, so we focus on the case of one vector only. Our goal is reaching a certain state x_F, starting from x_0.

Hence the modified system is

$$x_{k+1} = Ax_k + bu_k \quad k = 0, 1, 2, \dots, \tag{7}$$

where $u_1, u_2, \dots \in \mathbb{C}$ are the scalar multiples we choose, and and at time k we have

$$x_k = A^k x_0 + bu_{k-1} + Abu_{k-2} + A^2 bu_{k-3} + \dots + A^{k-1} bu_0.$$

Hence, at time k we can reach all states x_k such that

$$x_k - A^k x_0 \in \underbrace{\operatorname{span}(b, Ab, A^2 b, \dots, A^{k-1} b)}_{K_k(A,b)}.$$

The linear subspace in the right-hand side is a very famous one in numerical linear algebra; it is the so-called *Krylov space*. A useful alternative characterization is

$$K_k(A, b) = \{p(A)b \colon p \text{ is a polynomial of degree } d < k\}.$$

Usually one focus on controllability at a sufficiently large time k, so we are interested in the *controllable space*

$$K(A, b) = \operatorname{Im} \begin{bmatrix} b & Ab & A^2 b & \dots \end{bmatrix}.$$

The infinite-size matrix may look daunting, but in fact we can stop at $k = n$: by the Cayley-Hamilton theorem, A^n is a linear combination of $A^{n-1}, A^{n-2}, \dots, A, I$, hence the space cannot grow after the first n iterations.

A pair (A, b) (or analogously in the matrix case (A, B)) is called *controllable* if its controllable space is the whole $\mathbb{C}^n$. This is the generic case, since for a general b the first n vectors of the form $A^k b$ are linearly independent, but it can be much smaller; for instance, if b is an eigenvector for A, we have $\dim K(A, b) = 1$.

How is controllability related to Stein equations? Consider the equation

$$X - AXA^* = bb^*. \tag{8}$$

Note that A and A^* have swapped position with respect to (1): unfortunately equations related to the system $x_{k+1} = Ax_k$ come in both variants; we always have to pay attention to where the transpose-conjugate lies.

For this equation, (6) reads

$$X = bb^* + Abb^*A^* + A^2 bb^* (A^*)^2 + \dots = \begin{bmatrix} b & Ab & A^2 b & \dots \end{bmatrix} \begin{bmatrix} b & Ab & A^2 b & \dots \end{bmatrix}^*;$$

in particular, $X \succ 0$ if and only if (A, b) is controllable: i.e., there is a vector $w \neq 0$ such that $w^* X = 0$ if and only if

$$w^* \begin{bmatrix} b & Ab & A^2 b & \dots \end{bmatrix} = 0.$$

Exercise 3. *Let $x_F \in \mathbb{C}^n$ be prescribed, and $x_0 = 0$. Show that*

$$\min |u_0|^2 + |u_1|^2 + |u_2|^2 + \dots$$

over all sequences u_k such that $\lim_{k\to\infty} x_k = x_F$ for (7) is equal to $x_F^* X^{-1} x_F$.

2.4 Iterative algorithms

We present here two iterative algorithms, which we will use to build our way towards algorithms for nonlinear equations.

The Stein equation (1) takes the form of a fixed-point equation; this fact suggests the fixed-point iteration

$$X_0 = 0, \qquad X_{k+1} = Q + A^* X_k A, \tag{9}$$

known as *Smith method* [79]. It is easy to see that the kth iterate X_k is the partial sum of (6) truncated to k terms, thus convergence is monotonic, i.e., $Q = X_0 \preceq X_1 \preceq X_2 \preceq \cdots \preceq X$. Moreover, some manipulations give

$$\begin{aligned}\operatorname{vec}(X - X_k) &= (I + M + M^2 + \dots)\operatorname{vec}(Q) - (I + M + M^2 + \cdots + M^{k-1})\operatorname{vec}(Q) \\ &= M^k (I + M + M^2 + \dots)\operatorname{vec}(Q) = M^k \operatorname{vec}(X),\end{aligned}$$

or, devectorizing,

$$X - X_k = (A^*)^k X A^k. \tag{10}$$

This relation (10) implies $\|X - X_k\| = \mathcal{O}(r^k)$ with $r = \rho(A)^2$, so convergence is linear when $\rho(A) < 1$, and it typically slows down when $\rho(A) \approx 1$.

For a dense problem, it is better to replace this algorithm with a "squaring" (or "doubling") variant. The sum of the first 2^{k+1} terms of the Neumann series can be computed iteratively using the identity

$$I+M+M^2+\cdots+M^{2^{k+1}-1} = (I+M+M^2+\cdots+M^{2^k-1})+M^{2^k}(I+M+M^2+\cdots+M^{2^k-1}),$$

without computing all the intermediate sums. Setting $\operatorname{vec} Q_k := (I + M + M^2 + \cdots + M^{2^k-1})\operatorname{vec} Q$ and $A_k := A^{2^k}$, one gets the iteration

$$A_0 = A, \qquad A_{k+1} = A_k^2, \tag{11a}$$

$$Q_0 = Q, \qquad Q_{k+1} = Q_k + A_k^* Q_k A_k. \tag{11b}$$

In view of the definitions, we have $Q_k = X_{2^k}$; so this method computes the 2^kth iterate of the Smith method directly with $\mathcal{O}(n^3 k)$ operations, without going through all intermediate iterates. Convergence is quadratic: $\|X - Q_k\| = \mathcal{O}(r^{2^k})$ with $r = \rho(A)^2$. The method (11) is known as *squared Smith*. It has been used in the context of parallel and high-performance computing [15], and reappeared in recent years, when it has been used for large and sparse equations [69, 75, 12] in combination with Krylov methods.

This is the first example of a *squaring* (or more often *doubling*) algorithm, i.e., a method to compute directly the doubling sequence $X_1, X_2, X_4, X_8, \dots$.

2.5 Sparse Stein equations

When one needs to solve a dense Stein equation, the doubling version is more performant, because of its faster convergence. However, another important case is that of the Stein equations for controllability (8), in the case when A is large and sparse. In this case, one does not want to form A_k and Q_k explicitly, since they are large and dense, typically. And, actually, we have a bigger problem: if Q_k is dense and $Q_k \to X$, then that means X is dense, too!

We can solve both problems in the non-squaring variant. That variant, for (8), reads

$$X_{k+1} = bb^* + AX_kA^*, \quad X_k = bb^* + Abb^*A^* + A^2bb^*(A^*)^2 \dots.$$

Note that

$$X_k = Z_kZ_k^*, \quad Z_k = \begin{bmatrix} b & Ab & A^2b & \cdots & A^{k-1}b \end{bmatrix} \in \mathbb{C}^{n\times k}.$$

In particular, X_k has rank k. This has an interesting theoretical consequence: we have

$$\|X - X_k\| = \mathcal{O}(r^k);$$

This has a consequence on the singular values (which are also its eigenvalues, since X is positive semidefinite):

$$\sigma_k(X) = \min_{\operatorname{rank} R=k} \|X - R\| \le \|X - X_k\| = \mathcal{O}(r^k).$$

The singular values of X decay exponentially. If r is sufficiently smaller than 1, only a handful of its singular values are above the machine precision, and we can successfully approximate X with a low-rank matrix. We can frame our iteration in terms of the Z_k:

$$\begin{cases} v_1 = b \\ v_{k+1} = Av_k \quad k = 0, 1, \dots \end{cases} \qquad Z_k = \begin{bmatrix} v_1 & v_2 & \dots & v_k \end{bmatrix}.$$

When v_k becomes sufficiently small, we can truncate the iteration, and then, $X \approx Z_kZ_k^*$. This is efficient, in practice, if r is sufficiently smaller than 1.

If convergence is slow and k becomes large, in some cases we can gain performance by compressing Z_k. We can do this with a thin SVD: we compute a thin SVD $Z_k = USV^*$ (remember: we are in the large and sparse case, so we still have $k < n$), and if some singular values are small we replace them with zeros. A slightly more efficient variant is with rank-revealing QR factorization in place of the SVD, but the idea is similar.

We shall see later a variant of this algorithm for Lyapunov equations, and we shall see there that we have some options to speed up its convergence.

2.6 Direct algorithms

Solving the $n^2 \times n^2$ system (4) directly with Gaussian elimination would cost $\mathcal{O}(n^6)$ operations, which is prohibitive even for very small n. There is a clever

idea due to Bartels and Stewart that can be used to solve equations of this kind in $\mathcal{O}(n^3)$ instead. We shall see this idea later for Lyapunov equations, since it is simpler to introduce in that case.

3 Lyapunov equations

Lyapunov equations

$$A^*X + XA + Q = 0, \quad Q = Q^* \succeq 0 \tag{12}$$

are the continuous-time counterpart of Stein equations. They arise from the study of continuous-time constant-coefficient linear systems

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = Ax(t). \tag{13}$$

Indeed, if we argue as in the discrete-time case, we can introduce an energy functional $V(x) := x^*Xx$ and show that

$$\frac{d}{dt}V(x(t)) = \dot{x}(t)^*Xx(t) + x(t)^*X\dot{x}(t) = x(t)^*(A^*X + XA)x(t) = -x(t)^*Qx(t).$$

So if $Q \succeq 0$ then the function $V(x(t))$ is non-increasing, and we can use this computation to show that in the system (13) $x(t)$ is bounded (if $X \succ 0$).

Today stability is more often proved by computing eigenvalues, but Stein equations (1) and Lyapunov equations (12) have survived in many other applications in systems and control theory, for instance in model order reduction [8, 43, 78], or as the inner step in Newton methods for other equations (see for instance (51) in the following).

The Lyapunov equation also comes in a variant that can be used to check controllability:

$$AX + XA^* + G = 0, \quad G = bb^*. \tag{14}$$

A continuous-time dynamical system with control comes in the form

$$\frac{d}{dt}x(t) = Ax(t) + bu(t),$$

where $u(t)$ is a (scalar) function that we can choose to modify the dynamics of the system. The proofs are slightly more involved in the continuous-time case.

The explicit solution formula for (14) is

$$x(t) = \int_0^\infty \exp(tA)G\exp(tA^*)\,dt,$$

and it holds assuming that all the eigenvalues of A are in the LHP. The equivalent of the formula TODO is the ugly-looking solution formula for the explicit solution of linear differential equations

$$x(t_F) = exp(At_F)x_0 + \int_0^{t_F} \exp(A(t_F - t))bu(t)dt.$$

One can show that for any t_F, x_F we can choose $u(t)$ so that $x(t_F) = x_F$, under the same condition as the one for discrete-time controllability:

$$\operatorname{rank}\begin{bmatrix} b & Ab & A^2b & \dots \end{bmatrix} = n.$$

One can prove that this rank condition holds if and only if the solution of (14) has full rank, and the solution gives information on the amount of energy required to reach x_F.

3.1 Solution properties

Using Kronecker products, one can rewrite (12) as

$$(I_n \otimes A^* + A^\top \otimes I_n) \operatorname{vec}(X) = -\operatorname{vec}(Q), \tag{15}$$

and a Schur decomposition $A = UTU^*$ produces

$$I_n \otimes A^* + A^\top \otimes I_n = (\bar{U} \otimes U)(I_n \otimes T^* + T^\top \otimes I_n)(U^\top \otimes U^*). \tag{16}$$

Again, this is a Schur-like factorization, where the middle term is lower triangular. One can tell when $I_n \otimes A^* + A^\top \otimes I_n$ is invertible by looking at its diagonal entries: that matrix is invertible (and hence (12) is uniquely solvable) if and only if $\bar{\lambda_i} + \lambda_j \neq 0$ for each pair of eigenvalues λ_i, λ_j of A. This holds, in particular, if the eigenvalues of A all lie in the left-half plane $\mathrm{LHP} = \{z \in \mathbb{C} \colon \operatorname{Re}(z) < 0\}$.

For a continuous-time linear system, the condition for asymptotic stability ($\lim_{t\to\infty} x(t) = 0$) is indeed that all the eigenvalues of A are in the LHP.

When the system is stable, an analogue of (6) is

$$X = \int_0^\infty \exp(A^* t) Q \exp(At)\, \mathrm{d}t. \tag{17}$$

Indeed, this integral converges for every choice of Q if and only if the eigenvalues of A all lie in LHP.

Notice the pleasant symmetry with the Stein case: the (discrete) sum turns into a (continuous) integral; the stability condition for discrete-time linear time-invariant dynamical systems $\rho(A) < 1$ turns into the one $\Lambda(A) \subset \mathrm{LHP}$ for continuous-time systems. We can prove analogues of all our results for Stein equations; they are sometimes more technically involved because of the presence of integrals, but the main results continue to hold with minor changes.

Perhaps a bit less evident is the equivalence between the condition $\bar{\lambda_i} + \lambda_j \neq 0$ (i.e., no two eigenvalues of A are mapped into each other by reflection with respect to the imaginary axis) and $\lambda_i \overline{\lambda_j} \neq 1$ (i.e., no two eigenvalues of A are mapped into each other by circle inversion with respect to the complex unit circle).

3.2 Direct algorithms

As in the Stein case, a direct solution by using Gaussian elimination on the $n^2 \times n^2$ system (15) costs $\mathcal{O}(n^6)$ floating point operations, and is impractical even for small values of n.

There is an algorithm due to Bartels and Stewart [6] that allows to get a solution in $\mathcal{O}(n^3)$ only.

Step 1 Reduce to a triangular equation. Take a Schur form $A = UTU^*$ to get

$$UT^*U^*X + XUTU^* + Q = 0.$$

Multiply both sides by U^* and U to get

$$T^*(U^*XU) + (U^*XU)T + (U^*QU) = 0,$$

i.e., a Lyapunov equation with triangular coefficients

$$T^*\hat{X} + \hat{X}T + \hat{Q} = 0, \quad \hat{X} = U^*XU, \quad \hat{Q} = U^*QU. \tag{18}$$

We can compute $\hat{Q}$ immediately.

Step 2 : Solve (18) with a substitution idea: we can compute the entries x_{ij} of $\hat{X}$ one at a time. Visualize the equation as

$$\begin{bmatrix} * & 0 & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ * & * & * & * \end{bmatrix} \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} + \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{bmatrix} + \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} = 0$$

We compute the (1,1) term in the LHS to get

$$\overline{t_{11}}x_{11} + x_{11}t_{11} + q_{11} = 0;$$

from this equation we can solve for x_{11}:

$$x_{11} = -\frac{q_{11}}{\overline{t_{11}} + t_{11}}.$$

Once we have computed x_{11}, we use the (2,1) entry of the equation to get x_{21}:

$$\overline{t_{21}}x_{11} + \overline{t_{22}}x_{21} + x_{21}t_{11} + q_{21} = 0,$$

i.e.,

$$x_{21} = \frac{-q_{21} - \overline{t_{21}}x_{11}}{\overline{t_{22}} + t_{11}}.$$

Similarly, one has

$$x_{ij} = \frac{-q_{ij} - \sum_{I=1}^{i-1} \overline{t_{iI}}x_{Ij} - \sum_{J=1}^{j-1} x_{iJ}t_{Jj}}{\overline{t_{ii}} + t_{jj}}.$$

Note that to compute i and j we only need entries that are *above* and *to the left* of (i, j), so we can compute the entries in the same order as vectorization.

Step 3 Once all entries of $\hat{X}$ are computed, we change basis again with $X = U\hat{X}U^*$.

It is interesting to note that this algorithm essentially correspond to inverting one by one the factors of (16):

$$\operatorname{vec}(X) = (\bar{U} \otimes U)\, \underbrace{(I_n \otimes T^* + T^\top \otimes I_n)^{-1}\, \underbrace{(U^\top \otimes U^*) \operatorname{vec}(Q)}_{=\operatorname{vec}(\hat{Q})}}_{=\operatorname{vec}(\hat{X})}.$$

This explanation motivates well why the substitution part of the algorithm works: we are basically solving a linear system with the lower triangular matrix $I_n \otimes T^* + T^\top \otimes I_n$ by forward substitution.

Remarks:

- One can prove that this algorithm works using the backward stability properties of substitution: the second step computes (in machine precision) $\widetilde{\hat{X}}$ which solves exactly a perturbed linear system

  $$(M + \Delta_M) \operatorname{vec}(\widetilde{\hat{X}}) = \operatorname{vec}(\widetilde{\hat{Q}} + \Delta_Q).$$

  This is sufficient to prove, for instance, that the (relative) residual of the equation is of the order of machine precision. It is interesting to note, though, that the perturbation Δ_M does not have the same Kronecker product structure as M; hence it is in general *not* true that $\widetilde{X}$ solves a perturbed equation $(A+\Delta_A)^*\widetilde{X} + \widetilde{X}(A+\Delta_A) + (Q+\Delta_Q) = 0$ with small Δ_A, Δ_Q. This is studied in Higham (2001) for a slightly more general equation.

- The same idea can be used for Stein equations (try it yourself!), or more generally for equations of the form $AXB + CXD = E$, with a little more work and suitable generalizations of the Schur form. System with more than two equations are a different beast; for them (as far as I know) there is no known $\mathcal{O}(n^3)$ algorithm.

3.3 Iterative algorithms

We focus now on iterative algorithms. To solve a Lyapunov equation iteratively, the main technique is turning it into a Stein equation. (This is a general observation: it is easier to find iterations to solve discrete-time problems than continuous-time ones.)

In this section, we switch to the notation of (14), since it will save us a few transpositions.

Lyapunov equations can be turned into Stein equations and *vice versa*. Indeed, for a given $\tau \in \mathbb{C}$, (12) is equivalent to

$$(A - \tau I)X(A^* - \bar{\tau} I) - (A + \bar{\tau} I)X(A^* + \tau I) - 2\operatorname{Re}(\tau) G = 0,$$

or, if $A - \tau I$ is invertible,

$$X - c(A)Xc(A)^* = 2\,\mathrm{Re}(\tau)(A-\tau I)^{-1}G(A-\tau I)^{-*}, \quad c(A) = (A+\bar\tau I)(A-\tau I)^{-1} = (A-\tau I)^{-1}(A+\bar\tau I). \tag{19}$$

If $\tau \in \mathrm{RHP}$, then the right-hand side is positive semidefinite and (19) is a Stein equation. The stability properties of $c(A)$ can be explicitly related to those of A via the following lemma.

Lemma 4 (properties of Cayley transforms)**.** *Let $\tau \in \mathrm{RHP}$. Then,*

(1) for $\lambda \in \mathbb{C}$, we have $|c(\lambda)| = \left|\frac{\lambda+\bar\tau}{\lambda-\tau}\right| < 1$ if and only if $\lambda \in \mathrm{LHP}$;

(2) for a matrix $A \in \mathbb{C}^{n\times n}$, we have $\rho(c(A)) < 1$ if and only if $\Lambda(A) \subset \mathrm{LHP}$.

A geometric argument to visualize (1) is the following. In the complex plane, τ and $= \bar\tau$ are symmetric with respect to the imaginary axis, with $-\bar\tau$ lying to its left. Thus a point $\lambda \in \mathbb{C}$ is closer to $-\bar\tau$ than to τ if and only if it lies in LHP. Part (2) follows from facts on the behaviour of eigenvalues of a matrix under rational functions [56, Proposition 1.7.3], which we will often use also in the following.

We will assume $\Lambda(A) \subset \mathrm{LHP}$. Then, thanks to Lemma 4, we have $\rho(c(A)) < 1$, so we can apply the Smith method (9) to (19).

$$X_0 = 0, \quad X_{k+1} = \hat G + c(A)X_k c(A)^*, \quad \hat G = 2\,\mathrm{Re}(\tau)(A-\tau I)^{-1}G(A-\tau I)^{-*}. \tag{20}$$

As we already know, convergence is linear with rate r^2, where $r = \rho(c(A))$. It is interesting to note which is the better value of τ to use. If λ is an eigenvalue of A, then the corresponding eigenvalue of $c(A)$ is

$$c(\lambda) = \frac{\lambda+\bar\tau}{\lambda-\tau}.$$

If $\tau \gg |\lambda|$, then $c(\lambda) \approx \frac{\bar\tau}{\tau}$, which is on the unit circle. If $\tau \ll |\lambda|$, then $c(\lambda) \approx 1$, which is also on the unit circle. So we can get fast convergence only when τ has the same order of magnitude as the eigenvalues of A. In particular, if A has both large and small eigenvalues, convergence is always going to be slow, because

$$\rho(A) = \max_{\lambda\in\Lambda(A)} |c(\lambda)|$$

is going to be close to 1 no matter how we choose τ.

To solve this issue, the solution is changing the value of τ at each iteration. The resulting algorithm is known as *ADI iteration* [68, 80]:

$$X_0 = 0, \qquad X_{k+1} = Q_k + c_k(A)^* X_k c_k(A), \tag{21}$$
$$Q_k = 2\,\mathrm{Re}(\tau_k)(A^*-\tau_k I)^{-1}Q(A-\bar\tau_k I)^{-1}, \quad c_k(A) = (A+\tau_k I)(A-\bar\tau_k I)^{-1} = (A-\bar\tau_k I)^{-1}(A+\tau_k I).$$

The sequence of *shifts* $\tau_k \in$ RHP can be chosen arbitrarily. Typically, one chooses a fixed number of shifts (say, d), and uses them cyclically, so that the factorization of $A - \tau_k I$ can be reused.

By writing a recurrence for the error $E_k = X - X_k$, one sees that

$$E_k = r_{k+1}(A)^* E_0 r_{k+1}(A) = r_{k+1}(A)^* X r_{k+1}(A), \quad r_{k+1}(A) = c_k(A) \dots c_1(A) c_0(A), \tag{22}$$

a formula which generalizes (10). If we reuse shifts cyclically, then every d steps we have multiplied the error by $r_d(A)$.

When A is normal, the problem of assessing the convergence speed of this iteration can be reduced to a scalar approximation theory problem. Note that

$$\|r_d(A)\| = \max_{\lambda \in \Lambda(A)} |r_d(\lambda)|, \qquad\qquad r_d(\lambda) =$$

If one knows a region $E \subset$ LHP that encloses the eigenvalues of A, the optimal choice of r_d is the degree-d rational function that minimizes

$$\frac{\sup_{z \in E} |r_d(z)|}{\inf_{z \in -E^*} |r_d(z)|}, \tag{23}$$

i.e., a rational function that is 'as large as possible' on E and 'as small as possible' on $-E^*$. Finding this rational function is known as *Zolotarev approximation problem*, and it was solved by its namesake for many choices of E, including $E = [a, b] \subseteq \mathbb{R}_+$: this choice of E corresponds to having a symmetric positive definite A for which a lower and upper bound on the spectrum are known. It is known that the optimal ratio (23) decays as ρ^d, where $\rho < 1$ is a certain value that depends on E, related to its so-called *logarithmic capacity*. See the recent review by Beckermann and Townsend [7] for more details. Optimal choices for the shifts for a normal A were originally studied by Wachspress [80, 38]. When A is non-normal, a similar bound can be obtained from its eigendecomposition $A = VDV^{-1}$, but it includes its eigenvalue condition number $\kappa(V) = \|V\|\|V\|^{-1}$, and thus it is of worse quality.

An important case, both in theory and in practice, is when the constant term Q has low rank. To study this case, we switch to the "controllability version" of the Lyapunov equation

$$AX + XA^* + G = 0, \quad G = bb^*$$

(We restrict for simplicity and ease of exposition to the case where b has rank 1; the case of larger rank is similar.)

A decomposition $X_k = Z_k Z_k^*$ can be derived from (21), and reads

$$\begin{aligned} Z_k &= \left[\sqrt{2\,\mathrm{Re}(\tau_{k-1})}(A - \tau_{k-1}I)^{-1}b, \quad c_{k-1}(A) Z_{k-1}\right] \\ &= \Big[\sqrt{2\,\mathrm{Re}(\tau_{k-1})}(A - \tau_{k-1}I)^{-1}b, \sqrt{2\,\mathrm{Re}(\tau_{k-2})}(A - \tau_{k-1}I)^{-1}(A + \bar{\tau}_{k-1}I)(A - \tau_{k-2}I)^{-1}b, \dots, \\ &\quad \sqrt{2\,\mathrm{Re}(\tau_0)}(A - \tau_{k-1}I)^{-1}(A + \bar{\tau}_{k-1}I)(A - \tau_{k-2}I)^{-1}(A + \bar{\tau}_{k-2}I) \cdots (A - \tau_0 I)^{-1}b\Big]. \end{aligned} \tag{24}$$

Hence Z_k is obtained by concatenating horizontally k terms $V_1, V_2, \dots, V_k$ of size $n \times p$ each. Each of them contains a rational function of A of increasing degree multiplied by b. All the factors in parentheses commute: hence that the factors V_j can be computed with the recurrence

$$\begin{aligned} Z_k = \begin{bmatrix} V_1 & V_2 & \cdots & V_k \end{bmatrix}, \qquad V_1 &= \sqrt{2\operatorname{Re}(\tau_{k-1})}(A - \tau_{k-1} I)^{-1} b, \\ V_{j+1} &= \frac{\sqrt{2\operatorname{Re}(\tau_{k-j-1})}}{\sqrt{2\operatorname{Re}(\tau_{k-j})}}(A - \tau_{k-j-1} I)^{-1}(A + \bar{\tau}_{k-j} I) V_j \\ &= \frac{\sqrt{2\operatorname{Re}(\tau_{k-j-1})}}{\sqrt{2\operatorname{Re}(\tau_{k-j})}} \left(V_j + (\tau_{k-j-1} + \bar{\tau}_{k-j})(A + \tau_{k-j-1} I)^{-1} V_j \right). \end{aligned} \tag{25}$$

This version of ADI is known as *low-rank ADI (LR-ADI)* [13]. After k steps, $X_k = Z_k Z_k^*$, but note that in the intermediate steps $j < k$ the quantity $\begin{bmatrix} V_1 & V_2 & \cdots & V_j \end{bmatrix} \begin{bmatrix} V_1 & V_2 & \cdots & V_j \end{bmatrix}^*$ differs from X_j in (21). Indeed, in this factorized version the shifts appear in reversed order, starting from τ_{k-1} and ending with τ_0. Nevertheless, we can use LR-ADI as an iteration in its own right: since we keep adding columns to Z_k at each step, $Z_k Z_k^*$ converges monotonically to X. This version is particularly convenient for problems in which A is large and sparse, because in each step we only need to solve a linear system with a shifted matrix $A - \tau I$, and we store in memory only the $n \times k$ matrix Z_k. In contrast, iterations such as (11) are not going to be efficient for problems with a large and sparse A, since powers of sparse matrices become dense.

The formula (24) displays the relationship between ADI and certain Krylov methods: since the LR-ADI iterates are constructed by applying rational functions of A iteratively to b, the LR-ADI iterate Z_k lies in the so-called *rational Krylov subspace* [74]

$$K_{q,k+1}(A, b) = \operatorname{span}\{q(A)^{-1} p(A) b : p \text{ is a polynomial of degree} \le k\}, \tag{26}$$

constructed with *pole polynomial* $q(z) = (z - \tau_0)(z - \tau_1) \cdots (z - \tau_{k-1})$. This suggests a different view: what is important is not the form of the ADI iteration, but rather the approximation space $K_{q,k}(A, b)$ to which its iterates belong. Once one has chosen suitable shifts and computed an orthogonal basis U_k of $K_{q,k+1}(A, b)$, (12) can be solved via *Galerkin projection*: we seek an iterate X_k of the form $X_k = U_k Y_k U_k^*$, and compute Y_k by solving the projected equation

$$0 = U_k^*(AX_k + X_k A^* + G)U = (U_k^* A U_k) Y_k + Y_k (U_k^* A^* U_k) + U_k^* G U_k,$$

which is a smaller $(k \times k)$ Lyapunov equation.

While the approximation properties of classical Krylov subspaces are related to polynomial approximation, those of rational Krylov subspaces are related to approximation with rational functions, as in the Zolotarev problem mentioned earlier. In many cases, rational approximation has better convergence properties, with an appropriate choice of the shifts. This happens also for Lyapunov

equations: algorithms based on rational Krylov subspaces (26) [37, 36] (including ADI which uses them implicitly) often display better convergence properties than equivalent ones in which U_k is chosen as a basis of a regular Krylov subspace.

Computing a basis for a rational Krylov subspace (26) is more expensive than computing one for a Krylov subspace: indeed, the former requires solving linear systems with $A - \tau_k I$ for many values of k, while the latter uses multiple linear systems with the same matrix A. However, typically, their faster convergence more than compensates for it. Another remarkable feature is the possibility to use an adaptive procedure based on the residual for shift selection [37].

See also the analysis in Benner, Li, Truhar [14], which shows that Galerkin projection can improve also on the ADI solution.

An important consequence of the convergence of these algorithms is that they can be used to give bounds on the rank of the solution X. Since we can find rational functions such that (23) decreases exponentially, the formula (22) shows that X can be approximated well with X_k, which has rank at most $k \cdot \operatorname{rank}(Q)$ in view of the decomposition (24). This observation has practical relevance, since in many applications p is very small, and the exponential decay in the singular values of X is very well visible and helps reducing the computational cost.

3.4 Remarks

There is vast literature already for linear matrix equations, especially when it comes to large and sparse problems. We refer the reader to the review by Simoncini [78] for more details. The literature typically deals with continuous-time Lyapunov equations more often than their discrete-time counterpart; however, Cayley transformations (19) can be used to convert one to the other.

In particular, it follows from our discussion that a step of ADI can be interpreted as transforming the Lyapunov equation (12) into a Stein equation (1) via a Cayley transform (19) and then applying one step of the Smith iteration (9). Hence the squared Smith method (11) can be interpreted as a doubling algorithm to construct the ADI iterate X_{2^k} in k iterations only, but with the significant limitation of using *only one shift* τ in ADI.

It is known that a wise choice of shifts has a major impact on the convergence speed of these algorithms; see e.g. Güttel [47]. A major challenge for doubling-type algorithms seems incorporating multiple shifts in this framework of repeated squaring. It seems unlikely that one can introduce more than one shift per doubling iteration, but even doing so would be an improvement, allowing one to leverage the theory of rational approximation that underlies ADI and Krylov space methods.

4 Optimal control and algebraic Riccati equations

The motivation for the next, more complicated class of equations comes once again from control systems. Consider either (7) or its continuous-time equivalent

$$\frac{d}{dt}x(t) = Ax(t) + bu(t),$$

where $u(t)$ is an arbitrary function that we add at each time t to modify ("control") the dynamics of the system. There are many choices of the sequence u_k, or the function $u(t)$, that make the corresponding system stable. An "optimal" one can be defined: given an "energy" function $Q \succeq 0$, we wish to minimize

$$\min \sum_{k=1}^{\infty} x_k^* Q x_k + u_k^* u_k$$

or respectively

$$\min \int_0^{\infty} (x(t)^* Q x(t) + u(t)^* u(t))\ dt$$

over all possible control functions. This minimization problem makes sense in terms of modelling: we want $x(t)$ to be as close as possible to 0 at all times, but we also have a cost to pay for our control function $u(t)$.

It turns out that the optimal control can be constructed using a solution X to a certain nonlinear matrix equation; for continuous systems, this equation takes the form (CARE, *continuous-time algebraic Riccati equation*)

$$Q + A^*X + XA - XGX = 0, \quad G = G^* = bb^* \succeq 0, \quad Q = Q^* \succeq 0, \quad A, G, Q, X \in \mathbb{C}^{n\times n}, \tag{27}$$

with the controlled system given by $\dot{x}(t) = Ax(t) + bu(t) = (A - GX)x(t)$, so $u(t) = -b^*x(t)$. We do not see a full proof here; there is one in [34, Chapter 10]. The main idea is noting that $x^*Qx + u^*u = x^*(Q + XGX)x = x^*((A-GX)^*X + X(A - GX))x$, which is the derivative of the energy function x^*Xx. Another proof based on calculus of variations techniques is in [64].

The corresponding equation for discrete-time system (the DARE, *discrete-time algebraic Riccati equation*) looks uglier:

$$X = Q + A^*X(I + GX)^{-1}A \quad G = G^* = bb^* \succeq 0, \quad Q = Q^* \succeq 0, \quad A, G, Q, X \in \mathbb{C}^{n\times n}. \tag{28}$$

Despite the very different form, this equation is a natural analogue of the CARE (27), exactly like Stein and Lyapunov equations are related to each other.

The corresponding controlled system is $x_{k+1} = Ax_k + Bu_k = (I+GX)^{-1}Ax_k$. This time it is a little less obvious that $(I+GX)^{-1}A = A + bf^*$ for a certain vector f^*; we can convince ourselves by expanding the inverse with the Neumann

series, assuming that $\rho(GX) < 1$:

$$\begin{aligned}(I+GX)^{-1}A &= (I - GX + GXGX - GXGXGX + \dots)A \\ &= A - GXA + GXGXA - GXGXGXA + \dots \\ &= A + b(-b^*XA + b^*Xbb^*XA - \dots).\end{aligned}$$

Even if the assumption $\rho(GX) < 1$ does not hold, one can use the identity

$$(I+GX)^{-1} = I - GX(I+GX)^{-1}$$

to get the same result. Due to the fact that there are several different ways to rewrite this inverse, the DARE can take many slightly different forms. If you read $X = Q + \text{something ugly}$, it is probably a DARE.

(One can generalize this computation to more than one control, and then another matrix R appears: u^*Ru is the u part of the "energy functional", and $G = BR^{-1}B^*$).

4.1 Relation to invariant subspaces

There is a beautiful chapter of linear algebra behind the solutions of the CARE (27) (and the DARE, as we see later). It all starts by rewriting it into the form

$$\begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X \end{bmatrix} M, \qquad M = A - GX. \tag{29}$$

This equation looks like a "fat version" of the eigenvalue-eigenvector relation $\mathcal{H}v = v\lambda$ for the matrix

$$\mathcal{H} = \begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix}. \tag{30}$$

Indeed, one can show that the two problems are related: if $Mw = w\lambda$ is an eigenvector/eigenvalue pair for M, then $\left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right]w, \lambda$ is an eigenvalue / eigenvector pair for $\mathcal{H}$.

This observation suggests a way to compute X. First, suppose for simplicity that $\mathcal{H}$ is diagonalizable. Then, select n out of its $2n$ eigenvalue/eigenvector pairs. In this way, $u_1, \dots, u_n$ are linear combinations of the columns of $\left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right]$; symmetrically, the columns of $\left[\begin{smallmatrix} I \\ X \end{smallmatrix}\right]$ are linear combinations of the columns of $U = [u_1, \dots, u_n]$, since the two column spaces are n-dimensional. So, if

$$U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \in \mathbb{C}^{2n \times n}, \quad X = U_2 U_1^{-1}.$$

Equation (29) is called an *invariant subspace relation*, since $\operatorname{Im} U$ is $\mathcal{H}$-invariant, i.e., $\mathcal{H} \operatorname{Im} U \subseteq \operatorname{Im} U$.

It remains to determine which eigenvalues and eigenvectors of $\mathcal{H}$ we have to select: from every choice of n eigenvectors out of $2n$ determines a solution of the CARE (assuming U_1 is invertible), but only one is the one we are looking for. If $\mathcal{H}$ has repeated eigenvalues, there can even be an infinite number of solutions.

The choice of solution comes from stability consideration: if the system with its optimal control is $\dot{x}(t) = (A - GX)x(t)$, we must have $\Lambda(A - GX) \subseteq LHP$, otherwise the system is not stable and we cannot (usually) hope that $\int x^*Qx + u^*u$ is finite.

This condition is sufficient, because luckily $\mathcal{H}$ has a remarkable property.

4.2 Hamiltonian property

Lemma 5. *Let*

$$\mathcal{H} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \in \mathbb{C}^{2n \times 2n}$$

be any matrix such that $H_{12} = H_{12}^*$, $H_{21} = H_{21}^*$, $H_{22} = -H_{11}^*$. Then, if λ is an eigenvalue of $\mathcal{H}$, then so is $-\overline{\lambda}$, and the two have the same multiplicity.

Proof. Consider the antisymmetric orthogonal matrix

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

for which we have $J^{-1} = J^* = -J$. Then, $J\mathcal{H}$ is Hermitian, i.e., $J\mathcal{H} = H^*J^*$, or, equivalently, $\mathcal{H}^* = -J\mathcal{H}J^{-1}$. In particular, $\mathcal{H}$ and $-\mathcal{H}^*$ are similar, and they must have the same spectrum; the thesis follows from this. □

Geometrically, the spectrum $\Lambda(\mathcal{H})$ is symmetric with respect to the imaginary axis. This property is important, because it shows that we can expect to have exactly n out of $2n$ eigenvalues of $\mathcal{H}$ in the left half-plane; i.e., the solution X that we are looking for is uniquely determined by choosing these n eigenvalues.

There are still a few points to settle to get a formal theorem; in particular, we need to exclude the possibility that $\mathcal{H}$ has eigenvalues exactly on the imaginary axis, since those do not come in pair (and are not considered "stable" nor "in the left half-plane" for our purposes). To do this, we need a few extra controllability assumptions.

Theorem 6. *[56, Theorems 7.9.1, 9.1.2 and 9.1.5] Assume that $Q \succeq 0$, $G \succeq 0$, (A, G) (or equivalently (A, b)) is controllable, and (A^*, Q) is controllable. Then,* (27) *has a (unique) solution X_+ such that*

(1) $X_+ = X_+^* \succeq 0$;

(2) $X_+ \succeq X$ for any other Hermitian solution X;

(3) $\Lambda(A - GX_+) \subset \mathrm{LHP}$ (and, more precisely, $\Lambda(A - GX_+) = \Lambda(\mathcal{H}) \cap \mathrm{LHP}$).

For a full proof of this theorem, we refer the reader to Lancaster and Rodman [56].

We also note that if X solves the CARE then we have the factorization

$$\begin{bmatrix} I & 0 \\ -X & I \end{bmatrix} \mathcal{H} \begin{bmatrix} I & 0 \\ X & I \end{bmatrix} = \begin{bmatrix} A - GX & -G \\ 0 & -(A - GX)^* \end{bmatrix},$$

which shows clearly how the eigenvalues come in conjugate pairs.

4.3 Algorithms

We shall discuss iterative algorithms in the following, when speaking about discrete-time equations, but first we see a direct algorithm.

We have given a characterization of the solution based on eigenvectors; however, the problem of computing eigenvalues can be numerically problematic, in cases when the eigenvector matrix V is ill-conditioned. Luckily, one can also extract an invariant subspace of $\mathcal{H}$ through the Schur factorization $\mathcal{H} = UTU^*$, which can be computed with a backward stable algorithm (since U is unitary).

Lemma 7. *Let $\mathcal{H}$ be a Hamiltonian matrix as above, and assume that $\mathcal{H}$ has n eigenvalues in* LHP *and n in* RHP*. Let $\mathcal{H} = UTU^*$ be its Schur factorization, and write*

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}, \quad T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}.$$

Assume that the eigenvalues are sorted so that $\Lambda(T_{11}) \subset$ LHP and $\Lambda(T_{22}) \subset$ RHP. Then, the columns of U_1 are a basis for the (unique) invariant subspace associated with the eigenvalues in the LHP*.*

Luckily, Matlab (as well as Julia, Numpy,...) has a function to reorder the Schur form.

```
function X = care_schur(A, G, Q)
H = [A -G; -Q -A'];
[U, T] = schur(H, 'complex');
[U2, T2] = ordschur(U, T, 'lhp');
n = size(A,1);
X = U2(n+1:2*n, 1:n) / U2(1:n, 1:n);
```

We show an example, which relies on the `carex` test suite, a collection of benchmark examples for algebraic Riccati equations.

```
>> [A,G,Q] = carex(4);
>> X = care_schur(A, G, Q)
X =
    0.8919 0.7366 0.6023 0.5212 0.5929 0.3488 0.2199 0.1415
    0.7366 1.3795 1.0765 0.8039 0.7005 0.5191 0.3348 0.1744
    0.6023 1.0765 1.4920 1.0138 0.8014 0.7435 0.4192 0.2031
    0.5212 0.8039 1.0138 1.1488 0.7327 0.5313 0.3410 0.1732
    0.5929 0.7005 0.8014 0.7327 0.5921 0.4293 0.2847 0.1476
    0.3488 0.5191 0.7435 0.5313 0.4293 0.3553 0.2377 0.1241
    0.2199 0.3348 0.4192 0.3410 0.2847 0.2377 0.1965 0.1024
    0.1415 0.1744 0.2031 0.1732 0.1476 0.1241 0.1024 0.0795
>> norm(A'*X + X*A + Q - X*G*X) / (norm(A'*X) + norm(X*A) + norm(Q) + norm(X*G*X))
ans =
    3.4242e-15
>> max(real(eig(A - G*X)))
ans =
    -0.1006
```

Note that the method produces a symmetric stabilizing solution X; we already know that this must be the case.

5 Discrete-time Riccati equations

We now consider (28), i.e.,

$$X = Q + A^*X(I + GX)^{-1}A \quad G = G^* = bb^* \succeq 0, \quad Q = Q^* \succeq 0, \quad A, G, Q, X \in \mathbb{C}^{n\times n}, \tag{31}$$

to be solved for $X = X^* \succeq 0$. Variants in which G, Q are not necessarily positive semidefinite also exist [71, 82], but we will not deal with them here to keep our presentation simpler. The non-linear term can appear in various slightly different forms: for instance, if $G = BR^{-1}B^*$ for certain matrices $B \in \mathbb{C}^{n\times m}, R \in \mathbb{C}^{m\times m}$, $R = R^* \succ 0$, then one sees with some algebra that

$$\begin{aligned} X(I + GX)^{-1} &= (I + XG)^{-1}X = X - X(I + GX)^{-1}GX \\ &= X - XBR^{-1/2}(I + R^{-1/2}B^*XBR^{-1/2})^{-1}R^{-1/2}B^*X \\ &= X - XB(R + B^*XB)^{-1}B^*X, \end{aligned} \tag{32}$$

and all these forms can be plugged into (28) to obtain a slightly different (but equivalent) equation. In particular, from the versions in the last two rows one sees that $X(I + GX)^{-1}$ is Hermitian, which is not evident at first sight. These identities become clearer if one considers the special case in which $\rho(GX) < 1$: in this case, one sees that the expressions in (32) are all different ways to rewrite the sum of the converging series $X - XGX + XGXGX - XGXGXGX + \dots$.

Note that the required inverses exist under our assumptions, because the eigenvalues of GX coincide with those of $G^{1/2}XG^{1/2} \succeq 0$.

5.1 Solution properties

For convenience, we assume in the following that A is invertible. The results in this section hold also when it is singular, but to formulate them properly one must deal with matrix pencils, infinite eigenvalues, and generalized invariant subspaces (or *deflating subspaces*), a technical difficulty that we would rather avoid here since it does not add much to our presentation. For a more general pencil-based presentation, see for instance Mehrmann [62].

For each solution X of the DARE (28), it holds that

$$\begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I & G \\ 0 & A^* \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} K, \qquad K = (I + GX)^{-1}A. \tag{33}$$

Equation (33) shows that $\mathrm{Im}[\begin{smallmatrix} I \\ X \end{smallmatrix}]$ is an *invariant subspace* of

$$\mathcal{S} = \begin{bmatrix} I & G \\ 0 & A^* \end{bmatrix}^{-1} \begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix}, \tag{34}$$

i.e., $\mathcal{S}$ maps this subspace into itself. In particular, the n eigenvalues (counted with multiplicity) of K are a subset of the $2n$ eigenvalues of $\mathcal{S}$: this can be seen by noticing that the matrix K represents (in a suitable basis) the linear operator $\mathcal{S}$ when restricted to said subspace. Conversely, if one takes a basis matrix $\left[\begin{smallmatrix} U_1 \\ U_2 \end{smallmatrix}\right]$ for an invariant subspace of $\mathcal{S}$, and if U_1 is invertible, then $\left[\begin{smallmatrix} I \\ U_2 U_1^{-1} \end{smallmatrix}\right]$ is another basis matrix, the equality (33) holds, and $X = U_2 U_1^{-1}$ is a solution of (28). Hence, (28) typically has multiple solutions, each associated to a different invariant subspace. However, among them there is a preferred one, which is the one typically sought in applications.

Theorem 8. *[56, Corollary 13.1.2 and Theorem 13.1.3] Assume that $Q \succeq 0$, $G \succeq 0$ and (A, G) is d-stabilizable. Then,* (28) *has a (unique) solution X_+ such that*

(1) $X_+ = X_+^* \succeq 0$;

(2) $X_+ \succeq X$ for any other Hermitian solution X;

(3) $\rho\left((I + GX_+)^{-1}A\right) \leq 1$.

If, in addition, (Q, A) is d-detectable, then $\rho\left((I + GX_+)^{-1}A\right) < 1$.

The hypotheses involve two classical definitions from control theory [34]: *d-stabilizable* (resp. *d-detectable*) means that all Jordan chains of A (resp. A^*) that are associated to eigenvalues *outside* the set $\{|\lambda| < 1\}$ are contained in the maximal (block) Krylov subspace $\operatorname{span}(B, AB, A^2B, \dots)$ (resp. $\operatorname{span}(C^*, A^*C^*, (A^*)^2C^*, \dots)$). We do not discuss further these hypotheses nor the theorem, which is not obvious to prove; we refer the reader to Lancaster and Rodman [56] for details, and we just mention that these hypotheses are typically satisfied in control theory applications. This solution X_+ is often called *stabilizing* (because of property 3) or *maximal* (because of property 2).

Various properties of the matrix $\mathcal{S}$ in (34) follow from the fact that it belongs to a certain class of structured matrices. Let $J = \left[\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right] \in \mathbb{C}^{2n \times 2n}$. A matrix $M \in \mathbb{C}^{2n \times 2n}$ is called *symplectic* if $M^* J M = J$, i.e., if it is unitary for the non-standard scalar product associated to J. The following properties hold.

Lemma 9. *(1) A matrix in the form* (34) *is symplectic if and only if $G = G^*, Q = Q^*$, and the two blocks called A, A^* in* (34) *are one the conjugate transpose of the other.*

(2) If λ is an eigenvalue of a symplectic matrix with right eigenvector v, then $\overline{\lambda}^{-1}$ is an eigenvalue of the same matrix with left eigenvector v^*J.

(3) Under the hypotheses of Theorem 8 (including the d-detectability one in the end), then the $2n$ eigenvalues of $\mathcal{S}$ are (counting multiplicities) the n eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ of $(I + GX_+)^{-1}A$ inside the unit circle, and the n eigenvalues $\overline{\lambda_i}^{-1}$, $i = 1, 2, \dots, n$ outside the unit circle. In particular, $\left[\begin{smallmatrix} I \\ X_+ \end{smallmatrix}\right]$ spans the unique invariant subspace of $\mathcal{S}$ of dimension n all of whose associated eigenvalues lie in the unit circle.

Parts 1 and 2 are easy to verify from the form (34) and the definition of symplectic matrix, respectively. To prove Part 3, plug X_+ into (33) and notice that K has n eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ inside the unit circle; these are also eigenvalues of $\mathcal{S}$. By Part 2, all other eigenvalues lie outside the unit circle.

5.2 Algorithms

The shape of (28) suggests the iteration

$$X_{k+1} = Q + A^* X_k (I + G X_k)^{-1} A, \qquad X_0 = 0. \tag{35}$$

This iteration can be rewritten in a form analogous to (33):

$$\begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} \begin{bmatrix} I \\ X_{k+1} \end{bmatrix} = \begin{bmatrix} I & G \\ 0 & A^* \end{bmatrix} \begin{bmatrix} I \\ X_k \end{bmatrix} K_k, \qquad K_k = (I + G X_k)^{-1} A. \tag{36}$$

Equivalently, one can write it as

$$\begin{bmatrix} U_{1k} \\ U_{2k} \end{bmatrix} = \mathcal{S}^{-1} \begin{bmatrix} I \\ X_k \end{bmatrix}, \qquad \begin{bmatrix} I \\ X_{k+1} \end{bmatrix} = \begin{bmatrix} U_{1k} \\ U_{2k} \end{bmatrix} (U_{1k})^{-1}. \tag{37}$$

This form highlights a connection with (inverse) subspace iteration (or orthogonal iteration), a classical generalization of the (inverse) power method to find multiple eigenvalues [81]. Indeed, we start from the $2n \times n$ matrix $\left[\begin{smallmatrix} I \\ X_0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} I \\ 0 \end{smallmatrix}\right]$, and at each step we first multiply it by $\mathcal{S}^{-1}$, and then we normalize the result by imposing that the first block is I. In inverse subspace iteration, we would make the same multiplication, but then we would normalize the result by taking the Q factor of its QR factorization, instead.

It follows from classical convergence results for the subspace iteration (see e.g. Watkins [81, Section 5.1]) that (37) converges to the invariant subspace associated to the n largest eigenvalues (in modulus) of $\mathcal{S}^{-1}$, i.e., the n smallest eigenvalues of $\mathcal{S}$. In view of Part 3 of Lemma 9, this subspace is precisely $\operatorname{Im}\left[\begin{smallmatrix} I \\ X_+ \end{smallmatrix}\right]$. Note that this unusual normalization is not problematic, since at each step of the iteration (and in the limit) the subspace does admit a basis in which the first n rows form an identity matrix. This argument shows the convergence of (35) to the maximal solution, under the d-detectability condition mentioned in Theorem 8, which ensures that there are no eigenvalues on the unit circle.

How would one construct a 'squaring' variant of this method? Note that that $\left[\begin{smallmatrix} U_{1k} \\ U_{2k} \end{smallmatrix}\right] = \mathcal{S}^{-k} \left[\begin{smallmatrix} I \\ 0 \end{smallmatrix}\right]$; hence one can think of computing $\mathcal{S}^{-2^k}$ by iterated squaring to obtain X_{2^k} in k steps. However, this idea would be problematic numerically, because it amounts to delaying the normalization in subspace iteration until the very last step. The key to solve this issue is using the LU-like decomposition obtained from (34)

$$\mathcal{S}^{-1} = \begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix}^{-1} \begin{bmatrix} I & G \\ 0 & A^* \end{bmatrix}.$$

We seek an analogous decomposition for the powers of $\mathcal{S}^{-1}$, i.e.,

$$\mathcal{S}^{-2^k} = \begin{bmatrix} A_k & 0 \\ -Q_k & I \end{bmatrix}^{-1} \begin{bmatrix} I & G_k \\ 0 & A_k^* \end{bmatrix}. \tag{38}$$

The following result shows how to compute this factorization with just one matrix inversion.

Lemma 10. *[70] Let $M_1, M_2, N_1, N_2 \in \mathbb{C}^{2n\times n}$. The factorization*

$$\begin{bmatrix} M_1 & M_2 \end{bmatrix}^{-1} \begin{bmatrix} N_1 & N_2 \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & I_n \end{bmatrix}^{-1} \begin{bmatrix} I_n & A_{21} \\ 0 & A_{22} \end{bmatrix}, \quad A_{11}, A_{12}, A_{21}, A_{22} \in \mathbb{C}^{n\times n} \tag{39}$$

exists if and only if $\begin{bmatrix} N_1 & M_2 \end{bmatrix}$ is invertible, and in that case its blocks A_{ij} are given by

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} N_1 & M_2 \end{bmatrix}^{-1} \begin{bmatrix} M_1 & N_2 \end{bmatrix}.$$

A proof follows from noticing that the factorization (39) is equivalent to the existence of a matrix $K \in \mathbb{C}^{2n\times 2n}$ such that

$$K \begin{bmatrix} M_1 & M_2 & N_1 & N_2 \end{bmatrix} = \begin{bmatrix} A_{11} & 0 & I_n & A_{12} \\ A_{21} & I_n & 0 & A_{22} \end{bmatrix},$$

and rearranging block columns in this expression.

One can apply Lemma 10 (with $[M_1\ M_2] = I$ and $[N_1\ N_2] = \left[\begin{smallmatrix} I & G_k \\ 0 & A_k^* \end{smallmatrix}\right] \left[\begin{smallmatrix} A_k & 0 \\ -Q_k & I \end{smallmatrix}\right]^{-1}$) to find a factorization of the term in parentheses in

$$\mathcal{S}^{-2^{k+1}} = \mathcal{S}^{-2^k}\mathcal{S}^{-2^k} = \begin{bmatrix} A_k & 0 \\ -Q_k & I \end{bmatrix}^{-1} \left(\begin{bmatrix} I & G_k \\ 0 & A_k^* \end{bmatrix} \begin{bmatrix} A_k & 0 \\ -Q_k & I \end{bmatrix}^{-1} \right) \begin{bmatrix} I & G_k \\ 0 & A_k^* \end{bmatrix}, \tag{40}$$

and use it to construct a decomposition (38) of $\mathcal{S}^{-2^{k+1}}$ starting from that of $\mathcal{S}^{-2^k}$. The fact that the involved matrices are symplectic can be used to prove that the relations $A_{11} = A_{22}^*$, $A_{21} = A_{21}^*$, $A_{12} = A_{12}^*$ will hold for the computed coefficients. We omit the details of this computation; what matters are the resulting formulas

$$A_{k+1} = A_k(I + G_kQ_k)^{-1}A_k, \tag{41a}$$

$$G_{k+1} = G_k + A_kG_k(I + Q_kG_k)^{-1}A_k^*, \tag{41b}$$

$$Q_{k+1} = Q_k + A_k^*(I + Q_kG_k)^{-1}Q_kA_k, \tag{41c}$$

with $A_0 = A, Q_0 = Q, G_0 = G$. These formulas are all we need to formulate a 'squaring' version of (35): for each k it holds that

$$\mathcal{S}^{-2^k} \begin{bmatrix} I_n \\ 0 \end{bmatrix} = \begin{bmatrix} I \\ Q_k \end{bmatrix} A_k^{-1},$$

hence $Q_k = X_{2^k}$, the 2^kth iterate of (35). It is not difficult to show by induction that $0 \preceq Q_0 \preceq Q_1 \preceq \cdots \leq Q_k \preceq \ldots$, and we have already argued above that $Q_k = X_{2^k} \to X_+$. In view of the interpretation as subspace iteration, the

convergence speed of (35) is linear and proportional to the ratio between the absolute values of the $(n+1)$st and nth eigenvalue of $\mathcal{S}$, i.e., between $\sigma := \rho((I+GX_+)A) < 1$ and its inverse σ^{-1}. The convergence speed of its doubling variant (41) is then quadratic with the same ratio [49].

The iteration (41), which goes under the name of *structure-preserving doubling algorithm*, has been used to solve DAREs and related equations by various authors, starting from Chu, Fan, Lin and Wang [31], but it also appears much earlier: for instance, Anderson [2] gave it an explicit system-theoretical meaning as constructing an equivalent system with the same DARE solution. The reader may find in the literature slightly different versions of (41), which are equivalent to them thanks to the identities (32).

More general versions of the factorization (38) and of the iteration (41), which guarantee existence and boundedness of the iterates under much weaker conditions, have been explored by Mehrmann and Poloni [63]. Kuo, Lin and Shieh [55] studied the theoretical properties of the factorization (38) for general powers $\mathcal{S}^t$, $t \in \mathbb{R}$, drawing a parallel with the so-called *Toda flow* for the QR algorithm.

The limit of the monotonic sequence $0 \preceq G_0 \preceq G_1 \preceq G_2 \preceq \dots$ also has a meaning: it is the maximal solution Y_+ of the so-called *dual equation*

$$Y = G + AY(I+QY)^{-1}A^*, \tag{42}$$

which is obtained swapping Q with G and A with A^* in (28). Indeed, SDA for the DARE (42) is obtained by swapping Q with G and A with A^* in (41), but this transformation leaves the formulas unchanged. The dual equation (42) appears sometimes in applications together with (28). From the point of view of linear algebra, the most interesting feature of its solution Y_+ is that $\left[\begin{smallmatrix} -Y_+ \\ I \end{smallmatrix}\right]$ is a basis matrix for the invariant subspace associated to the other eigenvalues of $\mathcal{S}$, those outside the unit circle. Indeed, (38) gives

$$\mathcal{S}^{2^k} \begin{bmatrix} 0 \\ I \end{bmatrix} = \begin{bmatrix} -G_k \\ I_n \end{bmatrix} A_k^{-*},$$

so $\left[\begin{smallmatrix} -Y_+ \\ I \end{smallmatrix}\right]$ is the limit of subspace iteration applied to $\mathcal{S}$ instead of $\mathcal{S}^{-1}$, with initial value $\left[\begin{smallmatrix} 0 \\ I \end{smallmatrix}\right]$. In particular, putting all pieces together, the following *Wiener-Hopf factorization* holds

$$\mathcal{S} = \begin{bmatrix} -Y_+ & I \\ I & X_+ \end{bmatrix} \begin{bmatrix} \left((I+QY_+)^{-1}A^*\right)^{-1} & 0 \\ 0 & (I+GX_+)^{-1}A \end{bmatrix} \begin{bmatrix} -Y_+ & I \\ I & X_+ \end{bmatrix}^{-1}. \tag{43}$$

This factorization relates explicitly the solutions X_+, Y_+ to a block diagonalization of $\mathcal{S}$.

An interesting limit case is the one when only the first part of Theorem 8 holds, (Q, A) is not d-detectable, and the solution X_+ exists but $\rho((I+GX_+)A) = 1$. In this case, $\mathcal{S}$ has eigenvalues on the unit circle, and it can be proved that all its Jordan blocks relative to these eigenvalues have even size: one can use a

result in Lancaster and Rodman[56, Theorem 12.2.3], after taking a factorization $G = BR^{-1}B^*$ with $R \succ 0$ and using another result in the same book [56, Theorem 12.2.1] to show that the hypothesis $\Psi(\eta) \succ 0$ holds.

It turns out that in this case the two iterations still converge, although (35) becomes sublinear and (41) becomes linear with rate 1/2. This is shown by Chiang, Chu, Guo, Huang, Lin and Xu [29]; the reader can recognize that the key step there is the study of the subspace iteration in presence of Jordan blocks of even multiplicity.

Note that the case in which the assumptions $Q \succeq 0, G \succeq 0$ do not hold is trickier, because there are examples where (28) does not have a stabilizing solution and $\mathcal{S}$ has Jordan blocks of odd size with eigenvalues on the unit circle: an explicit example is

$$A = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}, \qquad G = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad Q = \begin{bmatrix} 1 & 0 \\ 0 & -10 \end{bmatrix}, \tag{44}$$

which produces a matrix $\mathcal{S}$ with two simple eigenvalues (Jordan blocks of size 1) $\lambda_\pm \approx 0.598 \pm 0.801i$ with $|\lambda| = 1$. Surprisingly, eigenvalues on the unit circle are a generic phenomenon for symplectic matrices, which is preserved under perturbations: a small perturbation of the matrices in (44) will produce a perturbed $\tilde{\mathcal{S}}$ with two simple eigenvalues $\tilde{\lambda}_\pm$ that satisfy exactly $|\lambda| = 1$, because otherwise Part 2 of Lemma 9 would be violated.

6 Continuous-time Riccati equations

We consider the equation to be solved for $X = X^* \succeq 0$. This equation is known as *continuous-time algebraic Riccati equation* (CARE), and arises in various problems connected to continuous-time control theory [34, Chapter 10].

6.1 Solution properties

The similarities between (29) and (33) suggest that CAREs can be turned into DAREs (and *vice versa*) by converting the two associated invariant subspace problems; the ingredient to turn one into the other is the Cayley transform.

Lemma 11. *Let $A, G = G^*, Q = Q^*$ be given, and take $\tau > 0$. Set*

$$\begin{bmatrix} A_d & G_d \\ -Q_d & A_d^* \end{bmatrix} = \begin{bmatrix} A - \tau I & -G \\ Q & A^* - \tau I \end{bmatrix}^{-1} \begin{bmatrix} A + \tau I & -G \\ Q & A^* + \tau I \end{bmatrix} = I + 2\tau \begin{bmatrix} A - \tau I & -G \\ Q & A^* - \tau I \end{bmatrix}^{-1}. \tag{45}$$

Assume that the inverse exists, and that A_d is invertible. Then, the DARE with coefficients A_d, G_d, Q_d has the same solutions as the CARE with coefficients A, G, Q (and, in particular, the same maximal / stabilizing solution).

These formulas (45) follow from constructing $\mathcal{S} := c(\mathcal{H}) = (\mathcal{H} - \tau I)^{-1}(\mathcal{H} + \tau I)$, and then applying Lemma 10 to construct a factorization

$$\mathcal{S} = \begin{bmatrix} I & G_d \\ 0 & A_d^* \end{bmatrix}^{-1} \begin{bmatrix} A_d & 0 \\ -Q_d & I \end{bmatrix}.$$

The matrix $\mathcal{S}$ that we have constructed has the same invariant subspaces as $\mathcal{H}$ because $c(\cdot)$ is an invertible rational function: indeed, from (29), it follows that

$$\mathcal{S}\begin{bmatrix} I \\ X \end{bmatrix} = c(\mathcal{H})\begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X \end{bmatrix} c(M), \qquad M = A - GX.$$

This relation coincides with (33), and shows that a solution X of the CARE is also a solution of the DARE constructed with (45). Thanks to Lemma (4), M has all its eigenvalues in LHP if and only if $c(M)$ has all its eigenvalues inside the unit circle, so the stabilizing property of the solution is preserved.

Methods to transform DAREs into CAREs and vice versa based on the Cayley transform appear frequently in the literature starting from the 1960s; see for instance Mehrmann [62], a paper which explores these transformations and mentions the presence of many "folklore results" based on the Cayley transforms, relating the properties of the two associated equations.

Even if we restrict ourselves to the assumption that A_d is invertible when treating the DARE, it is important to remark that Lemma 11 does not generalize completely to the case when A_d is singular [62, Section 6]. By considering the poles of $c(\mathcal{H})$ as a function of τ, one sees that A_d is singular if and only if $\tau \in \Lambda(\mathcal{H})$. When this happens, even if $\mathcal{S}$ 'exists' in a suitable sense as an equivalent matrix pencil, an invariant subspace of $\mathcal{H}$ for which $\tau \in \Lambda(M)$ cannot be converted to the form (33), but only to the subtly weaker form

$$\begin{bmatrix} A_d & 0 \\ -Q_d & I \end{bmatrix}\begin{bmatrix} I \\ X \end{bmatrix}(M - \tau I) = \begin{bmatrix} I & G_d \\ 0 & A_d^* \end{bmatrix}\begin{bmatrix} I \\ X \end{bmatrix}(M + \tau I), \quad M = A - GX. \tag{46}$$

with an additional singular matrix $M - \tau I$ in the left-hand side. Thus we cannot write the equality (33), which identifies X as a solution of the DARE: hence the DARE has fewer solutions than the CARE. The stabilizing solution is always preserved by this transformation, though, because $\Lambda(M) \subset$ LHP cannot contain $\tau > 0$.

6.2 Algorithms

In view of the relation between DAREs and CAREs that we have just outlined, a natural algorithm is using the formulas (45) to convert (27) into an equivalent (28) and solving it using (41). This algorithm has been suggested by Chu, Fan and Lin [30] as a doubling algorithm for CAREs. This algorithm inherits all the nice convergence properties of SDA for DAREs; in particular, among them, the fact that it also works (at reduced linear speed) on problems in which $A - GX_+$ has eigenvalues on the imaginary axis [29].

While SDA works well in general, a delicate point is the choice of the shift value τ. In principle almost every choice of τ works, since $\mathcal{H} - \tau I$ is singular only for at most $2n$ values of τ, but in practice choosing the wrong value of τ may affect accuracy negatively. Dangers arise not from singularity of $\mathcal{H} - \tau I$ (which is actually harmless with a matrix pencil formulation), but from singularity

in (45), and also from taking τ too large or too small by orders of magnitude. A heuristic approach based on golden section search has been suggested [30].

In practice, one would prefer to avoid the Cayley transform or at least delay it as much as possible; this observation leads to another popular algorithm for CAREs. We start from the following observation.

Lemma 12. *If $\mathcal{S} = c(\mathcal{H})$ (with a parameter $\tau \in \mathbb{R}$), then*

$$\mathcal{S}^2 = c\left(\frac{1}{2}\left(\mathcal{H} + \tau^2\mathcal{H}^{-1}\right)\right). \tag{47}$$

This identity can be verified directly, using the fact that rational functions of the same matrix $\mathcal{H}$ all commute with each other.

Applying this identity repeatedly, we get $\mathcal{S}^{2^k} = c(\mathcal{H}_k)$, where

$$\mathcal{H}_{k+1} = \frac{1}{2}\left(\mathcal{H}_k + \tau^2\mathcal{H}_k^{-1}\right), \qquad \mathcal{H}_0 = \mathcal{H}. \tag{48}$$

Hence one can hold off the Cayley transform and just compute the sequence $\mathcal{H}_k$ directly, starting from (30). This constructs a sequence which represents implicitly $\mathcal{S}^{2^k}$.

Constructing the matrices $\mathcal{H}_k$ is numerically much less troublesome than constructing explicitly $\mathcal{S}^{2^k}$ or its inverse $\mathcal{S}^{-2^k}$. Indeed, it is instructive to consider the behaviour of these iterations in a basis in which $\mathcal{H}$ is diagonal (when it exists). Let λ be a generic diagonal entry (i.e., an eigenvalue) of $\mathcal{H}$. Then, $\mathcal{S} = c(\mathcal{H})$ has the corresponding eigenvalue $c(\lambda)$, and $\mathcal{S}^{2^k}$ has the eigenvalue $c(\lambda)^{2^k}$. If $\lambda \in$ LHP, then $|c(\lambda)| < 1$ (Lemma 4), and hence $c(\lambda)^{2^k} \to 0$ when $k \to \infty$. Similarly, if λ is in the right half-plane, then $|c(\lambda)| > 1$ and $c(\lambda)^{2^k} \to \infty$. Thus $\mathcal{S}^{2^k}$ (as well as its inverse) has some eigenvalues that converge to zero, and some that diverge to infinity, as k grows. This is one of the reasons why it is preferable to keep $\mathcal{S}$ in its factored form (38). On the other hand, the eigenvalues of $\mathcal{H}_k$ converge to finite values $c^{-1}(0) = -\tau$ and $c^{-1}(\infty) = \tau$, so this computation suggests that the direct computation of $\mathcal{H}_k$ is feasible.

The *sign function method*[72, 35, 40] to solve CAREs consists exactly in computing the iteration (48) up to convergence, obtaining a matrix $\mathcal{H}_\infty = \lim_{k\to\infty} \mathcal{H}_k$ that has numerically n eigenvalues equal to $\tau \in$ RHP and n equal to $-\tau \in$ LHP, and then computing

$$\operatorname{Im}\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \ker(\mathcal{H}_\infty + \tau I), \qquad U_1, U_2 \in \mathbb{C}^{n\times n}, \qquad X_+ = U_2 U_1^{-1}. \tag{49}$$

The method takes its name from the fact that the limit matrix $\mathcal{H}_\infty$ (for $\tau = 1$) is the so-called *matrix sign function* of $\mathcal{H}$. We refer the reader to its analysis in Higham [48, Chapter 5], in which one clearly sees that one of the main ingredients is the formula 47 relating the iteration to repeated squaring.

Scaling is an important detail that deserves a discussion. Replacing $\mathcal{H}$ with a positive multiple of itself corresponds to multiplying each term of (27) by a

positive quantity; this operation does not change the solutions of the equation, nor the maximal / stabilizing properties of X_+. In SDA, scaling is limited to choosing the parameter of the initial Cayley transform, but in the sign method we have more freedom: we can take a different τ_k at each step of (48). We remark that scaling for the sign method is usually presented in the literature in a slightly different form: one replaces (48) with

$$\mathcal{H}_{k+1} = \frac{1}{2}\left((\tau_k^{-1}\mathcal{H}_k) + (\tau_k^{-1}\mathcal{H}_k)^{-1}\right). \tag{50}$$

The two forms are essentially equivalent, as they return iterates $\mathcal{H}_k$ that differ only by a multiplicative factor, which is then irrelevant in the final step (49). Irrespective of formulation, the main result is that a judicious choice of scaling can speed up the convergence of (48) or (50). A cheap and effective choice of scaling, *determinantal scaling*, $\tau_k = (\det \mathcal{H}_k)^{\frac{1}{n}}$ has been suggested by Byers [26]. Other related choices of scaling and their performances have been discussed by Higham [48, Chapter 5] and Kenney and Laub [52]. The general message is that scaling has a great impact in the first steps of the iteration, when it can greatly improve convergence, but once the residual starts to decrease its effect in the later steps becomes negligible.

Scaling also has an impact on stability; the stability of the sign iteration as a method to compute invariant subspaces (and hence ultimately Riccati solutions) has been studied by Bai and Demmel [3] and Byers, He and Mehrmann [27]. The two interesting messages are that (expectedly) the sign function method suffers when $\mathcal{H}$ is ill-conditioned, but that (unexpectedly) the invariant subspaces extracted from $\mathcal{H}_\infty$ has better stability properties than $\mathcal{H}_\infty$ itself. A version of the sign iteration that uses matrix pencils to reduce the impact of these inversions have been suggested by Benner and Byers [11].

Another useful computational detail is that one can rewrite the sign function method (48) as

$$\mathcal{M}_{k+1} = \frac{1}{2}(\mathcal{M}_k + \tau^2 J\mathcal{M}_k^{-1}J), \qquad \mathcal{M}_k = \mathcal{H}_k J,$$

which is cheaper because one can take advantage of the fact that the matrices $\mathcal{M}_k$ are Hermitian [26]. Indeed, it is a general observation that most of the matrix algebra operations needed in doubling-type algorithms can be reduced to operations on symmetric/Hermitian matrices; see for instance also (45).

6.3 Remarks

The formulation in the sign iteration allows one to introduce some form of per-iteration scaling in the setting of a doubling-type algorithm. It would be interesting to see if this scaling can be transferred to the SDA setting, and which computational advantage it brings. Note that, in view of (47), scaling the sign iteration is equivalent to changing the parameter τ in the Cayley transform. So SDA does incorporate a form of scaling, but only at the first iteration, when one chooses τ.

In general, it is unclear if scaling after the first iteration produces major gains in convergence speed. It would be appealing to try and study this kind of scaling with the tools of polynomial and rational approximation, like it has been done in more details for non-doubling algorithms, with the aim of deriving optimal choices for the parameters τ and σ_k.

There is another classical iterative algorithm to solve algebraic Riccati equations (both in discrete and continuous time), and it is Newton's method. For the simpler case of CAREs, Newton's method [53] consists in determining X_{k+1} by solving at each step the Lyapunov equation

$$(A-GX_k)^*(X_{k+1}-X_k)+(X_{k+1}-X_k)(A-GX_k) = -(Q+A^*X_k+X_kA-X_kGX_k) \tag{51}$$

or the equivalent one

$$(A - GX_k)^*X_{k+1} + X_{k+1}(A - GX_k) = -Q - X_kGX_k.$$

A line search procedure, which improves convergence speed in practice, has been introduced by Benner and Byers [10]. The method can be used, in particular, for large and sparse equations in conjunction with low-rank ADI [13].

The reader may wonder if there is an explicit relation between doubling algorithms and Newton-type algorithms, considering especially that both exhibit quadratic convergence (which, moreover, in both cases degrades to linear with rate $1/2$ if $A-GX_+$ has purely imaginary eigenvalues [44]). The answer, unfortunately, seems to be no. An argument that suggests that the two iterations are genuinely different is that the iterates produced by Newton's method approach X_+ from *above* [53] (i.e., $X_1 \succeq X_2 \succeq \cdots \succeq X_k \succeq X_{k+1} \succeq \cdots \succeq X_+$), not from *below* like the iterates Q_k of SDA in (41c).

Some more recent algorithms for large and sparse CAREs essentially merge the Newton step (51) and the ADI iteration (25) into a single iteration [59, 77, 9]. It is again unclear whether there is an explicit relation between these two families of methods.

An interesting question is what is the 'non-doubling' analogue of the sign method and of SDA. One can convert the CARE to discrete-time using (45) and formulate (35), but to our knowledge this method does not have a more appealing presentation in terms of a simple iterative method for (27), like it has in all the other discrete-time examples.

Another 'philosophical' observation is that the sign function method does not avoid a Cayley-type transformation; it merely pushes it back to the very last step (49), where the sub-expression $\mathcal{H}+\tau I$ appears; this operation takes the role of a discretizing transformation that maps the eigenvalue $-\tau$ into a value inside a given circle and the eigenvalue τ into one outside. A discretizing transformation of some sort seems inevitable in this family of algorithms, although delaying it until the very last step seems beneficial for accuracy, because at that point we have complete control of the location of eigenvalues.

7 Unilateral equations and NMEs

We end our discussion of the family of Riccati-type equations with a pair of oft-neglected cousins, and present them with an application that shows clearly the relationship between them. Consider the matrix Laurent polynomial

$$P(z) = Az^{-1} + Q + A^* z, \qquad Q = Q^* \succ 0, \qquad A, Q \in \mathbb{C}^{n\times n}. \tag{52}$$

The problem of *spectral factorization* (of quadratic matrix polynomials) consists in determining a factorization

$$P(z) = (zY^* - I)X(z^{-1}Y - I), \qquad X = X^* \succ 0, \qquad X, Y \in \mathbb{C}^{n\times n}, \tag{53}$$

such that $\rho(Y) \leq 1$. In particular, the left factor is invertible for $|z| < 1$, and the right factor is invertible for $|z| > 1$.

Equating coefficients in (52) and (53) gives $-XY = A$, $Q = X + Y^*XY$. We can eliminate one among X and Y from this system of two equations, getting two equations with a single unknown each

$$0 = A + QY + A^*Y^2, \tag{54}$$

$$Q = X + A^*X^{-1}A. \tag{55}$$

The first one (54) is called *unilateral quadratic matrix equation* [18], while the second one (55) is known with the (rather undescriptive) name of *nonlinear matrix equation* (NME) [46, 45, 49].

While (54) looks more appealing at first, as it reveals direct ties with the palindromic quadratic eigenvalue problem [46, 45, 60], it is in fact (55) that reveals more structure: for instance, (55) has Hermitian solutions (see below), while the structure in the solutions of (54) is much less apparent.

7.1 Solution properties

It follows from (53) that $P(\lambda) \succeq 0$ for each λ that belongs to the unit circle (hence $\lambda^{-1} = \bar{\lambda}$), so this is a necessary condition for the solvability of this problem. It can be proved that it is sufficient, too, and that a maximal / stabilizing solution exists.

Theorem 13. *[39, Theorem 2.2] Assume that $P(z)$ is regular and $P(\lambda) \succeq 0$ for each λ on the unit circle. Then,* (55) *has a (unique) solution X_+ such that*

(1) $X_+ = X_+^* \succ 0$;

(2) $X_+ \succeq X$ for any other Hermitian solution X;

(3) $\rho(Y) = \rho(-X_+^{-1}A) \leq 1$

If, in addition, $P(\lambda) \succ 0$ for each λ on the unit circle, then $\rho(-X_+^{-1}A) < 1$.

Once again, we can rewrite (55) as an invariant subspace problem.

$$\begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} 0 & -I \\ A^* & 0 \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} Y, \qquad Y = -X^{-1}A. \tag{56}$$

We assume again that A is invertible to avoid technicalities with matrix pencils. The matrix

$$\mathcal{S} = \begin{bmatrix} 0 & -I \\ A^* & 0 \end{bmatrix}^{-1} \begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} \tag{57}$$

is symplectic, and so is the slightly more general form

$$\begin{bmatrix} G & -I \\ A^* & 0 \end{bmatrix}^{-1} \begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix}. \tag{58}$$

Lemma 14. *(1) A matrix in the form* (58) *is symplectic if and only if $G = G^*, Q = Q^*$, and the two blocks called A, A^* in* (34) *are one the conjugate transpose of the other.*

(2) If λ is an eigenvalue of a symplectic matrix with right eigenvector v, then $\overline{\lambda}^{-1}$ is an eigenvalue of the same matrix with left eigenvector v^*J.

(3) If the hypotheses of Theorem 13 hold (including the strict positivity one in the end), then the $2n$ eigenvalues of $\mathcal{S}$ are (counting multiplicities) the n eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of $-X_+^{-1}A$ inside the unit circle, and the n eigenvalues $\overline{\lambda_i}^{-1}$, $i = 1, 2, \ldots, n$ outside the unit circle.

The symplectic structure behind this equation is the same one as the DARE, and indeed Part 2 of this lemma is identical to Part 2 of Lemma 9. Indeed, Engwerda, Ran and Rijkeboer [39, Section 7] note that (55) can be reduced to a DARE, although it is one that does not fall inside our framework since it has $G \preceq 0$.

7.2 Algorithms

The formulation (55) suggests immediately the iterative algorithm

$$X_{k+1} = Q - A^* X_k^{-1} A. \tag{59}$$

Clearly we cannot start this iteration from 0, so we take $X_1 = Q$ instead. An interesting interpretation of this algorithm is as iterated Schur complements of block Toeplitz tridiagonal matrices. The Schur complement of the $(1,1)$ block of the tridiagonal matrix

$$\underbrace{\begin{bmatrix} X_k & A^* & & & \\ A & Q & A^* & & \\ & A & Q & \ddots & \\ & & \ddots & \ddots & A^* \\ & & & A & Q \end{bmatrix}}_{h \text{ blocks}},$$

is

$$\underbrace{\begin{bmatrix} X_{k+1} & A^* & & & \\ A & Q & A^* & & \\ & A & Q & \ddots & \\ & & \ddots & \ddots & A^* \\ & & & A & Q \end{bmatrix}}_{h-1 \text{ blocks}}.$$

Hence the whole iteration can be interpreted as constructing successive Schur complements of the tridiagonal matrix

$$\mathcal{Q}_m := \underbrace{\begin{bmatrix} Q & A^* & & & \\ A & Q & A^* & & \\ & A & Q & \ddots & \\ & & \ddots & \ddots & A^* \\ & & & A & Q \end{bmatrix}}_{m \text{ blocks}}. \tag{60}$$

It can be seen that $\mathcal{Q}_m$ is positive semidefinite, under the assumptions of Theorem 13: a quick sketch of a proof is as follows. The matrix $\mathcal{Q}_m$ is a submatrix of

$$\begin{bmatrix} Q & A^* & & & A \\ A & Q & A^* & & \\ & A & Q & \ddots & \\ & & \ddots & \ddots & A^* \\ A^* & & & A & Q \end{bmatrix} = (\Phi \otimes I) \begin{bmatrix} P(1) & & & & \\ & P(\zeta) & & & \\ & & P(\zeta^2) & & \\ & & & \ddots & \\ & & & & P(\zeta^{-1}) \end{bmatrix} (\Phi \otimes I)^{-1},$$

which the equation shows to be similar (using the Fourier matrix Φ and properties of Fourier transforms) to a block diagonal matrix that contains $P(z)$ from (52) evaluated in the roots of unity $1, \zeta, \zeta^2, \ldots, \zeta^{-1}$.

Hence, in particular, all the X_k are positive semidefinite. One can further show that $Q = X_0 \succeq X_1 \succeq X_2 \succeq \cdots \succeq X_k \succeq \ldots$. The sequence X_k is monotonic and bounded from below, hence it converges, and one can show that its limit is X_+ [39, Section 4] (to do this, verify the property in Point (2) of Theorem 13 by proving that $X_k \succeq X$ at each step of the iteration).

A doubling variant of (59) can be constructed starting from this Schur complement interpretation. The Schur complement of the submatrix formed by the

odd-numbered blocks $(1, 3, 5, \ldots, 2m-1)$ of

$$\underbrace{\begin{bmatrix} U_k & A_k^* & & & \\ A_k & U_k & A_k^* & & \\ & A_k & \ddots & \ddots & \\ & & \ddots & U_k & A_k^* \\ & & & A_k & Q_k \end{bmatrix}}_{2m \text{ blocks}},$$

is

$$\underbrace{\begin{bmatrix} U_{k+1} & A_{k+1}^* & & & \\ A_{k+1} & U_{k+1} & A_{k+1}^* & & \\ & A_{k+1} & \ddots & \ddots & \\ & & \ddots & U_{k+1} & A_{k+1}^* \\ & & & A_{k+1} & Q_{k+1} \end{bmatrix}}_{m \text{ blocks}},$$

with

$$A_{k+1} = -A_k U_k^{-1} A_k, \tag{61a}$$
$$Q_{k+1} = Q_k - A_k^* U_k^{-1} A_k, \tag{61b}$$
$$U_{k+1} = U_k - A_k^* U_k^{-1} A_k - A_k U_k^{-1} A_k^*. \tag{61c}$$

We can construct the Schur complement of the first $2^k - 1$ blocks of $\mathcal{Q}_{2^k}$ in two different ways: either we make $2^k - 1$ iterations of (59), resulting in X_{2^k}, or we make k iterations of (61), starting from $A_0 = A, Q_0 = U_0 = Q$, resulting in Q_k. This shows that $Q_k = X_{2^k}$.

This peculiar way to take Schur complements of Toeplitz tridiagonal matrices was introduced by Buzbee, Golub and Nielson [25] to solve certain differential equations, and then later applied to matrix equations similar to (54) and (55) by Bini, Gemignani, and Meini [16, 17, 65]. The iteration (61) is known as *cyclic reduction*.

One can derive the same iteration from repeated squaring, in the same way as we obtained SDA as a modified subspace iteration [58]. We seek formulas to update a factorization of the kind

$$\mathcal{S}^{-2^k} = \begin{bmatrix} A_k & 0 \\ -Q_k & I \end{bmatrix}^{-1} \begin{bmatrix} G_k & -I \\ A_k^* & 0 \end{bmatrix}.$$

To do this, we write (analogously to (40))

$$\mathcal{S}^{-2^{k+1}} = \mathcal{S}^{-2^k} \mathcal{S}^{-2^k} = \begin{bmatrix} A_k & 0 \\ -Q_k & I \end{bmatrix}^{-1} \left(\begin{bmatrix} G_k & -I \\ A_k^* & 0 \end{bmatrix} \begin{bmatrix} A_k & 0 \\ -Q_k & I \end{bmatrix}^{-1} \right) \begin{bmatrix} G_k & -I \\ A_k^* & 0 \end{bmatrix}$$

and use Lemma 10 (with $[M_1\ M_2] = I_{2n}$) to find a factorization in the form (39) of the term in parentheses, which then combines with the outer terms to produce the sought decomposition. The resulting formulas are

$$A_{k+1} = -A_k(Q_k - G_k)^{-1}A_k, \tag{62a}$$
$$Q_{k+1} = Q_k - A_k^*(Q_k - G_k)^{-1}A_k, \tag{62b}$$
$$G_{k+1} = G_k + A_k(Q_k - G_k)^{-1}A_k^*, \tag{62c}$$

and one sees that they coincide with (61), after setting $U_k = Q_k - G_k$. With an argument analogous to the one in Section 5, one sees that

$$\mathcal{S}^{-2^k}\begin{bmatrix} 0 \\ -I \end{bmatrix} = \begin{bmatrix} I \\ Q_k \end{bmatrix},$$

thus $\left[\begin{smallmatrix} I \\ Q_k \end{smallmatrix}\right]$ converges to a basis of the invariant subspace associated to the eigenvalues of $\mathcal{S}$ inside the unit circle.

This formulation (62) is known as *SDA-II* [58, 32].

7.3 Remarks

Even though we have mentioned spectral factorization only here, it can be formulated for more complicated matrix functions also in the context of DAREs and CAREs; in fact, it is a classical topic, and another facet of the multiple connections between matrix equations and control theory [4, 5, 76].

The interpretation as Schur complement is a powerful trick, which reveals a greater picture in this family of methods. It may possibly be used to understand more about the stability of these methods, since Schur complementation and Gaussian elimination on symmetric positive definite matrices is a well understood topic from the numerical point of view.

Many authors have studied variants of (55). Typically, one replaces the nonlinear term with various functions of the form $A^*f(X)A$, or adds more nonlinear terms. In the modified versions, it is often possible to prove convergence of the fixed-point algorithm with arguments of monotonicity, and prove the existence of a solution under some assumptions. However, after any nontrivial modification the connection with invariant subspaces is lost. This fact, coupled with lack of applications, makes these variants much less interesting than the original equation, in the eyes of the author.

8 Nonsymmetric variants in applied probability

Many of the equations treated here have nonsymmetric variants which appear naturally in queuing theory, a sub-field of applied probability. In the analysis of *quasi-birth-death models* [57, 20], one encounters equations of the form

$$0 = A + QY + BY^2, \qquad A, B, Q, Y \in \mathbb{R}^{n\times n}, \tag{63}$$

where $A, B \geq 0$ (we use the notation $M \geq N$ to denote that a matrix M is entrywise larger than N, i.e., $M_{ij} \geq N_{ij}$ for all i, j), and the matrix $-Q$ is an M-matrix, i.e., $Q_{ij} \geq 0$ for $i \neq j$ and $\Lambda(Q) \subset \overline{\text{LHP}}$. These equations have a solution $Y \geq 0$ which has a natural probabilistic interpretation. The solution X to $X = Q - BX^{-1}A$ and the solution of the associated dual equation $0 = Z^2A + ZQ + B$ also appear naturally and have a related probabilistic meaning [57, Chapter 6][20, Section 5.6].

Similarly, the equation

$$Q + BX + XA - XGX = 0, \quad Q, X \in \mathbb{R}^{m\times n}, A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{m\times m}, G \in \mathbb{R}^{n\times m}. \tag{64}$$

appears in the study of so-called *fluid queues*, or *stochastic flow models* [73, 51, 33]. The matrices A, B are M-matrices, while $G, -Q \geq 0$. One can formulate nonsymmetric analogues of basic matrix iterations and doubling algorithms. Unfortunately, the theory does not translate perfectly to this setting, due to the sign differences between the two cases: in the symmetric equations $G, Q \succeq 0$, while in the nonsymmetric case $G, -Q \geq 0$. Due to this asymmetry, the signs in the two cases do not match, and one needs to formulate different arguments. For instance, in the symmetric case one proves that the inverses that appear in (41) exist because $G_k \succeq 0$, $Q_k \succeq 0$; while in its nonsymmetric analogue $G_k, -Q_k \geq 0$, and one proves that $I + G_kQ_k$ and $I + Q_kG_k$ are M-matrices to show that those inverses exist.

The equation (28) does not appear to have an immediate analogue in queuing theory, but this fact seems just an accident, since some of the results that involve (64) could have been formulated with an equivalent equation resembling more (28) than (27) instead. There is a distinction between discrete-time and continuous-time models also in applied probability, but in many cases it does not affect directly the shape of the equations; for instance (63) takes the same form for discrete- and continuous-time QBDs. The role of discretizing transformations such as Cayley transforms in this context has been studied by Bini, Meini, and Poloni [21].

For reasons of space, we cannot give here a complete treatment of these nonsymmetric variants. Huang, Li and Lin [49] in their book enter into more detail about the doubling algorithms for these equations, but a great part of the theory (including existence results and probabilistic interpretations for the iterates of various numerical methods) is unfortunately available only in the queuing theory literature, strictly entangled with its applications.

An interesting remark is that the M-matrix structure allows one to construct stability proofs more easily. Conditioning and stability results for these equations have been studied by some authors [86, 85, 66, 84, 28], relying heavily on the sign and M-matrix structure. The forward stability proof in Nguyen and Poloni [66] is, to date, one of the very few complete stability proofs for a doubling-type algorithm.

9 Conclusions

In this paper, we presented from a consistent point of view doubling algorithms for symmetric Riccati-type equations, relating them to the basic iterations of which they are a 'squaring' variant. We have included various algorithms that belong to the same family but have appeared independently, such as the sign iteration and cyclic reduction. We have outlined relations between doubling algorithms, the subspace iteration, ADI-type and Krylov subspace methods, and Schur complementation of tridiagonal block Toeplitz matrices. This theory, in turn, forms only a small portion of the far larger topic of numerical algorithms for Riccati-type equations and control theory. This field of research is an incredibly vast one, spanning at least six decades of literature and various communities between engineering and mathematics, so we have surely omitted or forgotten many relevant contributions; we apologize with the missing authors.

We hope that the reader can benefit from our paper by both gaining theoretical insight, and having available some numerical algorithms for these equations. Indeed, with respect to many competitors, doubling-based algorithms have the advantage that they reduce to the simple coupled matrix iterations (41) or (61), which are easy to code and fast to run in many computational environments.

Another interesting remark that was suggested by a referee is that some recent lines of research consider this family of matrix equations under different types of data sparsity than low-rank: for instance, Palitta and Simoncini [67] consider banded data, and Kressner, Massei and Robol [54] and Massei, Palitta and Robol [61] consider semi-separable (low-rank off-diagonal blocks) and hierarchically semiseparable structures. Much earlier, Grasedyck, Hackbusch and Khoromskij [42] considered using hierarchical matrices to solve Riccati equations. All these structures are (at least up to a degree) preserved by the operations involved in doubling methods [23, 83]. These novel techniques may open up new lines of research for doubling-type algorithms.

References

[1] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank. *Matrix Riccati equations*. Systems & Control: Foundations & Applications. Birkhäuser Verlag, Basel, 2003. In control and systems theory. doi:10.1007/978-3-0348-8081-7.

[2] B. D. O. Anderson. Second-order convergent algorithms for the steady-state Riccati equation. *Internat. J. Control*, 28(2):295–306, 1978. doi:10.1080/00207177808922455.

[3] Z. Bai and J. Demmel. Using the matrix sign function to compute invariant subspaces. *SIAM J. Matrix Anal. Appl.*, 19(1):205–225, 1998. doi:10.1137/S0895479896297719.

[4] H. Bart, I. Gohberg, M. A. Kaashoek, and A. C. M. Ran. *Factorization of matrix and operator functions: the state space method*, volume 178 of

Operator Theory: Advances and Applications. Birkhäuser Verlag, Basel, 2008. Linear Operators and Linear Systems.

[5] H. Bart, I. Gohberg, M. A. Kaashoek, and A. C. M. Ran. *A state space approach to canonical factorization with applications*, volume 200 of *Operator Theory: Advances and Applications*. Birkhäuser Verlag, Basel; Birkhäuser Verlag, Basel, 2010. Linear Operators and Linear Systems. doi:10.1007/978-3-7643-8753-2.

[6] R. H. Bartels and G. W. Stewart. Algorithm 432: solution of the matrix equation $AX + XB = C$. *Comm. ACM*, 15:820–826, 1972.

[7] B. Beckermann and A. Townsend. Bounds on the singular values of matrices with displacement structure. *SIAM Rev.*, 61(2):319–344, 2019. Revised reprint of "On the singular values of matrices with displacement structure" [MR3717820]. doi:10.1137/19M1244433.

[8] P. Benner, T. Breiten, and T. Damm. Generalised tangential interpolation for model reduction of discrete-time MIMO bilinear systems. *Internat. J. Control*, 84(8):1398–1407, 2011. doi:10.1080/00207179.2011.601761.

[9] P. Benner, Z. Bujanović, P. Kürschner, and J. Saak. RADI: a low-rank ADI-type algorithm for large scale algebraic Riccati equations. *Numer. Math.*, 138(2):301–330, 2018. doi:10.1007/s00211-017-0907-5.

[10] P. Benner and R. Byers. An exact line search method for solving generalized continuous-time algebraic Riccati equations. *IEEE Trans. Automat. Control*, 43(1):101–107, 1998. doi:10.1109/9.654908.

[11] P. Benner and R. Byers. An arithmetic for matrix pencils: theory and new algorithms. *Numer. Math.*, 103(4):539–573, 2006. doi:10.1007/s00211-006-0001-x.

[12] P. Benner, G. El Khoury, and M. Sadkane. On the squared Smith method for large-scale Stein equations. *Numer. Linear Algebra Appl.*, 21(5):645–665, 2014. doi:10.1002/nla.1918.

[13] P. Benner, J.-R. Li, and T. Penzl. Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. *Numer. Linear Algebra Appl.*, 15(9):755–777, 2008. doi:10.1002/nla.622.

[14] P. Benner, R.-C. Li, and N. Truhar. On the ADI method for Sylvester equations. *J. Comput. Appl. Math.*, 233(4):1035–1045, 2009. doi:10.1016/j.cam.2009.08.108.

[15] P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí. Numerical solution of discrete stable linear matrix equations on multicomputers. *Parallel Algorithms Appl.*, 17(2):127–146, 2002. doi:10.1080/10637190208941436.

[16] D. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM J. Matrix Anal. Appl.*, 17(4):906–926, 1996. doi:10.1137/S0895479895284804.

[17] D. A. Bini, L. Gemignani, and B. Meini. Computations with infinite Toeplitz matrices and polynomials. *Linear Algebra Appl.*, 343/344:21–61, 2002. Special issue on structured and infinite systems of linear equations. doi:10.1016/S0024-3795(01)00341-X.

[18] D. A. Bini, B. Iannazzo, G. Latouche, and B. Meini. On the solution of algebraic Riccati equations arising in fluid queues. *Linear Algebra Appl.*, 413(2-3):474–494, 2006. doi:10.1016/j.laa.2005.04.019.

[19] D. A. Bini, B. Iannazzo, and B. Meini. *Numerical solution of algebraic Riccati equations*, volume 9 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2012.

[20] D. A. Bini, G. Latouche, and B. Meini. *Numerical methods for structured Markov chains*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2005. Oxford Science Publications. doi:10.1093/acprof:oso/9780198527688.001.0001.

[21] D. A. Bini, B. Meini, and F. Poloni. Transforming algebraic Riccati equations into unilateral quadratic matrix equations. *Numer. Math.*, 116(4):553–578, 2010. doi:10.1007/s00211-010-0319-2.

[22] S. Bittanti, A. Laub, and J. Willems. *The Riccati equation*. Communications and Control Engineering. Springer-Verlag, Berlin, 1991.

[23] S. Börm, L. Grasedyck, and W. Hackbusch. *Hierarchical matrices*. Max-Planck-Institut für Mathematik inden Naturwissenschaften, Leipzig, Germany, 2003. Lecture note 21.

[24] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994. doi:10.1137/1.9781611970777.

[25] B. L. Buzbee, G. H. Golub, and C. W. Nielson. On direct methods for solving Poisson's equations. *SIAM J. Numer. Anal.*, 7:627–656, 1970. doi:10.1137/0707049.

[26] R. Byers. Solving the algebraic Riccati equation with the matrix sign function. *Linear Algebra Appl.*, 85:267–279, 1987. doi:10.1016/0024-3795(87)90222-9.

[27] R. Byers, C. He, and V. Mehrmann. The matrix sign function method and the computation of invariant subspaces. *SIAM J. Matrix Anal. Appl.*, 18(3):615–632, 1997. doi:10.1137/S0895479894277454.

[28] C. Chen, R.-C. Li, and C. Ma. Highly accurate doubling algorithm for quadratic matrix equation from quasi-birth-and-death process. *Linear Algebra Appl.*, 583:1–45, 2019. doi:10.1016/j.laa.2019.08.018.

[29] C.-Y. Chiang, E. K.-W. Chu, C.-H. Guo, T.-M. Huang, W.-W. Lin, and S.-F. Xu. Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case. *SIAM J. Matrix Anal. Appl.*, 31(2):227–247, 2009. doi:10.1137/080717304.

[30] E. K.-W. Chu, H.-Y. Fan, and W.-W. Lin. A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations. *Linear Algebra Appl.*, 396:55–80, 2005. doi:10.1016/j.laa.2004.10.010.

[31] E. K.-W. Chu, H.-Y. Fan, W.-W. Lin, and C.-S. Wang. Structure-preserving algorithms for periodic discrete-time algebraic Riccati equations. *Internat. J. Control*, 77(8):767–788, 2004. doi:10.1080/00207170410001714988.

[32] E. K.-W. Chu, T.-M. Hwang, W.-W. Lin, and C.-T. Wu. Vibration of fast trains, palindromic eigenvalue problems and structure-preserving doubling algorithms. *J. Comput. Appl. Math.*, 219(1):237–252, 2008. doi:10.1016/j.cam.2007.07.016.

[33] A. da Silva Soares. *Fluid queues – Building upon the analogy with QBD processes*. PhD thesis, 2005.

[34] B. N. Datta. *Numerical methods for linear control systems*. Elsevier Academic Press, San Diego, CA, 2004. Design and analysis.

[35] E. D. Denman and A. N. Beavers, Jr. The matrix sign function and computations in systems. *Appl. Math. Comput.*, 2(1):63–94, 1976. doi:10.1016/0096-3003(76)90020-5.

[36] V. Druskin, L. Knizhnerman, and V. Simoncini. Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation. *SIAM J. Numer. Anal.*, 49(5):1875–1898, 2011. doi:10.1137/100813257.

[37] V. Druskin and V. Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Systems Control Lett.*, 60(8):546–560, 2011. doi:10.1016/j.sysconle.2011.04.013.

[38] N. S. Ellner and E. L. Wachspress. Alternating direction implicit iteration for systems with complex spectra. *SIAM J. Numer. Anal.*, 28(3):859–870, 1991. doi:10.1137/0728045.

[39] J. C. Engwerda, A. C. M. Ran, and A. L. Rijkeboer. Necessary and sufficient conditions for the existence of a positive definite solution of the matrix equation $X + A^*X^{-1}A = Q$. *Linear Algebra Appl.*, 186:255–275, 1993. doi:10.1016/0024-3795(93)90295-Y.

[40] J. D. Gardiner and A. J. Laub. A generalization of the matrix-sign-function solution for algebraic Riccati equations. *International Journal of Control*, 44(3):823–832, 1986. doi:10.1080/00207178608933634.

[41] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.

[42] L. Grasedyck, W. Hackbusch, and B. N. Khoromskij. Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices. *Computing*, 70(2):121–165, 2003. doi:10.1007/s00607-002-1470-0.

[43] S. Gugercin and A. C. Antoulas. A survey of model reduction by balanced truncation and some new results. *Internat. J. Control*, 77(8):748–766, 2004. doi:10.1080/00207170410001713448.

[44] C.-H. Guo and P. Lancaster. Analysis and modification of Newton's method for algebraic Riccati equations. *Math. Comp.*, 67(223):1089–1105, 1998. doi:10.1090/S0025-5718-98-00947-8.

[45] C.-H. Guo and W.-W. Lin. The matrix equation $X + A^T X^{-1} A = Q$ and its application in nano research. *SIAM J. Sci. Comput.*, 32(5):3020–3038, 2010. doi:10.1137/090758209.

[46] C.-H. Guo and W.-W. Lin. Solving a structured quadratic eigenvalue problem by a structure-preserving doubling algorithm. *SIAM J. Matrix Anal. Appl.*, 31(5):2784–2801, 2010. doi:10.1137/090763196.

[47] S. Güttel. Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. *GAMM-Mitt.*, 36(1):8–31, 2013. doi:10.1002/gamm.201310002.

[48] N. J. Higham. *Functions of matrices*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008. Theory and computation. doi:10.1137/1.9780898717778.

[49] T.-M. Huang, R.-C. Li, and W.-W. Lin. *Structure-preserving doubling algorithms for nonlinear matrix equations*, volume 14 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2018. doi:10.1137/1.9781611975369.ch1.

[50] V. Ionescu, C. Oară, and M. Weiss. *Generalized Riccati theory and robust control*. John Wiley & Sons, Ltd., Chichester, 1999. A Popov function approach.

[51] R. L. Karandikar and V. Kulkarni. Second-order fluid flow models: Reflected Brownian motion in a random environment. *Oper. Res*, 43:77–88, 1995.

[52] C. Kenney and A. J. Laub. On scaling Newton's method for polar decomposition and the matrix sign function. *SIAM J. Matrix Anal. Appl.*, 13(3):698–706, 1992. doi:10.1137/0613044.

[53] D. Kleinman. On an iterative technique for Riccati equation computations. *IEEE Transactions on Automatic Control*, 13(1):114–115, February 1968. doi:10.1109/TAC.1968.1098829.

[54] D. Kressner, S. Massei, and L. Robol. Low-rank updates and a divide-and-conquer method for linear matrix equations. *SIAM J. Sci. Comput.*, 41(2):A848–A876, 2019. doi:10.1137/17M1161038.

[55] Y.-C. Kuo, W.-W. Lin, and S.-F. Shieh. Structure-preserving flows of symplectic matrix pairs. *SIAM J. Matrix Anal. Appl.*, 37(3):976–1001, 2016. doi:10.1137/15M1019155.

[56] P. Lancaster and L. Rodman. *Algebraic Riccati equations*. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, 1995.

[57] G. Latouche and V. Ramaswami. *Introduction to matrix analytic methods in stochastic modeling*. ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; American Statistical Association, Alexandria, VA, 1999. doi:10.1137/1.9780898719734.

[58] 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(1):26–39, 2006. doi:10.1137/040617650.

[59] Y. Lin and V. Simoncini. A new subspace iteration method for the algebraic Riccati equation. *Numer. Linear Algebra Appl.*, 22(1):26–47, 2015. doi:10.1002/nla.1936.

[60] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Structured polynomial eigenvalue problems: good vibrations from good linearizations. *SIAM J. Matrix Anal. Appl.*, 28(4):1029–1051, 2006. doi:10.1137/050628362.

[61] S. Massei, D. Palitta, and L. Robol. Solving rank-structured Sylvester and Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 39(4):1564–1590, 2018. doi:10.1137/17M1157155.

[62] V. Mehrmann. A step toward a unified treatment of continuous and discrete time control problems. In *Proceedings of the Fourth Conference of the International Linear Algebra Society (Rotterdam, 1994)*, volume 241/243, pages 749–779, 1996. doi:10.1016/0024-3795(95)00257-X.

[63] V. Mehrmann and F. Poloni. Doubling algorithms with permuted Lagrangian graph bases. *SIAM J. Matrix Anal. Appl.*, 33(3):780–805, 2012. doi:10.1137/110850773.

[64] V. L. Mehrmann. *The autonomous linear quadratic control problem*, volume 163 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, 1991. Theory and numerical solution. doi:10.1007/BFb0039443.

[65] 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, 2002. doi:10.1090/S0025-5718-01-01368-0.

[66] G. T. Nguyen and F. Poloni. Componentwise accurate fluid queue computations using doubling algorithms. *Numer. Math.*, 130(4):763–792, 2015. doi:10.1007/s00211-014-0675-4.

[67] D. Palitta and V. Simoncini. Numerical methods for large-scale Lyapunov equations with symmetric banded data. *SIAM J. Sci. Comput.*, 40(5):A3581–A3608, 2018. doi:10.1137/17M1156575.

[68] D. W. Peaceman and H. H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.*, 3:28–41, 1955.

[69] T. Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 1999/00. doi:10.1137/S1064827598347666.

[70] F. Poloni and T. Reis. A structure-preserving doubling algorithm for Lur'e equations. *Numer. Linear Algebra Appl.*, 23(1):169–186, 2016. doi:10.1002/nla.2019.

[71] A. C. M. Ran and H. L. Trentelman. Linear quadratic problems with indefinite cost for discrete time systems. *SIAM J. Matrix Anal. Appl.*, 14(3):776–797, 1993. doi:10.1137/0614055.

[72] J. D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *Internat. J. Control*, 32(4):677–687, 1980. doi:10.1080/00207178008922881.

[73] L. C. G. Rogers. Fluid models in queueing theory and Wiener-Hopf factorization of Markov chains. *Ann. Appl. Probab.*, 4:390–413, 1994.

[74] A. Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra Appl.*, 58:391–405, 1984. doi:10.1016/0024-3795(84)90221-0.

[75] M. Sadkane. A low-rank Krylov squared Smith method for large-scale discrete-time Lyapunov equations. *Linear Algebra and its Applications*, 436(8):2807 – 2827, 2012. Special Issue dedicated to Danny Sorensen's 65th birthday. URL: http://www.sciencedirect.com/science/article/pii/S0024379511005337, doi:10.1016/j.laa.2011.07.021.

[76] A. H. Sayed and T. Kailath. A survey of spectral factorization methods. *Numer. Linear Algebra Appl.*, 8(6-7):467–496, 2001. Numerical linear algebra techniques for control and signal processing. doi:10.1002/nla.250.

[77] V. Simoncini. Analysis of the rational Krylov subspace projection method for large-scale algebraic Riccati equations. *SIAM J. Matrix Anal. Appl.*, 37(4):1655–1674, 2016. doi:10.1137/16M1059382.

[78] V. Simoncini. Computational methods for linear matrix equations. *SIAM Rev.*, 58(3):377–441, 2016. doi:10.1137/130912839.

[79] R. A. Smith. Matrix equation $XA + BX = C$. *SIAM J. Appl. Math.*, 16:198–201, 1968. doi:10.1137/0116017.

[80] E. L. Wachspress. Iterative solution of the Lyapunov matrix equation. *Appl. Math. Lett.*, 1(1):87–90, 1988. doi:10.1016/0893-9659(88)90183-8.

[81] D. S. Watkins. *The matrix eigenvalue problem*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007. *GR* and Krylov subspace methods. doi:10.1137/1.9780898717808.

[82] J. Willems. Least squares stationary optimal control and the algebraic riccati equation. *IEEE Transactions on Automatic Control*, 16(6):621–634, 1971.

[83] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li. Fast algorithms for hierarchically semiseparable matrices. *Numer. Linear Algebra Appl.*, 17(6):953–976, 2010. doi:10.1002/nla.691.

[84] J. Xue and R.-C. Li. Highly accurate doubling algorithms for M-matrix algebraic Riccati equations. *Numer. Math.*, 135(3):733–767, 2017. doi:10.1007/s00211-016-0815-0.

[85] J. Xue, S. Xu, and R.-C. Li. Accurate solutions of M-matrix algebraic Riccati equations. *Numer. Math.*, 120(4):671–700, 2012. doi:10.1007/s00211-011-0421-0.

[86] J. Xue, S. Xu, and R.-C. Li. Accurate solutions of M-matrix Sylvester equations. *Numer. Math.*, 120(4):639–670, 2012. doi:10.1007/s00211-011-0420-1.