Received: 1 January 2020 | Revised: 8 May 2020 | Accepted: 15 May 2020

DOI: 10.1002/gamm.202000018

ORIGINAL PAPER

# Iterative and doubling algorithms for Riccati-type matrix equations: A comparative introduction

**Federico Poloni**

Dipartimento di Informatica, Università di Pisa, Pisa, Italy

**Correspondence**
Federico Poloni, Dipartimento di Informatica, Largo Pontecorvo, 56127 Pisa, Italy. Email: federico.poloni@unipi.it

**Abstract**

We review a family of algorithms for Lyapunov- and Riccati-type equations which are all related to each other by the idea of *doubling*: they construct the iterate $Q_k = X_{2^k}$ of another naturally-arising fixed-point iteration $(X_h)$ via a sort of repeated squaring. The equations we consider are Stein equations $X - A^* X A = Q$, Lyapunov equations $A^* X + X A + Q = 0$, discrete-time algebraic Riccati equations $X = Q + A^* X(I + G X)^{-1} A$, continuous-time algebraic Riccati equations $Q + A^* X + X A - X G X = 0$, palindromic quadratic matrix equations $A + Q Y + A^* Y^2 = 0$, and nonlinear matrix equations $X + A^* X^{-1} A = Q$. We draw comparisons among these algorithms, highlight the connections between them and to other algorithms such as subspace iteration, and discuss open issues in their theory.

**KEYWORDS**

algebraic Riccati equation, control theory, doubling algorithm, numerical linear algebra

## 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 [64], a review book edited by Bittanti et al. [23], various treatises which consider them from different points of view such as [1,4,5,20,25,39,58,74], and recently also a book devoted specifically to doubling [57].

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. We focus on introducing 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; we try to present the proofs only up to a level of detail that makes the results plausible and allows the interested reader to fill the gaps.

The basic idea behind doubling algorithms can be explained through the "model problem" of computing $w_h = M^{2^h} v$ for a certain matrix $M \in \mathbb{C}^{n\times n}$, $v \in \mathbb{C}^n$, and $h \in \mathbb{N}$. There are two possible ways to approach this computation:

(1) Compute $v_{k+1} = M v_k$, for $k = 0, 1, \dots, 2^h - 1$ starting from $v_0 = v$; then the result is $w_h = v_{2^h}$.

  

(2) Compute $M_{k+1} = (M_k)^2$, for $k = 0, 1, \ldots, h-1$, starting from $M_0 = M$; then the result is $w_h = M_h v$ (repeated squaring).

It is easy to verify that $M_k v = v_{2^k}$ for each $k$. Hence $k$ iterations of (2) correspond to $2^k$ iterations of (1). We say that (2) is a *squaring variant*, or *doubling variant*, of (1). Each of the two versions has its own pros and cons, and in different contexts one or the other may be preferred. If $h$ is moderate and $M$ is large and sparse, one should favor variant (1): sparse matrix-vector products can be computed efficiently, while the matrices $M_k$ would become dense rather quickly, and one would need to compute and store all their $n^2$ entries. On the other hand, if $M$ is a dense matrix of nontrivial size (let us say $n \approx 10^3$ or $10^4$) and $h$ is reasonably large, then variant (2) wins: fewer iterations are needed, and the resulting computations are rich in matrix multiplications and BLAS level-3 operations, hence they can be performed on modern computers even more efficiently than their flop counts suggest. This problem is an oversimplified version, but it captures the spirit of doubling algorithms, and explains perfectly in which cases they work best.

Regarding competing methods: we mention briefly in our exposition Newton-type algorithms, ADI, and Krylov-type algorithms. We do not treat here direct methods, including Schur decomposition-based methods [66,78,91], methods based on structured QR [26,28,71], on symplectic URV decompositions [15,33], and linear matrix inequalities [25]. Although these competitors may be among the best methods for dense problems, they do not fit the scope of our exposition and they do not lend themselves to an immediate comparison with the algorithms that we discuss.

The equations that we treat arise mostly from the study of dynamical systems, both in discrete and continuous time. In our exposition, we chose to start from the discrete-time versions: while continuous-time Riccati equations are simpler and more common in literature, it is more natural to start from discrete-time problems in this context. Indeed, when we discuss algorithms for continuous-time problems we shall see that often the first step is a reduction to a discrete-time problem (possibly implicit).

In the following, we use the notation $A \succ B$ (resp. $A \succcurlyeq B$) 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 $\mathrm{LHP} = \{z \in \mathbb{C} : \mathrm{Re}(z) < 0\}$ 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 $M$, that is, 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 $\mathrm{vec}(MXN) = (N^\top \otimes M)\mathrm{vec}(X)$ [ [48], 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*).

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

for $A, X, Q \in \mathbb{C}^{n\times n}$. This equation often arises in the study of discrete-time constant-coefficient linear systems

$$x_{k+1} = A x_k. \tag{2}$$

A classical application of Stein equations is the following. If $X$ solves (1), then by multiplying by $x_k^*$ and $x_k$ on both sides one sees that $V(x) := x^* X x$ is decreasing over the trajectories of (2), that is, $V(x_{k+1}) \le V(x_k)$. This fact can be used to prove stability of the dynamical system (2).

### 2.1 | Solution properties

The Stein equation (1) is linear, and can be rewritten using Kronecker products as

$$(I_{n^2} - A^\top \otimes A^*)\,\mathrm{vec}(X) = \mathrm{vec}(Q). \tag{3}$$

If $A = UTU^*$ is a Schur factorization of $A$, then we can factor the system matrix as

$$I_{n^2} - M = I_{n^2} - A^\top \otimes A^* = (\bar{U} \otimes U)(I_{n^2} - T^\top \otimes T^*)(U^\top \otimes U^*), \qquad M = A^\top \otimes A^*, \tag{4}$$

which is a Schur-like factorization where the middle term is lower triangular. One can tell when $I - M$ is invertible by looking at its diagonal entries: $I - M$ is invertible (and hence (1) is uniquely solvable) if and only if $\lambda_i \overline{\lambda_j} \neq 1$ for each pair of eigenvalues $\lambda_i, \lambda_j$ of $A$. This holds, in particular, when $\rho(A) < 1$. When the latter condition holds, we can apply the Neumann inversion formula

$$(I - M)^{-1} = I + M + M^2 + \cdots, \tag{5}$$

which gives (after de-vectorization) an expression for the unique solution as an infinite series

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

It is apparent from (6) that $X \succcurlyeq 0$. A reverse result holds, but with strict inequalities: if (1) holds with $X \succ 0$ and $Q \succ 0$, then $\rho(A) < 1$ [39, Exercise 7.10].

## 2.2 | Algorithms

As discussed in the introduction, we do not describe here direct algorithms of the Bartels-Stewart family [6,37,45,47] (which, essentially, exploit the decomposition (4) to reduce the cost of solving (3) from $\mathcal{O}(n^6)$ to $\mathcal{O}(n^3)$) even if they are often the best performing ones for dense linear (Stein or Lyapunov) equations. Rather, 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

$$X_0 = 0, \qquad X_{k+1} = Q + A^* X_k A, \tag{7}$$

known as *Smith method* [90]. It is easy to see that the $k$th iterate $X_k$ is the partial sum of (6) (and (5)) truncated to $k+1$ terms, thus convergence is monotonic, that is, $Q = X_0 \preccurlyeq X_1 \preccurlyeq X_2 \preccurlyeq \cdots \preccurlyeq X$. Moreover, some manipulations give

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

or, devectorizing,

$$X - X_k = (A^*)^{k+1} X A^{k+1}. \tag{8}$$

This relation (8) implies $\|X - X_k\| = \mathcal{O}(r^k)$ for each $r > \rho(A)^2$, so convergence is linear when $\rho(A) < 1$, and it typically slows down when $\rho(A) \approx 1$.

A doubling variant comes from splitting the partial sums into two halves. The truncated sums of (5) to $2^{k+1}$ terms 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

$$A_0 = A, \qquad A_{k+1} = A_k^2, \tag{9a}$$

$$Q_0 = Q, \qquad Q_{k+1} = Q_k + A_k^* Q_k A_k. \tag{9b}$$

In view of the definitions, we have $Q_k = X_{2^k}$; so this method computes the $2^k$th iterate of the Smith method directly with $\mathcal{O}(k)$ operations, without going through all intermediate ones. Convergence is quadratic: $\|X - Q_k\| = \mathcal{O}(r^{2^k})$ for each $r > \rho(A)^2$. The method (9) is known as *squared Smith*. It has been used in the context of parallel and high-performance

computing [16], and reappeared in recent years, when it has been used for large and sparse equations [12,80,86] in combination with Krylov methods.

## 3 | LYAPUNOV EQUATIONS

Lyapunov equations

$$A^* X + X A + Q = 0, \quad Q = Q^* \succcurlyeq 0 \tag{10}$$

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

$$\frac{d}{dt}x(t) = A\,x(t). \tag{11}$$

A classical application is the following. If $X$ solves (10), by multiplying on by $x(t)^*$ and $x(t)$ on both sides one sees that $V(x) := x^*Xx$ is decreasing over the trajectories of (11), that is, $\frac{d}{dt}V(x(t)) \leq 0$. This fact can be used to prove stability of the dynamical system (11). Today stability is more often proved by computing eigenvalues, but Stein equations (1) and Lyapunov equations (10) have survived in many other applications in systems and control theory, for instance in model order reduction [8,50,89], or as the inner step in Newton methods for other equations (see for instance (46) in the following).

### 3.1 | Solution properties

Using Kronecker products, one can rewrite (10) as

$$(I_n \otimes A^* + A^\top \otimes I_n)\,\mathrm{vec}\,(X) = -\mathrm{vec}(Q), \tag{12}$$

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

$$I_n \otimes A^* + A^\top \otimes I_n = (\bar{U} \otimes U)(I_n \otimes T^* + T^\top \otimes I_n)(U^\top \otimes U^*). \tag{13}$$

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 (10) is uniquely solvable) if and only if $\overline{\lambda_i} + \lambda_j \neq 0$ for each pair of eigenvalues $\lambda_i, \lambda_j$ of $A$. This holds, in particular, if the eigenvalues of $A$ all lie in LHP $= \{z \in \mathbb{C} : \mathrm{Re}(z) < 0\}$. When the latter condition holds, an analogue of (6) is

$$X = \int_0^\infty \exp(A^*t)Q\exp(At)\;dt. \tag{14}$$

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 $\Lambda(A) \subset$ LHP for continuous-time systems. Perhaps a bit less evident is the equivalence between the condition $\overline{\lambda_i} + \lambda_j \neq 0$ (ie, no two eigenvalues of $A$ are mapped into each other by reflection with respect to the imaginary axis) and $\lambda_i\overline{\lambda_j} \neq 1$ (ie, no two eigenvalues of $A$ are mapped into each other by circle inversion with respect to the complex unit circle).

Lyapunov equations can be turned into Stein equations and vice versa. Indeed, for a given $\tau \in \mathbb{C}$, (10) is equivalent to

$$(A^* - \tau I)\,X\,(A - \overline{\tau}I) - (A^* + \overline{\tau}I)\,X\,(A + \tau I) - 2\mathrm{Re}(\tau)Q = 0,$$

or, if $A - \overline{\tau}I$ is invertible,

$$X - c(A)^*\,X\,c\,(A) = 2\mathrm{Re}(\tau)(A^* - \tau I)^{-1}Q\,(A - \overline{\tau}I)^{-1}, \qquad c\,(A) = (A + \tau I)(A - \overline{\tau}I)^{-1} = (A - \overline{\tau}I)^{-1}(A + \tau I). \tag{15}$$

If $\tau \in \mathrm{RHP}$, then the right-hand side is positive semidefinite and (15) is a Stein equation. The stability properties of $c(A)$ can be explicitly related to those of $A$ via the following lemma.

**Lemma 1** (Properties of Cayley transforms). *Let* $\tau \in \mathrm{RHP}$. *Then*,

(1) *for* $\lambda \in \mathbb{C}$, *we have* $|c(\lambda)| = |\frac{\lambda+\tau}{\lambda-\overline{\tau}}| < 1$ *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, $-\tau$ and $\overline{\tau}$ are symmetric with respect to the imaginary axis, with $-\tau$ lying to its left. Thus a point $\lambda \in \mathbb{C}$ is closer to $-\tau$ than to $\overline{\tau}$ if and only if it lies in LHP. Part (2) follows from facts on the behavior of eigenvalues of a matrix under rational functions [64, Proposition 1.7.3], which we will often use also in the following.

Another important property of the solutions $X$ of Lyapunov and Stein equations is the decay of their singular values in many practical cases. We defer its discussion to the following section, since a proof follows from the properties of certain solution algorithms.

## 3.2 | Algorithms

As in the Stein case, one can implement a direct $\mathcal{O}(n^3)$ Bartels-Stewart algorithm [6] by exploiting the decomposition (13): the two outer factors have Kronecker product structure, and the inner factor is lower triangular, allowing for forward substitution. An interesting variant allows one to compute the Cholesky factor of $X$ directly from the one of $Q$ [55].

Again, we focus our interest on iterative algorithms. We will assume $\Lambda(A) \subset \mathrm{LHP}$. Then, thanks to Lemma 1, we have $\rho(c(A)) < 1$, so we can apply the Smith method (7) to (15). In addition, we can change the value of $\tau$ at each iteration. The resulting algorithm is known as *ADI iteration* [79,92]:

$$\begin{aligned} &X_0 = 0, && X_{k+1} = Q_k + c_k(A)^* X_k c_k(A), \\ &Q_k = 2\mathrm{Re}(\tau_k)(A^* - \tau_k I)^{-1} Q (A - \overline{\tau}_k I)^{-1}, && c_k(A) = (A + \tau_k I)(A - \overline{\tau}_k I)^{-1} = (A - \overline{\tau}_k I)^{-1}(A + \tau_k I). \end{aligned} \tag{16}$$

The sequence of *shifts* $\tau_k \in \mathrm{RHP}$ can be chosen arbitrarily, with the only condition that $\overline{\tau}_k \notin \Lambda(A)$. By writing a recurrence for the error $E_k = X - X_k$, one sees that

$$E_k = r_{k+1}(A)^* E_0 r_{k+1}(A) = r_{k+1}(A)^* X r_{k+1}(A), \quad r_{k+1}(A) = c_k(A) \cdots c_1(A) c_0(A), \tag{17}$$

a formula which generalizes (8). 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

$$\|r_k(A)\| = \max_{\lambda\in\Lambda(A)} |r_k(\lambda)|, \qquad \|r_k(A)^*\| = \|r_k(-A^*)^{-1}\| = \frac{1}{\min_{\lambda\in\Lambda(A)} |r_k(-\lambda^*)|}.$$

If one knows a region $E \subset \mathrm{LHP}$ that encloses the eigenvalues of $A$, the optimal choice of $r_k$ is the degree-$k$ rational function that minimizes

$$\frac{\sup_{z\in E} |r_k(z)|}{\inf_{z\in -E^*} |r_k(z)|}, \tag{18}$$

that is, 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 (18) decays as $\rho^k$, 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 [43,92]. When $A$ is nonnormal, 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 $Q$ has low rank. One usually writes $Q = C^*C$, where $C \in \mathbb{C}^{p\times n}$ is a short-fat matrix, motivated by a standard notation in control theory. A decomposition $X_k = Z_k Z_k^*$ can be derived from (16), and reads

$$\begin{aligned} Z_k &= \begin{bmatrix} \sqrt{2\mathrm{Re}(\tau_{k-1})}(A^* - \tau_{k-1}I)^{-1}C^*, & c_{k-1}(A)^* Z_{k-1} \end{bmatrix} \\ &= \Big[\sqrt{2\mathrm{Re}(\tau_{k-1})}(A^* - \tau_{k-1}I)^{-1}C^*, \sqrt{2\mathrm{Re}(\tau_{k-2})}(A^* - \tau_{k-1}I)^{-1}(A^* + \overline{\tau}_{k-1}I)(A^* - \tau_{k-2}I)^{-1}C^*, \dots, \\ &\quad \sqrt{2\mathrm{Re}(\tau_0)}(A^* - \tau_{k-1}I)^{-1}(A^* + \overline{\tau}_{k-1}I)(A^* - \tau_{k-2}I)^{-1}(A^* + \overline{\tau}_{k-2}I)\cdots(A^* - \tau_0 I)^{-1}C^*\Big]. \end{aligned} \tag{19}$$

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 $C^*$. All the factors in parentheses commute: hence that the factors $V_j$ can be computed with the recurrence

$$\begin{aligned} Z_k = \begin{bmatrix} V_1 & V_2 & \dots & V_k \end{bmatrix}, \qquad V_1 &= \sqrt{2\mathrm{Re}(\tau_{k-1})}(A^* - \tau_{k-1}I)^{-1}C^*, \\ V_{j+1} &= \frac{\sqrt{2\mathrm{Re}(\tau_{k-j-1})}}{\sqrt{2\mathrm{Re}(\tau_{k-j})}}(A^* - \tau_{k-j-1}I)^{-1}(A^* + \overline{\tau}_{k-j}I)V_j \\ &= \frac{\sqrt{2\mathrm{Re}(\tau_{k-j-1})}}{\sqrt{2\mathrm{Re}(\tau_{k-j})}}(V_j + (\tau_{k-j-1} + \overline{\tau}_{k-j})(A^* + \tau_{k-j-1}I)^{-1}V_j). \end{aligned} \tag{20}$$

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 $\begin{bmatrix} V_1 & V_2 & \dots & V_j \end{bmatrix}\begin{bmatrix} V_1 & V_2 & \dots & V_j \end{bmatrix}^*$ differs from $X_j$ in (16). Indeed, in this factorized version the shifts appear in reversed order, starting from $\tau_{k-1}$ and ending with $\tau_0$. 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 $p$ linear systems with a shifted matrix $A^* - \tau I$, and we store in memory only the $n \times kp$ matrix $Z_k$. In contrast, iterations such as (9) are not going to be efficient for problems with a large and sparse $A$, since powers of sparse matrices become dense.

The formula (19) displays the relationship between ADI and certain Krylov methods: since the LR-ADI iterates are constructed by applying rational functions of $A^*$ iteratively to $C^*$, the LR-ADI iterate $Z_k$ lies in the so-called *rational Krylov subspace* [85]

$$K_{q,k+1}(A^*, C^*) = \mathrm{span}\{q(A^*)^{-1}p(A^*)C^* \ : \ p \text{ is a polynomial of degree} \le k\}, \tag{21}$$

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 $K_{q,k}(A^*, C^*)$ to which its iterates belong. Once one has chosen suitable shifts and computed an orthogonal basis $U_k$ of $K_{q,k+1}(A^*, C^*)$, (10) 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

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

which is a smaller $(kp \times kp)$ 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 (21) [41,42] (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 or of an extended Krylov subspace

$$K_{k_1,k_2}(A^*, C^*) = \mathrm{span}\{\ell(A^*)C^* \ : \ \ell \text{ is a Laurent polynomial of degrees } (k_1, k_2)\}. \tag{22}$$

Computing a basis for a rational Krylov subspace (21) is more expensive than computing one for an extended Krylov subspace (22): 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 [42].

See also the analysis in Benner et al. [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 (18) decreases exponentially, the formula (17) 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 (19). 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.

### 3.3 | 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 [89] for more details. The literature typically deals with continuous-time Lyapunov equations more often than their discrete-time counterpart; however, Cayley transformations (15) 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 10 into a Stein equation 1 via a Cayley transform (15) and then applying one step of the Smith iteration (7). Hence the squared Smith method (9) 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* $\tau$ in ADI.

It is known that a wise choice of shifts has a major impact on the convergence speed of these algorithms; see for example, Güttel [54]. 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 | DISCRETE-TIME RICCATI EQUATIONS

We consider the equation

$$X = Q + A^* X(I + GX)^{-1}A \qquad G = G^* \succcurlyeq 0, \qquad Q = Q^* \succcurlyeq 0, \qquad A, G, Q, X \in \mathbb{C}^{n\times n}, \tag{23}$$

to be solved for $X = X^* \succcurlyeq 0$. This equation is known as *discrete-time algebraic Riccati equation* (DARE), and arises in various problems connected to discrete-time control theory [39, chapter 10]. Variants in which $G, Q$ are not necessarily positive semidefinite also exist [82,94], but we will not deal with them here to keep our presentation simpler. The nonlinear 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

$$\begin{aligned} X(I + GX)^{-1} &= (I + XG)^{-1}X = X - X(I + GX)^{-1}GX \\ &= X - XBR^{-1/2}(I + R^{-1/2}B^*XBR^{-1/2})^{-1}R^{-1/2}B^*X \\ &= X - XB(R + B^*XB)^{-1}B^*X, \end{aligned} \tag{24}$$

and all these forms can be plugged into (23) 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 (24) are all different ways to rewrite the sum of the converging series $X - XGX + XGXGX - XGXGXGX + \cdots$.

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

## 4.1 | 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 [72].

For each solution $X$ of the DARE (23), it holds that

$$\begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I & G \\ 0 & A^* \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} K, \qquad K = (I + GX)^{-1}A. \tag{25}$$

Equation (25) shows that $\operatorname{Im} \begin{bmatrix} I \\ X \end{bmatrix}$ is an *invariant subspace* of

$$\mathcal{S} = \begin{bmatrix} I & G \\ 0 & A^* \end{bmatrix}^{-1} \begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix}, \tag{26}$$

that is, $\mathcal{S}$ 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 $\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$ for an invariant subspace of $\mathcal{S}$, and if $U_1$ is invertible, then $\begin{bmatrix} I \\ U_2 U_1^{-1} \end{bmatrix}$ is another basis matrix, the equality (25) holds, and $X = U_2 U_1^{-1}$ is a solution of (23). Hence, (23) 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 1** ([64], Corollary 13.1.2 and Theorem 13.1.3). *Assume that $Q \succcurlyeq 0$, $G \succcurlyeq 0$ and $(A, G)$ is d-stabilizable. Then, (23) has a (unique) solution $X_+$ such that*

(1) $X_+ = X_+^* \succcurlyeq 0$;
(2) $X_+ \succcurlyeq X$ *for any other Hermitian solution* $X$;
(3) $\rho((I + GX_+)^{-1}A) \leq 1$.

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

The hypotheses involve two classical definitions from control theory [39]: *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 or the theorem, which is not obvious to prove; we refer the reader to Lancaster and Rodman [64] 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 (26) follow from the fact that it belongs to a certain class of structured matrices. Let $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \in \mathbb{C}^{2n \times 2n}$. A matrix $M \in \mathbb{C}^{2n \times 2n}$ is called *symplectic* if $M^* J M = J$, that is, if it is unitary for the nonstandard scalar product associated to $J$. The following properties hold.

**Lemma 2.**

(1) *A matrix in the form* (26) *is symplectic if and only if $G = G^*$, $Q = Q^*$, and the two blocks called $A, A^*$ in* (26) *are one the conjugate transpose of the other.*
(2) *If $\lambda$ 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* 1 *(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 $(I + G X_+)^{-1}A$ inside the unit circle, and the $n$ eigenvalues*

$\overline{\lambda_i}^{-1}$, $i = 1, 2, \ldots, n$ *outside the unit circle. In particular,* $\begin{bmatrix} I \\ X_+ \end{bmatrix}$ *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 (26) and the definition of symplectic matrix, respectively. To prove Part 3, plug $X_+$ into (25) 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.

## 4.2 | Algorithms

The shape of (23) suggests the iteration

$$X_{k+1} = Q + A^* X_k (I + G X_k)^{-1} A, \qquad X_0 = 0. \tag{27}$$

This iteration can be rewritten in a form analogous to (25):

$$\begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} \begin{bmatrix} I \\ X_{k+1} \end{bmatrix} = \begin{bmatrix} I & G \\ 0 & A^* \end{bmatrix} \begin{bmatrix} I \\ X_k \end{bmatrix} K_k, \qquad K_k = (I + G X_k)^{-1} A. \tag{28}$$

Equivalently, one can write it as

$$\begin{bmatrix} U_{1k} \\ U_{2k} \end{bmatrix} = \mathcal{S}^{-1} \begin{bmatrix} I \\ X_k \end{bmatrix}, \qquad \begin{bmatrix} I \\ X_{k+1} \end{bmatrix} = \begin{bmatrix} U_{1k} \\ U_{2k} \end{bmatrix} (U_{1k})^{-1}. \tag{29}$$

This form highlights a connection with (inverse) subspace iteration (or orthogonal iteration), a classical generalization of the (inverse) power method to find multiple eigenvalues [93]. Indeed, we start from the $2n \times n$ matrix $\begin{bmatrix} I \\ X_0 \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix}$, 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 eg, Watkins [93, section 5.1]) that (29) converges to the invariant subspace associated to the $n$ largest eigenvalues (in modulus) of $\mathcal{S}^{-1}$, that is, the $n$ smallest eigenvalues of $\mathcal{S}$. In view of Part 3 of Lemma 2, this subspace is precisely $\operatorname{Im} \begin{bmatrix} I \\ X_+ \end{bmatrix}$. 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 (27) to the maximal solution, under the d-detectability condition mentioned in Theorem 1, which ensures that there are no eigenvalues on the unit circle.

How would one construct a "squaring" variant of this method? Note that that $\begin{bmatrix} U_{1k} \\ U_{2k} \end{bmatrix} = \mathcal{S}^{-k} \begin{bmatrix} I \\ 0 \end{bmatrix}$; 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 (26)

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

We seek an analogous decomposition for the powers of $\mathcal{S}^{-1}$, that is,

$$\mathcal{S}^{-2^k} = \begin{bmatrix} A_k & 0 \\ -Q_k & I \end{bmatrix}^{-1} \begin{bmatrix} I & G_k \\ 0 & A_k^* \end{bmatrix}. \tag{30}$$

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

**Lemma 3** ([81]). Let $M_1, M_2, N_1, N_2 \in \mathbb{C}^{2n\times n}$. The factorization

$$\begin{bmatrix} M_1 & M_2 \end{bmatrix}^{-1} \begin{bmatrix} N_1 & N_2 \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & I_n \end{bmatrix}^{-1} \begin{bmatrix} I_n & A_{21} \\ 0 & A_{22} \end{bmatrix}, \qquad A_{11}, A_{12}, A_{21}, A_{22} \in \mathbb{C}^{n\times n} \tag{31}$$

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 (31) 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 3 $\left(\text{with } [M_1\, M_2] = I \text{ and } [N_1\ N_2] = \begin{bmatrix} I & G_k \\ 0 & A_k^* \end{bmatrix} \begin{bmatrix} A_k & 0 \\ -Q_k & I \end{bmatrix}^{-1}\right)$ to find a factorization of the term in parentheses in

$$\mathcal{S}^{-2^{k+1}} = \mathcal{S}^{-2^k}\mathcal{S}^{-2^k} = \begin{bmatrix} A_k & 0 \\ -Q_k & I \end{bmatrix}^{-1} \left( \begin{bmatrix} I & G_k \\ 0 & A_k^* \end{bmatrix} \begin{bmatrix} A_k & 0 \\ -Q_k & I \end{bmatrix}^{-1} \right) \begin{bmatrix} I & G_k \\ 0 & A_k^* \end{bmatrix}, \tag{32}$$

and use it to construct a decomposition (30) 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

$$A_{k+1} = A_k(I + G_kQ_k)^{-1}A_k, \tag{33a}$$

$$G_{k+1} = G_k + A_kG_k(I + Q_kG_k)^{-1}A_k^*, \tag{33b}$$

$$Q_{k+1} = Q_k + A_k^*(I + Q_kG_k)^{-1}Q_k\,A_k, \tag{33c}$$

with $A_0 = A, Q_0 = Q, G_0 = G$. These formulas are all we need to formulate a "squaring" version of (27): 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 $2^k$th iterate of (27). It is not difficult to show by induction that $0 \preccurlyeq Q_0 \preccurlyeq Q_1 \preccurlyeq \cdots \le Q_k \preccurlyeq \cdots$, 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 (27) is linear and proportional to the ratio between the absolute values of the $(n+1)$st and $n$th eigenvalue of $\mathcal{S}$, that is, between $\sigma := \rho((I + GX_+)A) < 1$ and its inverse $\sigma^{-1}$. The convergence speed of its doubling variant (33) is then quadratic with the same ratio [57].

The iteration (33), 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 et al. [35], 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 (33), which are equivalent to them thanks to the identities (24).

More general versions of the factorization (30) and of the iteration (33), which guarantee existence and boundedness of the iterates under much weaker conditions, have been explored by Mehrmann and Poloni [73]. Kuo et al. [63] studied

the theoretical properties of the factorization (30) for general powers $S^t$, $t \in \mathbb{R}$, drawing a parallel with the so-called *Toda flow* for the QR algorithm.

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

$$Y = G + A\,Y(I + QY)^{-1}A^*, \tag{34}$$

which is obtained swapping $Q$ with $G$ and $A$ with $A^*$ in (23). Indeed, SDA for the DARE (34) is obtained by swapping $Q$ with $G$ and $A$ with $A^*$ in (33), but this transformation leaves the formulas unchanged. The dual equation 34 appears sometimes in applications together with (23). From the point of view of linear algebra, the most interesting feature of its solution $Y_+$ is that $\begin{bmatrix} -Y_+ \\ I \end{bmatrix}$ is a basis matrix for the invariant subspace associated to the other eigenvalues of $S$, those outside the unit circle. Indeed, (30) gives

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

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

$$S = \begin{bmatrix} -Y_+ & I \\ I & X_+ \end{bmatrix} \begin{bmatrix} ((I + Q\,Y_+)^{-1}A^*)^{-1} & 0 \\ 0 & (I + G\,X_+)^{-1}A \end{bmatrix} \begin{bmatrix} -Y_+ & I \\ I & X_+ \end{bmatrix}^{-1}. \tag{35}$$

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

An interesting limit case is the one when only the first part of Theorem 1 holds, $(Q, A)$ is not $d$-detectable, and the solution $X_+$ exists but $\rho((I + G\,X_+)A) = 1$. In this case, $S$ 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 [64, Theorem 12.2.3], after taking a factorization $G = B\,R^{-1}B^*$ with $R \succ 0$ and using another result in the same book [64, 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 (27) becomes sublinear and (33) becomes linear with rate 1/2. This is shown by Chiang et al. [32]; 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 $Q \succcurlyeq 0$, $G \succcurlyeq 0$ do not hold is trickier, because there are examples where (23) does not have a stabilizing solution and $S$ has Jordan blocks of odd size with eigenvalues on the unit circle: an explicit example is

$$A = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}, \qquad G = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad Q = \begin{bmatrix} 1 & 0 \\ 0 & -10 \end{bmatrix}, \tag{36}$$

which produces a matrix $S$ 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 (36) will produce a perturbed $\tilde{S}$ with two simple eigenvalues $\tilde{\lambda}_\pm$ that satisfy exactly $|\lambda| = 1$, because otherwise Part 2 of Lemma 2 would be violated.

## 5 | CONTINUOUS-TIME RICCATI EQUATIONS

We consider the equation

$$Q + A^*X + XA - XGX = 0, \qquad G = G^* \succcurlyeq 0, \qquad Q = Q^* \succcurlyeq 0, \qquad A, G, Q, X \in \mathbb{C}^{n\times n}, \tag{37}$$

to be solved for $X = X^* \succcurlyeq 0$. This equation is known as *continuous-time algebraic Riccati equation* (CARE), and arises in various problems connected to continuous-time control theory [39, chapter 10]. Despite the very different form,

this equation is a natural analogue of the DARE (23), exactly like Stein and Lyapunov equations are related to each other.

## 5.1 | Solution properties

For each solution $X$ of the CARE, it holds

$$\begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X \end{bmatrix} M, \qquad M = A - GX. \tag{38}$$

Hence, $\begin{bmatrix} I \\ X \end{bmatrix}$ is an invariant subspace of

$$\mathcal{H} = \begin{bmatrix} A & -G \\ -Q & -A^* \end{bmatrix}. \tag{39}$$

Like in the discrete-time case, this relation implies that the $n$ eigenvalues of $M$ are a subset of those of $\mathcal{H}$; moreover, we can construct a solution $X = U_2 U_1^{-1}$ to (37) from an invariant subspace $\operatorname{Im} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$, whenever $U_1$ is invertible. Among all solutions, there is a preferred one.

**Theorem 2** ([64], Theorems 7.9.1, 9.1.2, and 9.1.5). *Assume that $Q \geqslant 0$, $G \geqslant 0$, and $(A, G)$ is c-stabilizable. Then,* (37) *has a (unique) solution $X_+$ such that*

(1) $X_+ = X_+^* \geqslant 0$;
(2) $X_+ \geqslant X$ *for any other Hermitian solution* $X$;
(3) $\Lambda(A - G X_+) \subset \overline{\text{LHP}}$.

*If, in addition, $(Q, A)$ is c-detectable, then $\Lambda(A - G X_+) \subset$ LHP.*

C-stabilizable and c-detectable are defined analogously to their discrete-time counterparts, with the only difference that the domain $\{|\lambda| < 1\}$ is replaced by the left half-plane LHP. Again, we do not comment on this theorem, whose proof is not obvious, and refer the reader to Lancaster and Rodman [64].

Exactly as in the discrete-time case, various interesting properties of the matrix $\mathcal{H}$ in (39) follow from the fact that it belongs to a certain class of structured matrices. A matrix $M \in \mathbb{C}^{2n \times 2n}$ is called *Hamiltonian* if $-M^* J = JM$, that is, if it is skew-self-adjoint with respect to the nonstandard scalar product induced by $J$. The following result holds.

**Lemma 4.**

(1) *A matrix in the form* (39) *is Hamiltonian if and only if $G = G^*$, $Q = Q^*$, and the two matrices called $A, A^*$ in* (39) *are one the conjugate transpose of the other.*
(2) *If $\lambda$ is an eigenvalue of a Hamiltonian matrix with right eigenvector $v$, then $-\overline{\lambda}$ is an eigenvalue of the same matrix with left eigenvector $v^* J$.*
(3) *If the hypotheses of Theorem 2 hold (including the c-detectability one), then the $2n$ eigenvalues of $\mathcal{H}$ are (counting multiplicities) the $n$ eigenvalues $\lambda_1, \ldots, \lambda_n$ of $A - G X_+$ in the left half-plane, and the $n$ eigenvalues $-\overline{\lambda_i}$, $i = 1, \ldots, n$ in the right half-plane. In particular, $\begin{bmatrix} I \\ X_+ \end{bmatrix}$ spans the unique invariant subspace of $\mathcal{H}$ of dimension $n$ all of whose associated eigenvalues lie in the left half-plane.*

Parts 1 and 2 are easy to verify from the block decomposition (39) and the definition of Hamiltonian matrix. To prove Part 3, plug $X_+$ into (25) and notice that $M$ has $n$ eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ in the left half-plane; these are also eigenvalues of $S$. By Part 2, all other eigenvalues lie in the right half-plane.

The similarities between (38) and (25) 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 5.** *Let $A, G = G^*, Q = Q^*$ be given, and take $\tau > 0$. Set*

$$\begin{bmatrix} A_d & G_d \\ -Q_d & A_d^* \end{bmatrix} = \begin{bmatrix} A - \tau I & -G \\ Q & A^* - \tau I \end{bmatrix}^{-1} \begin{bmatrix} A + \tau I & -G \\ Q & A^* + \tau I \end{bmatrix} = I + 2\tau \begin{bmatrix} A - \tau I & -G \\ Q & A^* - \tau I \end{bmatrix}^{-1}. \tag{40}$$

*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 (40) follow from constructing $S := c(\mathcal{H}) = (\mathcal{H} - \tau I)^{-1}(\mathcal{H} + \tau I)$, and then applying Lemma 3 to construct a factorization

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

The matrix $S$ that we have constructed has the same invariant subspaces as $\mathcal{H}$ because $c(\cdot)$ is an invertible rational function: indeed, from (38), it follows that

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

This relation coincides with (25), and shows that a solution $X$ of the CARE is also a solution of the DARE constructed with (40). Thanks to Lemma (1), $M$ 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 [72], 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 5 does not generalize completely to the case when $A_d$ is singular [72, section 6]. By considering the poles of $c(\mathcal{H})$ as a function of $\tau$, one sees that $A_d$ is singular if and only if $\tau \in \Lambda(\mathcal{H})$. When this happens, even if $S$ "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 (25), but only to the subtly weaker form

$$\begin{bmatrix} A_d & 0 \\ -Q_d & I \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} (M - \tau I) = \begin{bmatrix} I & G_d \\ 0 & A_d^* \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} (M + \tau I), \qquad M = A - GX \tag{41}$$

with an additional singular matrix $M - \tau I$ in the left-hand side. Thus we cannot write the equality (25), 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 $\tau > 0$.

## 5.2 | Algorithms

In view of the relation between DAREs and CAREs that we have just outlined, a natural algorithm is using the formulas (40) to convert (37) into an equivalent (23) and solving it using (33). This algorithm has been suggested by Chu et al. [34] 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 [32].

While SDA works well in general, a delicate point is the choice of the shift value $\tau$. In principle almost every choice of $\tau$ works, since $\mathcal{H} - \tau I$ is singular only for at most $2n$ values of $\tau$, but in practice choosing the wrong value of $\tau$ 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 (40), and also from taking $\tau$ too large or too small by orders of magnitude. A heuristic approach based on golden section search has been suggested [34].

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

$$S^2 = c\left(\frac{1}{2}(\mathcal{H} + \tau^2\mathcal{H}^{-1})\right). \tag{42}$$

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 $S^{2^k} = c(\mathcal{H}_k)$, where

$$\mathcal{H}_{k+1} = \frac{1}{2}(\mathcal{H}_k + \tau^2\mathcal{H}_k^{-1}), \qquad \mathcal{H}_0 = \mathcal{H}. \tag{43}$$

Hence one can hold off the Cayley transform and just compute the sequence $\mathcal{H}_k$ directly, starting from (39). This constructs a sequence which represents implicitly $S^{2^k}$.

Constructing the matrices $\mathcal{H}_k$ is numerically much less troublesome than constructing explicitly $S^{2^k}$ or its inverse $S^{-2^k}$. Indeed, it is instructive to consider the behavior of these iterations in a basis in which $\mathcal{H}$ is diagonal (when it exists). Let $\lambda$ be a generic diagonal entry (ie, an eigenvalue) of $\mathcal{H}$. Then, $S = c(\mathcal{H})$ has the corresponding eigenvalue $c(\lambda)$, and $S^{2^k}$ has the eigenvalue $c(\lambda)^{2^k}$. If $\lambda \in$ LHP, then $|c(\lambda)| < 1$ (Lemma 1), and hence $c(\lambda)^{2^k} \to 0$ when $k \to \infty$. Similarly, if $\lambda$ is in the right half-plane, then $|c(\lambda)| > 1$ and $c(\lambda)^{2^k} \to \infty$. Thus $S^{2^k}$ (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 $S$ in its factored form (30). 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* [40,46,83] to solve CAREs consists exactly in computing the iteration (43) 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

$$\mathrm{Im}\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \ker(\mathcal{H}_\infty + \tau I), \qquad U_1, U_2 \in \mathbb{C}^{n\times n}, \qquad X_+ = U_2U_1^{-1}. \tag{44}$$

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 [56, chapter 5], in which one clearly sees that one of the main ingredients is the formula 42 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 (37) 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 $\tau_k$ at each step of (43). We remark that scaling for the sign method is usually presented in the literature in a slightly different form: one replaces (43) with

$$\mathcal{H}_{k+1} = \frac{1}{2}((\tau_k^{-1}\mathcal{H}_k) + (\tau_k^{-1}\mathcal{H}_k)^{-1}). \tag{45}$$

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 (44). Irrespective of formulation, the main result is that a judicious choice of scaling can speed up the convergence of (43) or (45). A cheap and effective choice of scaling, *determinantal scaling*, $\tau_k = (\det \mathcal{H}_k)^{\frac{1}{n}}$ has been suggested by Byers [29]. Other related choices of scaling and their performances have been discussed by Higham [ [56], chapter 5] and Kenney and Laub [60]. 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 et al. [30]. 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 (43) 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 [29]. 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 (40).

### 5.3 | 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 (42), scaling the sign iteration is equivalent to changing the parameter $\tau$ in the Cayley transform. So SDA does incorporate a form of scaling, but only at the first iteration, when one chooses $\tau$.

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 nondoubling algorithms, with the aim of deriving optimal choices for the parameters $\tau$ and $\sigma_k$.

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 [61] consists in determining $X_{k+1}$ by solving at each step the Lyapunov equation

$$(A - GX_k)^*(X_{k+1} - X_k) + (X_{k+1} - X_k)(A - GX_k) = -(Q + A^* X_k + X_k A - X_k G X_k) \tag{46}$$

or the equivalent one

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

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 [51]). 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* [61] (ie, $X_1 \succcurlyeq X_2 \succcurlyeq \cdots \succcurlyeq X_k \succcurlyeq X_{k+1} \succcurlyeq \cdots \succcurlyeq X_+$), not from *below* like the iterates $Q_k$ of SDA in (33c).

Some more recent algorithms for large and sparse CAREs essentially merge the Newton step (46) and the ADI iteration (20) into a single iteration [9,68,88]. It is again unclear whether there is an explicit relation between these two families of methods.

An interesting question is what is the "nondoubling" analogue of the sign method and of SDA. One can convert the CARE to discrete-time using (40) and formulate (27), but to the best of our knowledge this method does not have a more appealing presentation in terms of a simple iterative method for (37), 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 (44), where the subexpression $\mathcal{H} + \tau I$ 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 $\tau$ 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.

## 6 | 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

$$P(z) = Az^{-1} + Q + A^* z, \qquad Q = Q^* \succ 0, \qquad A, Q \in \mathbb{C}^{n\times n}. \tag{47}$$

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

$$P(z) = (zY^* - I)X(z^{-1}Y - I), \qquad X = X^* \succ 0, \qquad X, Y \in \mathbb{C}^{n\times n}, \tag{48}$$

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 (47) and (48) 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

$$0 = A + Q\,Y + A^*Y^2, \tag{49}$$

$$Q = X + A^*\,X^{-1}A. \tag{50}$$

The first one (49) is called *unilateral quadratic matrix equation* [19], while the second one (50) is known with the (rather undescriptive) name of *nonlinear matrix equation* (NME) [52,53,57].

While (49) looks more appealing at first, as it reveals direct ties with the palindromic quadratic eigenvalue problem [52,53,69], it is in fact (50) that reveals more structure: for instance, (50) has Hermitian solutions (see below), while the structure in the solutions of (49) is much less apparent.

## 6.1 | Solution properties

It follows from (48) that $P(\lambda) \succcurlyeq 0$ for each $\lambda$ that belongs to the unit circle (hence $\lambda^{-1} = \overline{\lambda}$), 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 3** ([44], Theorem 2.2). *Assume that $P(z)$ is regular and $P(\lambda) \succcurlyeq 0$ for each $\lambda$ on the unit circle. Then,* (50) *has a (unique) solution $X_+$ such that*

(1) $X_+ = X_+^* \succ 0$;
(2) $X_+ \succcurlyeq X$ *for any other Hermitian solution* $X$;
(3) $\rho(Y) = \rho(-X_+^{-1}A) \leq 1$

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

Once again, we can rewrite (50) as an invariant subspace problem.

$$\begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} 0 & -I \\ A^* & 0 \end{bmatrix} \begin{bmatrix} I \\ X \end{bmatrix} Y, \qquad Y = -X^{-1}A. \tag{51}$$

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

$$\mathcal{S} = \begin{bmatrix} 0 & -I \\ A^* & 0 \end{bmatrix}^{-1} \begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} \tag{52}$$

is symplectic, and so is the slightly more general form

$$\begin{bmatrix} G & -I \\ A^* & 0 \end{bmatrix}^{-1} \begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix}. \tag{53}$$

**Lemma 7.**

(1) *A matrix in the form* (53) *is symplectic if and only if $G = G^*$, $Q = Q^*$, and the two blocks called $A$, $A^*$ in* (26) *are one the conjugate transpose of the other.*

(2) *If $\lambda$ 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 3 hold (including the strict positivity one in the end), then the $2n$ eigenvalues of $S$ are (counting multiplicities) the $n$ eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ of $-X_+^{-1} A$ inside the unit circle, and the $n$ eigenvalues $\overline{\lambda_i}^{-1}$, $i = 1, 2, \dots, n$ 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 2. Indeed, Engwerda et al. [44, section 7] note that (50) can be reduced to a DARE, although it is one that does not fall inside our framework since it has $G \lneq 0$.

## 6.2 | Algorithms

The formulation (50) suggests immediately the iterative algorithm

$$X_{k+1} = Q - A^* X_k^{-1} A. \tag{54}$$

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

$$\mathcal{Q}_m := \underbrace{\begin{bmatrix} Q & A^* & & & \\ A & Q & A^* & & \\ & A & Q & \ddots & \\ & & \ddots & \ddots & A^* \\ & & & A & Q \end{bmatrix}}_{m \text{ blocks}}. \tag{55}$$

It can be seen that $\mathcal{Q}_m$ is positive semidefinite, under the assumptions of Theorem 3: 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 $\Phi$ and properties of Fourier transforms) to a block diagonal matrix that contains $P(z)$ from (47) 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 $Q = X_0 \succcurlyeq X_1 \succcurlyeq X_2 \succcurlyeq \cdots \succcurlyeq X_k \succcurlyeq \cdots$. The sequence $X_k$ is monotonic and bounded from below, hence it converges, and one can show that its limit is $X_+$ [44, section 4] (to do this, verify the property in Point (2) of Theorem 3 by proving that $X_k \succcurlyeq X$ at each step of the iteration).

A doubling variant of (54) 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

$$A_{k+1} = -A_k\, U_k^{-1}\, A_k, \tag{56a}$$

$$Q_{k+1} = Q_k - A_k^*\, U_k^{-1}\, A_k, \tag{56b}$$

$$U_{k+1} = U_k - A_k^*\, U_k^{-1}\, A_k - A_k\, U_k^{-1}\, A_k^*. \tag{56c}$$

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 (54), resulting in $X_{2^k}$, or we make $k$ iterations of (56), 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 et al. [27] to solve certain differential equations, and then later applied to matrix equations similar to (49) and (50) by Bini et al. [17,18,75]. The iteration (56) 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 [67]. We seek formulas to update a factorization of the kind

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

To do this, we write (analogously to (32))

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

and use Lemma 3 (with $[M_1\ M_2] = I_{2n}$) to find a factorization in the form (31) of the term in parentheses, which then combines with the outer terms to produce the sought decomposition. The resulting formulas are

$$A_{k+1} = -A_k(Q_k - G_k)^{-1}A_k, \tag{57a}$$

$$Q_{k+1} = Q_k - A_k^*(Q_k - G_k)^{-1}A_k, \tag{57b}$$

$$G_{k+1} = G_k + A_k(Q_k - G_k)^{-1}A_k^*, \tag{57c}$$

and one sees that they coincide with (56), after setting $U_k = Q_k - G_k$. With an argument analogous to the one in Section 4, one sees that

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

thus $\begin{bmatrix} I \\ Q_k \end{bmatrix}$ converges to a basis of the invariant subspace associated to the eigenvalues of $S$ inside the unit circle.

This formulation (57) is known as *SDA-II* [36,67].

### 6.3 | 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 [4,5,87].

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 (50). 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.

## 7 | NONSYMMETRIC VARIANTS IN APPLIED PROBABILITY

Many of the equations treated here have nonsymmetric variants which appear naturally in queuing theory, a subfield of applied probability. In the analysis of *quasi-birth-death models* [21,65], one encounters equations of the form

$$0 = A + Q\,Y + B\,Y^2, \qquad A, B, Q, Y \in \mathbb{R}^{n\times n}, \tag{58}$$

where $A, B \geq 0$ (we use the notation $M \geq N$ to denote that a matrix $M$ is entrywise larger than $N$, that is, $M_{ij} \geq N_{ij}$ for all $i,j$), and the matrix $-Q$ is an $M$-matrix, that is, $Q_{ij} \geq 0$ 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 [65, chapter 6, 21, section 5.6].

Similarly, the equation

$$Q + BX + XA - XGX = 0, \qquad Q, X \in \mathbb{R}^{m\times n}, \qquad A \in \mathbb{R}^{n\times n}, \qquad B \in \mathbb{R}^{m\times m}, \qquad G \in \mathbb{R}^{n\times m}. \tag{59}$$

appears in the study of so-called *fluid queues*, or *stochastic flow models* [38,59,84]. 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 $G, Q \geqslant 0$, 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 (33) exist because $G_k \geqslant 0$, $Q_k \geqslant 0$; 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.

Equation (23) 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 (59) could have been formulated with an equivalent equation resembling more (23) than (37) 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 (58) 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 et al. [22].

For reasons of space, we cannot give here a complete treatment of these nonsymmetric variants. Huang, Li and Lin [57] 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 [31,76,96-98], relying heavily on the sign and M-matrix structure. The forward stability proof in Nguyen and Poloni [76] is, to date, one of the very few complete stability proofs for a doubling-type algorithm.

## 8 | 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 (33) or (56), 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 [77] consider banded data, and Kressner et al. [62] and Massei et al. [70] consider semi-separable (low-rank off-diagonal blocks) and hierarchically semiseparable structures. Much earlier, Grasedyck et al. [49] 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 [24,95]. These novel techniques may open up new lines of research for doubling-type algorithms.

## ACKNOWLEDGEMENTS

Federico Poloni acknowledges the support of Istituto Nazionale di Alta Matematica (INDAM), and of a PRA (*progetti di ricerca di ateneo*) project of the University of Pisa.

## REFERENCES

[1] H. Abou-Kandil et al., *Matrix Riccati equations*, in *Systems & Control: Foundations & Applications*, Birkhäuser Verlag, Basel, 2003 In control and systems theory.

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

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

[4] H. Bart et al., *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 et al., *A state space approach to canonical factorization with applications,* volume 200 *of operator theory: advances and applications*, Birkhäuser Verlag, Basel, 2010.

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

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

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

[9] P. Benner et al., RADI: A low-rank ADI-type algorithm for large scale algebraic Riccati equations, Numer. Math. **138** (2018), 301–330. https://doi.org/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** (1998), 101–107. https://doi.org/10.1109/9.654908.

[11] P. Benner and R. Byers, An arithmetic for matrix pencils: Theory and new algorithms, Numer. Math. **103** (2006), 539–573. https://doi.org/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** (2014), 645–665. https://doi.org/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** (2008), 755–777. https://doi.org/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** (2009), 1035–1045. https://doi.org/10.1016/j.cam.2009.08.108.

[15] P. Benner, V. Mehrmann, and H. Xu, A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils, Numer. Math. **78** (1998), 329–358. https://doi.org/10.1007/s002110050315.

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

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

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

[19] D. A. Bini et al., On the solution of algebraic Riccati equations arising in fluid queues, Linear Algebra Appl. **413** (2006), 474–494. https://doi.org/10.1016/j.laa.2005.04.019.

[20] 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.

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

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

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

[24] S. Börm, L. Grasedyck, and W. Hackbusch, *Hierarchical matrices*, Vol **21**, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, Germany, 2003.

[25] S. Boyd et al., *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. https://doi.org/10.1137/1.9781611970777.

[26] A. Bunse-Gerstner and V. Mehrmann, A symplectic QR like algorithm for the solution of the real algebraic Riccati equation, IEEE Trans. Automat. Control **31** (1986), 1104–1113. https://doi.org/10.1109/TAC.1986.1104186.

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

[28] R. Byers, A Hamiltonian QR algorithm, SIAM J. Sci. Stat. Comput. **7** (1986), 212–229. https://doi.org/10.1137/0907015.

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

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

[31] 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** (2019), 1–45. https://doi.org/10.1016/j.laa.2019.08.018.

[32] C.-Y. Chiang et al., Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case, SIAM J. Matrix Anal. Appl. **31** (2009), 227–247. https://doi.org/10.1137/080717304.

[33] D. Chu, X. Liu, and V. Mehrmann, A numerical method for computing the Hamiltonian Schur form, Numer. Math. **105** (2007), 375–412. https://doi.org/10.1007/s00211-006-0043-0.

[34] 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** (2005), 55–80. https://doi.org/10.1016/j.laa.2004.10.010.

[35] E. K.-W. Chu et al., Structure-preserving algorithms for periodic discrete-time algebraic Riccati equations, Int. J. Control **77** (2004), 767–788. https://doi.org/10.1080/00207170410001714988.

[36] E. K.-W. Chu et al., Vibration of fast trains, palindromic eigenvalue problems and structure-preserving doubling algorithms, J. Comput. Appl. Math. **219** (2008), 237–252. https://doi.org/10.1016/j.cam.2007.07.016.

[37] K.-W. E. Chu, The solution of the matrix equations $AXB - CXD = E$ and $(YA - DZ, YC - BZ) = (E, F)$, Linear Algebra Appl. **93** (1987), 93–105. https://doi.org/10.1016/S0024-3795(87)90314-4.

[38] A. da Silva Soares, Fluid queues – Building upon the analogy with QBD processes, Ph.D Thesis, 2005.

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

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

[41] 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** (2011), 1875–1898. https://doi.org/10.1137/100813257.

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

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

[44] 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** (1993), 255–275. https://doi.org/10.1016/0024-3795(93)90295-Y.

[45] M. A. Epton, Methods for the solution of $AXD - BXC = E$ and its application in the numerical solution of implicit ordinary differential equations, BIT **20** (1980), 341–345. https://doi.org/10.1007/BF01932775.

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

[47] J. D. Gardiner et al., Solution of the Sylvester matrix equation $AXBT + CXDT=E$, ACM Trans. Math. Softw **18** (1992), 223–231. https://doi.org/10.1145/146847.146929.

[48] G. H. Golub and C. F. Van Loan, Matrix computations, in *Johns Hopkins Studies in the Mathematical Sciences*, 4th ed., Johns Hopkins University Press, Baltimore, MD, 2013.

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

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

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

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

[53] 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** (2010), 2784–2801. https://doi.org/10.1137/090763196.

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

[55] S. J. Hammarling, Numerical solution of the stable, nonnegative definite Lyapunov equation, IMA J. Numer. Anal. **2** (1982), 303–323. https://doi.org/10.1093/imanum/2.3.303.

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

[57] 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. https://doi.org/10.1137/1.9781611975369.

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

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

[60] 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** (1992), 698–706. https://doi.org/10.1137/0613044.

[61] D. Kleinman, On an iterative technique for Riccati equation computations, IEEE Trans. Automat. Control **13** (1968), 114–115. https://doi.org/10.1109/TAC.1968.1098829.

[62] 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** (2019), A848–A876. https://doi.org/10.1137/17M1161038.

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

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

[65] 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)*, American Statistical Association, Philadelphia, PA, 1999.

[66] A. J. Laub, A Schur method for solving algebraic Riccati equations, IEEE Trans. Automat. Control **24** (1979), 913–921. https://doi.org/10.1109/TAC.1979.1102178.

[67] 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** (2006), 26–39. https://doi.org/10.1137/040617650.

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

[69] D. S. Mackey et al., Structured polynomial eigenvalue problems: Good vibrations from good linearizations, SIAM J. Matrix Anal. Appl. **28** (2006), 1029–1051. https://doi.org/10.1137/050628362.

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

[71] V. Mehrmann, A symplectic orthogonal method for single input or single output discrete time optimal quadratic control problems, SIAM J. Matrix Anal. Appl. **9** (1988), 221–247. https://doi.org/10.1137/0609019.

[72] V. Mehrmann, *A step toward a unified treatment of continuous and discrete time control problems*. Proceedings of the 4th Conference of the International Linear Algebra Society (Rotterdam, 1994), vol. 241/243, 1996, pp. 749–779. https://doi.org/10.1016/0024-3795(95)00257-X.

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

[74] V. L. Mehrmann, *The autonomous linear quadratic control problem, volume 163 of lecture notes in control and information sciences*, Springer-Verlag, Berlin, Germany, 1991. Theory and numerical solution. https://doi.org/10.1007/BFb0039443.

[75] B. Meini, Efficient computation of the extreme solutions $X + A^*X^{-1}A = Q$ and $X - A^*X^{-1}A = Q$, Math. Comp. **71** (2002), 1189–1204. https://doi.org/10.1090/S0025-5718-01-01368-0.

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

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

[78] T. Pappas, A. J. Laub, and N. R. Sandell Jr., On the numerical solution of the discrete-time algebraic Riccati equation, IEEE Trans. Automat. Control **25** (1980), 631–641. https://doi.org/10.1109/TAC.1980.1102434.

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

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

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

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

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

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

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

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

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

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

[89] V. Simoncini, Computational methods for linear matrix equations, SIAM Rev. **58** (2016), 377–441. https://doi.org/10.1137/130912839.
[90] R. A. Smith, Matrix equation $XA + BXC$, SIAM J. Appl. Math. **16** (1968), 198–201. https://doi.org/10.1137/0116017.
[91] P. Van Dooren, A generalized eigenvalue approach for solving Riccati equations, SIAM J. Sci. Stat. Comput. **2** (1981), 121–135. https://doi.org/10.1137/0902010.
[92] E. L. Wachspress, Iterative solution of the Lyapunov matrix equation, Appl. Math. Lett. **1** (1988), 87–90. https://doi.org/10.1016/0893-9659(88)90183-8.
[93] D. S. Watkins, *The matrix eigenvalue problem*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007. GR and Krylov subspace methods. https://doi.org/10.1137/1.9780898717808.
[94] J. Willems, Least squares stationary optimal control and the algebraic riccati equation, IEEE Trans. Automat. Control **16** (1971), 621–634.
[95] J. Xia et al., Fast algorithms for hierarchically semiseparable matrices, Numer. Linear Algebra Appl. **17** (2010), 953–976. https://doi.org/10.1002/nla.691.
[96] J. Xue and R.-C. Li, Highly accurate doubling algorithms for $M$-matrix algebraic Riccati equations, Numer. Math. **135** (2017), 733–767. https://doi.org/10.1007/s00211-016-0815-0.
[97] J. Xue, S. Xu, and R.-C. Li, Accurate solutions of $M$-matrix algebraic Riccati equations, Numer. Math. **120** (2012), 671–700. https://doi.org/10.1007/s00211-011-0421-0.
[98] J. Xue, S. Xu, and R.-C. Li, Accurate solutions of $M$-matrix Sylvester equations, Numer. Math. **120** (2012), 639–670. https://doi.org/10.1007/s00211-011-0420-1.

**How to cite this article:** Poloni F. Iterative and doubling algorithms for Riccati-type matrix equations: A comparative introduction. *GAMM-Mitteilungen*. 2020;e202000018. https://doi.org/10.1002/gamm.202000018