Zentrum für Technomathematik
Fachbereich 3 – Mathematik und Informatik

Computational Methods for Linear–Quadratic Optimization

Peter Benner

Report 98–04

Berichte aus der Technomathematik

Report 98–04 August 1998

Computational Methods for Linear–Quadratic Optimization

Peter Benner
Zentrum für Technomathematik
Fachbereich 3 – Mathematik und Informatik
Universität Bremen
28334 Bremen
Germany
E-mail: benner@math.uni-bremen.de

Abstract

In this paper we survey some computational methods for linear-quadratic optimization problems as they appear in control theory. As a model we use the classical linear-quadratic regulator (LQR) problem. This is an optimal control problem for a linear time-invariant dynamical system in the sense that a quadratic performance criterion is to be minimized. Most of the modern (robust) control problems for linear-time invariant systems can also be considered as optimization problems with quadratic performance criterion and their solution often requires the same basic numerical methods as the LQR problem. We review several alternative problems that provide a solution to the LQR problem. From the point of view of numerical computations the most intriguing result is that the optimal control can be derived from a particular solution of an algebraic Riccati equation. The desired solutions of these algebraic Riccati equations can be obtained by methods of numerical linear algebra and therefore the solution can be determined without discretization error and with low computational cost. Different strategies that re-present the current state of the art for the numerical solution of these algebraic Riccati equations are discussed.

Keywords. linear-quadratic optimal control, optimization, algebraic Riccati equation, Hamiltonian eigenproblem, numerical methods.

Mathematics Subject Classification (1991). 93–01, 49N05, 49N10, 65F15, 65H10, 65K10, 93B40, 93C60.

1 Introduction

We consider the numerical solution of the continuous-time autonomous linear-quadratic optimal control problem

Minimize

$$\mathcal{J}(x^0,u) \;=\; \frac{1}{2}\int_0^{t_f} \left(y(t)^T Q y(t) + u(t)^T R u(t)\right)\, dt \tag{1}$$

subject to the dynamics

$$\begin{aligned} \dot{x}(t) &= Ax(t)+Bu(t), \qquad t>0, \quad x(0)=x^0, && (2)\\ y(t) &= Cx(t), \qquad t\geq 0, && (3) \end{aligned}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $Q \in \mathbb{R}^{p\times p}$, $R \in \mathbb{R}^{m\times m}$, $Q = Q^T$, $R = R^T$, and $R \geq 0$, i.e., R is positive semidefinite. Moreover,

$$u \in \mathcal{PC}_m[0,t_f] \;:=\; \{u(t)\in\mathbb{R}^m;\; u(t) \text{ piecewise continuous on } [0,t_f]\}.$$

The system (2)–(3) describes a *linear time-invariant dynamical system* where $x(t) \in \mathbb{R}^n$ are the *states*, $u(t) \in \mathbb{R}^m$ are the *inputs* (or *controls*) and $y(t) \in \mathbb{R}^p$ are the *outputs* of the system.

This problem serves as the standard problem in classical control theory since the fundamental work of Kalman and Bucy [39, 40] and has been treated in numerous books and publications since then; see, e.g., [3, 5, 27, 62] and many more. Though linear systems seldom describe reality, the analysis and synthesis problems for nonlinear systems are frequently solved using linearizations around working points. Solving these linearized problems requires again working with the above model. Some classes of nonlinear problems can even be tackled directly with a linear model [51].

The computational methods for solving the optimal control (linear-quadratic optimization) problem (1)–(3) mostly make use of the fact that under moderate assumptions, the optimal control is given by the feedback law

$$u_*(t) \;:=\; -R^{-1}B^T X_*(t)x(t), \qquad t\geq 0, \tag{4}$$

where for $t_f < \infty$, $X_*(t)$ is the unique solution of the *Riccati matrix differential equation*

$$\dot{X}(t) \;=\; \mathcal{R}(X(t)) \;:=\; F + A^T X(t) + X(t)A - X(t)GX(t), \tag{5}$$

with terminal condition $X(t_f) = 0$ while for $t_f = \infty$, $X_*(t) \equiv X_*$ is time-invariant and given as a particular solution of the *algebraic Riccati equation* (ARE)

$$0 \;=\; \mathcal{R}(X) \;:=\; F + A^T X + XA - XGX, \tag{6}$$

where $F := C^T Q C$ and $G := B R^{-1} B^T$. The solution of (6) yielding the optimal control as given in (4) is the unique *stabilizing* solution X_*, i.e., $A - GX_*$ is stable in the sense that all its eigenvalues are in the open left half plane $\mathbb{C}^-$. The performance index is then given by $\mathcal{J}(x^0, u_*) = \frac{1}{2} x_0^T X_* x_0$ while the trajectory implied by the optimal control is given by inserting (4) into (2) and then integrating, yielding

$$x_*(t) = e^{(A-GX_*)t} x_0, \qquad t \geq 0. \tag{7}$$

Several extensions of the optimal control problem (1)–(3) can also be solved via this approach; see, e.g., [69] for a nice overview of such extensions.

During the past 20 years more sophisticated control strategies have been developed; see, e.g., [30, 36, 68, 72]. These strategies take into account robustness of a control law (given as a *controller* or *regulator*) with respect to external perturbations. The computation of such modern $\mathcal{H}_2$- and $\mathcal{H}_\infty$-controllers frequently also boils down to the solution of AREs of the form (6), just the coefficient matrices A, G, and F are derived in a different way and are more or less complicated expressions in terms of the system matrices. Therefore, the computational methods for optimal control problems as given in (1)–(3) carry over to robust control problems.

In most control applications, the model (1)–(3) is used to study the long-time behavior of the (controlled) physical system. We will therefore focus on the infinite time horizon case $t_f = \infty$. Most optimal and robust control problems for linear, time-invariant systems with infinite time horizon can be expressed as linear-quadratic optimization problems that can be solved via AREs. In the survey of numerical methods for linear-quadratic optimization we will therefore concentrate on solution methods for algebraic Riccati equations of the form (6).

The remainder of this paper is organized as follows: in Section 2 we will summarize some basic properties of linear time-invariant systems and then see how the optimal control problem (1)–(3) can be transformed to other problems which all provide solution strategies of the optimization problem. The most frequently used strategy is the approach based on solving AREs. Therefore we review some basic properties of these quadratic matrix equations in Section 3. Computational methods for the numerical solution of AREs are then discussed in Section 4. Some conclusions are given in Section 5.

2 Linear–Quadratic Optimal Control and Related Problems

First, we introduce some notation from linear algebra. By $\lambda(A)$ we denote the spectrum of a square matrix $A \in \mathbb{R}^{n \times n}$, i.e., the set of its eigenvalues. Analogously, we write $\lambda(A, E)$ for the generalized spectrum of a matrix pencil $A - \lambda E$, i.e., the set of its finite and infinite eigenvalues. A matrix is called *stable* if $\lambda(A) \subset \mathbb{C}^-$.

A *linear, time-invariant system* (LTI system) in *state-space presentation* is given by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \qquad t > 0, \qquad x(0) = x_0, \\ y(t) &= Cx(t) + Du(t), \qquad t \geq 0, \end{aligned} \tag{8}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$. Here, we will assume $D = 0$. The case $D \neq 0$ does not impose any additional mathematical difficulties (see, e.g., [69]), only the notation becomes more complicated.

We will also need some properties of LTI systems.

Definition 2.1 *Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$.*

a) The following conditions are equivalent to the controllability *of the matrix pair (A, B):*

(i) For all $x_1 \in \mathbb{R}^n$ there exist $t_1 \geq 0$ and $u \in \mathcal{PC}_m[0, t_1]$ such that $x(t_1) = x_1$.

(ii) $\operatorname{rank}\left([B, AB, A^2B, \ldots, A^{n-1}B]\right) = n$.

(iii) $\operatorname{rank}\left([A - \lambda I_n, B]\right)) = n$ *for all* $\lambda \in \mathbb{C}$.

b) The following conditions are equivalent to the observability *of the matrix pair (C, A):*

(i) The matrix pair (A^T, C^T) is controllable.

(ii) $\operatorname{rank}\left([C^T, (CA)^T, (CA^2)^T, \ldots, (CA^{n-1})^T]^T\right) = n$.

(iii) $\operatorname{rank}\left([A^T - \lambda I, C^T]^T\right) = n$ *for all* $\lambda \in \mathbb{C}$.

c) The following conditions are equivalent to the stabilizability *of the matrix pair (A, B):*

(i) $\operatorname{rank}\left([A - \lambda I, B]\right)) = n$ *for all* $\lambda \in \mathbb{C}$ *with* $Re(\lambda) \geq 0$.

(ii) There exists $K \in \mathbb{R}^{m\times n}$ such that $A + BK$ is stable.

d) The following conditions are equivalent to the detectability *of the matrix pair (C, A):*

(i) The matrix pair (A^T, C^T) is stabilizable.

(ii) $\operatorname{rank}\left([A^T - \lambda I, C^T]^T\right) = n$ *for all* $\lambda \in \mathbb{C}$, $Re(\lambda) \geq 0$.

(iii) There exists $K \in \mathbb{R}^{n\times p}$ such that $A + KC$ is stable.

(iv) If $x(t)$ is a solution of $\dot{x} = Ax$ and $Cx(t) \equiv 0$, then $\lim_{t\to\infty} x(t) = 0$.

e) A matrix $K \in \mathbb{R}^{m\times n}$ is stabilizing *for (A, B) iff $A + BK$ is stable.*

In the following subsections, we will review the standard theory of the optimal control problem (1)–(3) and its relation to other (under certain assumptions) mathematically equivalent problems. This theory can also be found in many textbooks (e.g., [3, 5, 27, 61, 69] and numerous others). We will use a compact presentation, following in part the derivation in [62]. First, we turn our attention to the problem of finding an optimal control $u(t)$ for (1)–(3).

2.1 Existence of solutions of the linear-quadratic optimal control problem

Consider the general *cost functional* given by

$$\mathcal{J}(u(\cdot)) \;=\; \int_0^{t_f} g(t,x,u)\,dt$$

where the *system* is described by the set of ordinary differential equations

$$\dot{x}(t) \;=\; f(t,x,u),$$

with *initial condition* $x(0) \;=\; x^0$ and no *target condition* $x(t_f)$ is prescribed.

In our case, the function g is given by

$$\begin{aligned} g(t,x,u) \;&\equiv\; g(x,u) \;\equiv\; g(x(t),u(t)) \\ &=\; \frac{1}{2}\left(x(t)^T C^T Q C x(t) + u(t)^T R u(t)\right) \\ &=\; \frac{1}{2}\left(y(t)^T Q y(t) + u(t)^T R u(t)\right). \end{aligned}$$

while the governing differential equation is defined via the function

$$f(t,x,u) \equiv f(x,u) \equiv f(x(t),u(t)) = Ax(t) + Bu(t).$$

Next, we define the *Hamilton function* by

$$\mathcal{H}(x,u,\mu) \;=\; -\,g(x,u) + \mu(t)^T f(x,u),$$

where the components of the *co-state* $\mu(t) \in \mathbb{R}^n$ satisfy $\dot{\mu}_j(t) = -\dfrac{\partial \mathcal{H}}{\partial x_j}$ for $j = 1,\ldots,n$, which is in our case equivalent to

$$\dot{\mu}(t) \;=\; C^T Q C x(t) - A^T \mu(t). \tag{9}$$

From the *Pontryagin Maximum Principle* for *autonomous systems* as given, e.g., in [62, Theorem 4.3] or [54, Theorem V.3] (and many other references) applied to our problem, we obtain:

Proposition 2.2 *Let $u_*(t) \in \mathcal{PC}_m[0,t_f]$ and let x_* be the trajectory determined by $\dot{x}(t) = Ax(t)+Bu_*(t)$, $x(0) = x^0$. Then in order for u_* to be optimal, i.e., $\mathcal{J}(u_*) \leq \mathcal{J}(u)$ for all $u \in \mathcal{PC}_m[0,t_f]$, it is necessary that the following two conditions hold.*

(i) $\mathcal{H}(x,u_*,\mu) \geq \mathcal{H}(x,u,\mu)$ *on* $[0,t_f]$ *for all* $u \in \mathcal{PC}_m[0,t_f]$;

(ii) $\mu(t_f) = 0$

Condition (i) is called the *maximum condition* while (ii) is a *transversality condition*.

As u is not constraint, we obtain from Proposition 2.2(i) that $\dfrac{\partial \mathcal{H}}{\partial u_j} = 0$ for $j = 1, \ldots, m$, and hence it follows that

$$-Ru(t) + B^T \mu(t) \;=\; 0 \tag{10}$$

must hold on $[0, t_f]$ for an optimal control. Moreover, the second derivative test implies $R \geq 0$ as a necessary condition for the existence of an optimal control minimizing the objective functional $\mathcal{J}(u)$.

Collecting all equations, i.e., the state equations together with the initial conditions, (9) together with the transversality condition, and (10), we obtain

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), & x(0) &= x^0, \\ \dot{\mu}(t) &= C^T Q C x(t) - A^T \mu(t), & \mu(t_f) &= 0, \\ 0 &= Ru(t) - B^T \mu(t). \end{aligned}$$

These equations can be combined to the *two-point boundary value problem*

$$\begin{bmatrix} I_n & 0 & 0 \\ 0 & I_n & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\mu} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} A & 0 & B \\ C^T Q C & -A^T & 0 \\ 0 & -B^T & R \end{bmatrix} \begin{bmatrix} x \\ \mu \\ u \end{bmatrix}, \quad \begin{aligned} x(0) &= x^0, \\ \mu(t_f) &= 0. \end{aligned} \tag{11}$$

Note that $\dot{u}$ only appears formally, so that (11) does not pose additional smoothness properties for u. Actually, (11) is a boundary value problem for a *differential-algebraic equation* where the co-state μ and the control u are related by a purely algebraic equation. Assuming R nonsingular, u can be removed from the system, yielding an ordinary boundary value problem; see the next section.

Due to the special structure of the autonomous linear-quadratic optimal control problem, the conditions derived from the Pontryagin Maximum Principle yield necessary and sufficient conditions for existence of an optimal control. These are summarized in the following theorem which is contained in any textbook treating this class of problems (see, e.g., [5, 27] and many others).

Theorem 2.3 *a) If $u_* \in \mathcal{PC}_m[0, t_f]$ is an optimal control for the linear-quadratic optimization problem (1)–(3), then there exists a co-state $\mu(t) \in \mathbb{R}^n$ such that $\left[(x_*(t))^T, (u_*(t))^T, (\mu(t))^T\right]^T$ satisfies the two-point boundary value problem (11).*

b) If $\left[(x_*(t))^T, (u_*(t))^T, (\mu(t))^T\right]^T$ satisfies the two-point boundary value problem (11) and Q, R are positive semidefinite, then $\mathcal{J}(u_*) \leq \mathcal{J}(u)$ for all $u \in \mathcal{PC}_m[0, t_f]$ and for all (x, u) satisfying (2).

The above theorem yields conditions for the existence of a solution of the optimal control problem by transforming the constrained optimization problem to a boundary value problem. This problem can in principle be solved using any feasible (analytical or numerical) method. Next we will see that the solution of our problem can be obtained from an initial-value problem or even an algebraic equation (in case $t_f = \infty$) which is much easier to solve.

2.2 From two-point boundary value problems to Riccati equations

Assuming that R is nonsingular (i.e., together with $R \geq 0$ this implies that R is positive definite, denoted here by $R > 0$), (10) is equivalent to

$$u(t) \;=\; R^{-1}B^T\mu(t) \tag{12}$$

such that the state equations can be written as

$$\dot{x}(t) \;=\; Ax(t) + Bu(t) \;=\; Ax(t) + BR^{-1}B^T\mu(t). \tag{13}$$

Using (13), the two-point boundary value problem (11) can be re-written as

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\mu}(t) \end{bmatrix} \;=\; \begin{bmatrix} A & BR^{-1}B^T \\ C^TQC & -A^T \end{bmatrix} \begin{bmatrix} x(t) \\ \mu(t) \end{bmatrix}, \qquad \begin{matrix} x(0) = x^0, \\ \mu(t_f) = 0. \end{matrix} \tag{14}$$

Making the *ansatz* $\mu(t) := -X(t)x(t)$, the terminal condition for the co-state transforms to $\mu(t_f) = -X(t_f)x(t_f)$, which together with $\mu(t_f) = 0$, and the fact that $x(t_f)$ is unspecified implies $X(t_f) = 0$. Employing

$$\dot{\mu}(t) = -\dot{X}(t)x(t) - X(t)\dot{x}(t)$$

we obtain from the first differential equation in (14)

$$\dot{x}(t) \;=\; Ax(t) - BR^{-1}B^TX(t)x(t)$$

while the second yields

$$\begin{aligned} C^TQCx(t) + A^TX(t)x(t) \;&=\; -\dot{X}(t)x(t) - X(t)\dot{x}(t) \\ &=\; -\dot{X}(t)x(t) - X(t)(Ax(t) - BR^{-1}B^TX(t)x(t)). \end{aligned}$$

The latter equation is equivalent to

$$\left(\dot{X}(t) + X(t)A + A^TX(t) - X(t)BR^{-1}B^TX(t) + C^TQC\right)x(t) \;=\; 0$$

for all $t \in (0, t_f)$. Hence, as $x(t)$ is unspecified, we obtain the *Riccati matrix differential equation* (RDE)

$$\dot{X}(t) \;=\; -\left(C^TQC + X(t)A + A^TX(t) - X(t)BR^{-1}B^TX(t)\right), \tag{15}$$

i.e., an autonomous nonlinear matrix-valued differential equation. Together with $X(t_f) = 0$ this yields an initial value problem in reverse time. From the theory of (autonomous) Riccati differential equations it follows that there exists a unique solution $X_*(t)$, $t \in [0, t_f]$, of the RDE with terminal condition $X(t_f) = 0$ for any $t_f < \infty$; see, e.g., [66].

Transposing equation (15) we see that $X(t)^T$ has to satisfy the same differential equation as $X(t)$ on the whole interval $[0, t_f]$. From the uniqueness of the solution

of the initial value problem given by (15) together with the terminal condition it follows that $X_*(t) = X_*(t)^T$, i.e., the solution $X_*(t)$ is symmetric.

Under the given assumptions we obtain that the two-point boundary value problem (14) has a unique solution given by

$$\mu_*(t) \;=\; X_*(t)x_*(t), \qquad t \in [0, t_f],$$

where $x_*(t)$ is the unique solution of the linear initial value problem

$$\dot{x}(t) \;=\; (A - BR^{-1}B^T X_*(t))x(t), \qquad x(0) = x^0.$$

Summarizing all results, we obtain the following theorem.

Theorem 2.4 *If $Q \geq 0$, $R > 0$, and $t_f < \infty$, then there exists a unique solution of the linear-quadratic optimal control problem (1)–(3). The optimal control is given by the* feedback law

$$u_*(t) = -R^{-1}B^T X_*(t)x(t), \tag{16}$$

where $X_*(t)$ satisfies the RDE

$$\dot{X}(t) = -(C^T QC + X(t)A + A^T X(t) - X(t)BR^{-1}B^T X(t))$$

with terminal condition $X(t_f) = 0$. Moreover, for any initial value x^0 the optimal cost is

$$\mathcal{J}(u_*(\cdot)) = \frac{1}{2}(x^0)^T X_*(0)x^0.$$

The optimal control is therefore given as a *closed-loop control*, i.e., the system state is used to determine the input via the feedback law (16). The matrix $K_*(t) := R^{-1}B^T X_*(t)$ is called the optimal *gain matrix*.

Remark 2.5 *Note that $X(t)$ is* independent *of x^0.*

So far we have considered the case of a finite time horizon. We will now turn our attention to the infinite time case.

2.3 The infinite time case: $t_f = \infty$

In this section we will assume $R > 0$, $Q \geq 0$, that the matrix pair (A, B) is stabilizable, and that the matrix pair $(C^T QC, A)$ is detectable. Moreover, we will require $\mathcal{J}(u_*(\cdot)) < \infty$ where

$$\mathcal{J}(u(\cdot)) \;=\; \frac{1}{2}\int_0^\infty \left(x(t)^T C^T QCx(t) + u(t)^T Ru(t)\right)\, dt.$$

Under the above assumptions we can make the following observations:

$$\begin{aligned} \lim_{t\to\infty} u(t) \neq 0 \qquad &\Longrightarrow \quad \mathcal{J}(u(\cdot)) = \infty \\ \lim_{t\to\infty} x(t)^T C^T QCx(t) > 0 \qquad &\Longrightarrow \quad \mathcal{J}(u(\cdot)) = \infty \end{aligned}$$

From this, we obtain as necessary conditions

$$\lim_{t\to\infty} u(t) = 0 \tag{17}$$

$$\lim_{t\to\infty} x(t)^T C^T Q C x(t) = 0. \tag{18}$$

Condition (18) together with the detectability of (C^TQC, A) (see Definition 2.1) implies

$$\lim_{t\to\infty} x(t) = 0. \tag{19}$$

Now we examine the asymptotic behavior of the finite time solution $X_*(t)$ of the Riccati matrix differential equation derived in the last section. Define $\widetilde{X}(t, t_f) = X(t_f - t)$. Then $\widetilde{X}$ satisfies the differential equation

$$\dot{\widetilde{X}} = C^TQC + \widetilde{X}A + A^T\widetilde{X} - \widetilde{X}BR^{-1}B^T\widetilde{X} \tag{20}$$

with *initial condition* $\widetilde{X}(0, t_f) = X(t_f) = 0$ for any $t_f < \infty$.

Now fix t, and let t_f go to infinity. From the last section we know that $\widetilde{X}$ exists and is unique for any $t \in [0, t_f]$ and every $t_f < \infty$. Assume that $\lim_{t_f\to\infty} \widetilde{X}(t, t_f)$ is unbounded. This implies that $X(0) = \widetilde{X}(t_f, t_f)$ is unbounded for $t_f \to \infty$, and hence, $\mathcal{J}(u(\cdot)) = (x^0)^T X(0) x^0$ is unbounded for $t_f \to \infty$. Therefore, in order to obtain a finite optimal cost $\mathcal{J}(u)$, we require that $\lim_{t_f\to\infty} \widetilde{X}(t, t_f)$ exists. Denoting this limit by $X_\infty(t)$, observing that the boundedness of $\widetilde{X}(t, t_f)$ implies $\lim_{t_f\to\infty} \dot{\widetilde{X}}(t, t_f) = 0$, and taking limits in (20), we obtain the *algebraic Riccati equation* (subsequently denoted by ARE)

$$0 = \mathcal{R}(X_\infty) = C^TQC + X_\infty A + A^T X_\infty - X_\infty BR^{-1}B^T X_\infty. \tag{21}$$

As $X_\infty(t)$ has to satisfy the same equation for any $t \in [0, \infty)$, it is clear that the solution is time-invariant, i.e., $X_\infty(t) \equiv X_\infty$.

Remark 2.6 *The quadratic matrix equation (21) is often referred to as* continuous-time *ARE in order to distinguish it from the* discrete-time *ARE arising in discrete linear-quadratic optimization problems.*

In contrast to the RDE with terminal condition, the solution of the ARE is not unique. There may be infinitely many solutions; for a deeper insight into the sets of solutions of AREs see [48] and the references therein. We will see now that among those, we need a particular one. First of all, as X_∞ is the limit of the symmetric solutions $X_{t_f}(t)$ for $t_f \to \infty$ of the RDE, X_∞ has to be symmetric as well. Still, this constraint does not limit the number of solutions sufficiently.

Let us consider the *closed-loop system* resulting from applying the optimal control U_* given by

$$\dot{x}(t) = (A - BR^{-1}B^T X_*)x(t) =: A_* x(t), \qquad x(t_0) = x^0.$$

The unique solution of this initial-value problem clearly is $x_*(t) := e^{A_* t} x_0$. From (19) we know that $\lim_{t\to\infty} x_*(t) = 0$ which implies that $\lambda(A_*) \subset \mathbb{C}^-$ must hold. So we need to choose the particular solution of the ARE that makes A_* stable. This solution is called the *stabilizing solution*. Fortunately, under standard assumptions of control theory, this solution exists and is unique.

Theorem 2.7 *If $F \geq 0$, $G \geq 0$, (A, G) is stabilizable, (F, A) is detectable, then the ARE*

$$0 \;=\; F + A^T X + XA - XGX \tag{22}$$

has a unique, symmetric, stabilizing solution $X_* \geq 0$, i.e., $\lambda(A - GX_*) \subset \mathbb{C}^-$.

For a proof, see, e.g., [48] or many of the references given therein.

There exist many related theorems, relaxing some of the conditions given in the theorem in one or the other direction. Under very mild conditions it can be shown that if the stabilizing solution of the ARE exists, it is unique. Assuming in addition that (F, A) is observable, we also get that $X > 0$. Stabilizing solutions may also exist if any of the definiteness assumptions or detectability is removed; in that case, X_* may be indefinite. See [48] for the most complete account of the solution theory of AREs. In robust control, the existence of the stabilizing solution of the ARE is often related to the existence of a controller, satisfying some robustness criterion, via the so-called *bounded real lemma*; see, e.g., [30] and many other references.

For the linear-quadratic optimization problem considered here, Theorem 2.7 has the following consequence.

Theorem 2.8 *If $Q \geq 0$, $R > 0$, (A, B) is stabilizable, and $(C^T QC, A)$ is detectable, then the linear-quadratic optimal control problem (1)–(3) with $t_f = \infty$ has a unique solution given by*

$$u_*(t) \;=\; -R^{-1} B^T X_* x(t).$$

where X_* is the unique stabilizing solution of the ARE (22) with $F = C^T QC$, $G = BR^{-1}B^T$.

So far we have seen that the linear-quadratic optimal control/optimization problem under certain assumptions has a unique solution which can be found equivalently via solving a boundary value problem or an RDE/ARE, depending on whether $t_f < \infty$ or $t_f = \infty$. This implies three alternative approaches for the computational solution of the linear-quadratic optimization problem:

- Solve the control problem as *constrained optimization* problem. This requires to discretize the integral expression as well as the differential equation and to form a quadratic program which can then be solved by any method feasible for quadratic programming.
- Solve the two-point boundary value problem (14) by any numerical method feasible for linear boundary value problems.

- Solve the Riccati differential equation (15) if $t_f < \infty$, or the algebraic Riccati equation (22) if $t_f = \infty$.

The first two approaches will not be considered here any further. They both involve a discretization error, even if exact arithmetic could be used. Moreover, the discretized system will become very large for the quadratic programming approach, resulting in huge requirements concerning computational work as well as workspace. For the boundary value approach, boundary conditions are only given for half the variables at the left boundary while boundary conditions for the remaining variables are given at the right boundary. This causes problems for most existing methods as they rely on a full set of boundary conditions on both sides.

Therefore, computational methods used in control engineering applications are usually based on the Riccati approach. In order to solve the RDE, in principle any solver for initial value problems can be used. There also exist methods using the special structure of the problem; see, e.g., [28, 44, 47]. These methods are reliable, well tested, and usually very efficient.

In the following we will focus on the case $t_f = \infty$. In this case, the Riccati equation approach to the linear-quadratic optimization problems requires the numerical solution of an ARE as given in (22). We will see in the next section that this can be achieved by computing the *exact* solution (at least in exact arithmetic) without discretization errors, using purely linear algebraic methods, with a cost of order n^3. Hence, this approach outperforms the first two approaches as far as accuracy as well as computational cost is concerned, not even taking into account the problems caused by discretizing an infinite horizon problem.

The general outline of an algorithm for linear-quadratic optimal control with $t_f = \infty$, based on the Riccati approach, can be summarized as follows:

1. Form $F = C^T Q C$, $G = B R^{-1} B^T$.
2. Compute the stabilizing solution X_* of the ARE
$$0 = F + A^T X + X A - X G X.$$
3. Compute the gain matrix $K_* = R^{-1} B^T X_*$.
4. Define the optimal control $u_*(t) = -K_* x(t)$.
5. Compute the trajectory $x_*(t) = e^{(A - B K_*)t} x^0$.

Remark 2.9 *Several generalizations of the linear-quadratic optimal control problem (1)–(3) have been considered in the literature. The most straightforward extensions are a cross-weighting term of the form $y^T(t) S u(t)$ in the cost functional or an additional feed-through term in (3), i.e., the observed states are given by $y(t) = Cx(t) + Du(t)$. These generalized linear-quadratic optimization problems and some others, like the ones resulting from adding error integrators into the system, reference tracking, rejection of measurable disturbances, etc., can all be*

solved by the Riccati equation approach outlined above; for a nice overview of such generalizations see [69, Chapter 1].

A more complicated generalization is given by considering descriptor systems of the form $E\dot{x}(t) = Ax(t) + Bu(t)$ *instead of (2). In case* E *is nonsingular it is obvious that we get back to the standard case by premultiplying the above equation with* E^{-1} *— though reliable numerical methods will work with the generalized equation; see, e.g., [4, 58, 13]. The case of singular* E *matrices is more involved and treated in depth in [58]. But again, in the end the corresponding linear-quadratic optimization problem can be solved via an ARE of the form (6).*

In the next section we will derive some properties of algebraic Riccati equations that are essential with respect to the numerical methods used to solve them.

3 The Algebraic Riccati Equation

In this section we consider the general ARE

$$0 = \mathcal{R}(X) = F + A^T X + XA - XGX \tag{23}$$

where $A, G, F, X \in \mathbb{R}^{n\times n}$, G, F, X are symmetric, and X is the sought-after solution matrix, in particular we are looking for the stabilizing solution as defined in the last section.

First, we define the following $2n \times 2n$ matrix:

$$H = \begin{bmatrix} A & G \\ F & -A^T \end{bmatrix}. \tag{24}$$

Let the columns of $[U^T, V^T]^T$, $U, V \in \mathbb{R}^{n\times n}$, span an H-invariant, n–dimensional subspace, i.e.,

$$\begin{bmatrix} A & G \\ F & -A^T \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} Z, \qquad Z \in \mathbb{R}^{n\times n}, \quad \lambda(Z) \subset \lambda(H). \tag{25}$$

Assuming U nonsingular, we obtain from the first row of (25)

$$AU + GV = UZ \qquad \Longleftrightarrow \qquad U^{-1}AU + U^{-1}GV = Z. \tag{26}$$

Inserting this into the equation resulting from evaluating the second row in (25) yields

$$FU - A^T V = VZ = VU^{-1}AU + VU^{-1}GV.$$

The above equation is equivalent to

$$0 = F - A^T VU^{-1} - VU^{-1}A - VU^{-1}GVU^{-1}. \tag{27}$$

Setting

$$X := -VU^{-1} \tag{28}$$

we see that X solves (23). Hence, from an H-invariant subspace of dimension n, given as the range of $\begin{bmatrix} U \\ V \end{bmatrix}$ with U nonsingular, we obtain a solution of the ARE.

What remains is the problem of how to choose U, V such that U is nonsingular, VU^{-1} is symmetric, and X is stabilizing. Before we can solve this problem, we need some properties of the matrix H in (24).

Definition 3.1 *A matrix $H \in \mathbb{R}^{2n\times 2n}$ is a* Hamiltonian matrix *if it satisfies $(JH)^T = JH$, where*

$$J := \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}. \tag{29}$$

It is easy to see that any Hamiltonian matrix must have the block representation as shown in (24). On the other hand, it is easy to verify that the matrix H defined in (24) is Hamiltonian according to Definition 3.1.

Remark 3.2 *We may interpret the defining property of Hamiltonian matrices as being* skew self-adjoint *with respect to the indefinite inner product defined by J, i.e., $< x, y >_J := x^T Jy$ for all $x, y \in \mathbb{R}^{2n}$ and J as defined in (29). For more insight into the consequences of this property see [34, 48].*

We have the following important properties of the spectrum $\lambda(H)$ of Hamiltonian matrices.

Lemma 3.3 *Let $H \in \mathbb{R}^{2n\times 2n}$ be a Hamiltonian matrix as in (24).*

a) If $\lambda \in \lambda(H)$, then $-\lambda \in \lambda(H)$.

b) If $G \geq 0$, $F \geq 0$, (A, G) is stabilizable, and (F, A) is detectable, then H has no eigenvalues on the imaginary axis, i.e., $\lambda(H) \cap i\mathbb{R} = \emptyset$.

As a consequence of Lemma 3.3, we can write the spectrum of H as

$$\lambda(H) \;=\; \{\lambda_1, \ldots, \lambda_n\} \cup \{-\lambda_1, \ldots, -\lambda_n\} \;=:\; \Delta \cup (-\Delta), \tag{30}$$

where $\mathrm{Re}(\lambda_j) \leq 0$ for all $j = 1, \ldots, n$. Under the conditions of Lemma 3.3b), we have that $\Delta \subset \mathbb{C}^-$ and moreover, that there exists a nonsingular matrix $T \in \mathbb{R}^{2n\times 2n}$ such that

$$T^{-1}HT \;=\; \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix}, \qquad \lambda(H_{11}) = \Delta. \tag{31}$$

For instance, (31) can be computed via a *Schur decomposition* of H; see, e.g., [35].

The following theorem relates (31) to the required stabilizing solution of the ARE (23).

Theorem 3.4 *[46, 48, 63] If $G \geq 0$, $F \geq 0$, (A, G) is stabilizable, (F, A) is detectable, and*

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}, \qquad T_{ij} \in \mathbb{R}^{n\times n} \text{ for } i, j \in \{1, 2\},$$

is a nonsingular matrix such that (31) holds, then

a) T_{11} *is nonsingular,*

b) $T_{21}T_{11}^{-1}$ *is symmetric,*

c) $X = -T_{21}T_{11}^{-1}$ *is the unique stabilizing solution of the ARE (23).*

This theorem provides the basis for most of the numerical methods proposed to solve AREs of the form (6) — besides the methods treating (23) as the problem of finding a particular root of a nonlinear (here quadratic) function. Theorem 3.4 reduces the problem of solving an ARE and hence, the problem of solving linear-quadratic optimization problems arising in control theory, to a *Hamiltonian eigenproblem*, i.e., to the problem of finding an invariant subspace of the associated Hamiltonian matrix H corresponding to a particular subset of eigenvalues of H. This subset of $\lambda(H)$ is given by the *stable* eigenvalues, i.e., those with negative real parts.

The numerical methods for AREs of the form (23) can be distinguished into methods based on considering the ARE as a nonlinear equation and those solving the ARE via the corresponding Hamiltonian eigenproblem. For instance, the following methods have been considered.

- Methods based on the nonlinear equation approach:
  - Newton's method; see [45, 48],
  - Fletcher-Powell/Davidon's method; see [56],
  - secant method; see [29],
  - quasi-Newton methods; see [60],
  - conjugate gradient method; see [33],
  - exact line search method; see [11, 13].
- Methods based on the invariant subspace approach:
  - QR type algorithms; see, e.g., [49, 22, 1],
  - eigenvector methods based on [63],
  - Jacobi-type methods; see [26, 21],
  - a method based on a two-sided decomposition [17],
  - spectral projection methods; see, e.g., [67, 8, 25, 55, 11, 12].

Overviews over the available numerical methods, partially from more general points of view, can be found in [50, 41, 58, 61, 69].

In the next section, we will review some of the numerical methods from both categories, focusing on those that have turned out to be the most reliable ones. (Unfortunately, those are not necessarily the most frequently used ones.)

4 Numerical Methods

In order to solve AREs of the form (23) employing numerical methods, we have seen in the last section that this is possible using linear algebraic methods. In numerical linear algebra, the following requirements are often imposed on an algorithm to be considered as an appropriate method.

1. The algorithm should be *numerically backward stable*. For computing an invariant subspace of a matrix $M \in \mathbb{R}^{n\times n}$, this means that the *computed* approximation to a basis of this subspace, given by the columns of $S \in \mathbb{R}^{n\times r}$, has to satisfy

$$(M+E)S = SZ, \qquad \|E\| \leq const. \cdot \varepsilon \cdot \|M\|,$$

   where ε is the relative machine precision. That is, the columns of S span an invariant subspace of a matrix near to M. For more details, see, e.g., [35] and the references therein.

   In general, for algorithms based on similarity transformations, backward stability is only achieved if only orthogonal transformations are used.

2. The algorithm should be *numerically strong stable*. That is, the computed solution should be the solution of a slightly perturbed problem of the same structure. For the example given above, this means that M, E, and $M+E$ must have the same structure.

   For the Hamiltonian eigenproblem as discussed in the last section this implies that similarity transformations have to preserve the Hamiltonian structure.

3. The algorithm should have polynomial complexity, in particular, for eigenproblems, a complexity of $\mathcal{O}(n^3)$ *flops*[1] is required.

As methods based on the nonlinear equation approach have to be competitive with methods based on solving the Hamiltonian eigenproblem, the above requirements should also be satisfied (in some sense) for such algorithms. The above requirements will drive the discussion of numerical methods throughout this section.

4.1 Methods for the Hamiltonian Eigenproblem

First, we will consider methods based on solving the Hamiltonian eigenproblem. Recall that in order to solve the ARE it is sufficient to find a nonsingular matrix $T \in \mathbb{R}^{2n\times 2n}$ such that (31) is satisfied. We will assume throughout this section that this is possible. One set of assumptions ensuring this is given in Theorem 3.4. But note that these are only sufficient assumptions, T and the stabilizing solution may exist under much more general circumstances.

[1] *floating point operations*: each of the scalar operations "+", "−", "∗", "/", "$\sqrt{\ }$" is counted as a flop, following [35].

Most algorithms for solving matrix eigenproblems, i.e., for computing eigenvalues and -vectors or invariant subspaces of some matrix $M \in \mathbb{R}^{n\times n}$, are based on the following approach:

1. *Compute an initial transformation matrix $S_0 \in \mathbb{R}^{2n\times 2n}$ in order to reduce M to some condensed form, i.e., compute*

$$M_0 \;:=\; S_0^{-1} M S_0. \tag{32}$$

2. *Then construct a sequence of similarity transformations such that in each step*

$$M_{j+1} \;:=\; S_{j+1}^{-1} M_j S_{j+1}, \qquad j = 0, 1, 2, \ldots, \tag{33}$$

the reduced form is preserved and moreover, if we define $T_j := \prod_{k=0}^{j} S_k$, then $\lim_{j\to\infty} T_j = T$ and $\lim_{j\to\infty} M_j = M_*$ exist and eigenvalues and eigenvectors and/or M-invariant subspaces can be read off from M_* and T.

The purpose of the initial reduction to a condensed form and the preservation of this form throughout the iteration is twofold: first, such a reduction is usually necessary in order to satisfy the complexity requirements — an iteration step (33) on a reduced form can usually be implemented much cheaper than for a full matrix; second, using such a reduced form it is usually easier to track the progress of the iteration and detect if the problem can be decoupled (*deflation*) into smaller subproblems that can then be treated separately. For details see [35, Chapters 7–8].

Example 4.1 *The most widely used algorithm in numerical linear algebra following the above approach is the QR algorithm; see, e.g., [35] and the references given therein. In this algorithm, the initial reduction step consists of a reduction to upper Hessenberg form, i.e.,*

$$M_0 \;:=\; S_0^{-1} M S_0 \;=\; \begin{bmatrix} \times & \cdots & \times \\ \times & \ddots & \vdots \\ & \ddots & \times \\ & & \times \end{bmatrix}, \tag{34}$$

where S_0 is orthogonal such that $S_0^{-1} = S_0^T$. In each iteration, some rational function p_j is chosen and a QR decomposition $p_j(M_j) = S_{j+1} R_{j+1}$ is computed. The next iterate is then given by $M_{j+1} := S_{j+1}^T M_j S_{j+1}$. (Note that all S_j are orthogonal!) Often, p_j is chosen as a shift polynomial $p_j(t) = \prod_{k=1}^{\ell}(t - \mu_k)$ where the μ_k are some approximations to eigenvalues of M. In real implementations, the QR decomposition of $p_j(M_j)$ is only performed implicitly, thereby allowing an implementation of the QR iteration step in only $\mathcal{O}(n^2)$ flops.

In most practical circumstances (convergence can be proved under some assumptions), this iteration converges to real Schur form, *i.e., the limit M_* of the iterates is a quasi-upper triangular matrix having 1×1 and 2×2 blocks on the diagonal. The 1×1 blocks correspond to real eigenvalues while 2×2 blocks represent pairs of complex conjugate eigenvalues of M. Usually, convergence takes*

place in $\mathcal{O}(n)$ iterations, making the overall computational cost of this algorithm $\mathcal{O}(n^3)$. All transformation matrices S_j and therefore all T_j's and their limit T are orthogonal. This implies that the QR algorithm is numerically backward stable. Moreover, the first k columns of T form an M-invariant subspace corresponding to the eigenvalues of $M_*(1:k,1:k)$ (assuming that either $(m_*)_{k,k}$ defines a 1×1 block or $M_*(k-1:k,k-1:k)$ represents a 2×2 block in the real Schur form).

We will now see how this ideas can be used to design numerical methods for the Hamiltonian eigenproblem and hence for AREs.

4.1.1 The eigenvector approach

The first approach goes back to [63] and can be found in many control engineering textbooks. Suppose the Hamiltonian matrix H in (24) is diagonalizable and the conditions of Theorem 3.4 are satisfied. Then the first n columns of T in Theorem 3.4 can be chosen as the eigenvectors of H corresponding to Δ, i.e., the stable eigenvalues of H. For complex conjugate pairs of eigenvalues $\lambda, \overline{\lambda}$ with corresponding eigenvectors $x, \overline{x}$ consider the real "quasi-diagonalization" implied by

$$M\,[\mathrm{Re}(x)\ \mathrm{Im}(x)] \;=\; [\mathrm{Re}(x)\ \mathrm{Im}(x)] \begin{bmatrix} \mathrm{Re}(\lambda) & \mathrm{Im}(\lambda) \\ \mathrm{Im}(\lambda) & \mathrm{Re}(\lambda) \end{bmatrix}$$

in order to keep computations real. Denoting the columns of T by t_j, $j = 1,\ldots,2n$, where t_j is the eigenvector corresponding to the eigenvalue λ_j if λ_j is real and $t_j = \mathrm{Re}(x_j)$, $t_{j+1} = \mathrm{Im}(x_j)$ in case $\lambda_{j+1} = \overline{\lambda_j}$, and partitioning $t_j = [u_j^T,\, v_j^T]^T$, $u_j, v_j \in \mathbb{R}^n$, then by Theorem 3.4 the stabilizing solution of the ARE (23) is given by

$$X_* \;=\; -[v_1,\ldots,v_n][u_1,\ldots,u_n]^{-1}. \tag{35}$$

The eigenvectors of any matrix can be computed using the QR algorithm in different ways: accumulating the transformation matrices such that the matrix T is accessible after convergence, selected eigenvectors can be computed using T by solving for each eigenvalue a set of linear equations. But it is more common that if eigenvectors are desired, the QR algorithm is used to compute only the real Schur form without accumulating the transformations, thereby reducing the computational cost to almost one third. The so-obtained eigenvalue information can then be used to compute the desired eigenvectors by *inverse iterations*. (Note that for inverse iteration, the original matrix M need to be available or at least all the information (S_0 and M_0) needed to perform the initial reduction to Hessenberg form.) More details for both strategies can be found in [35, Section 7.6].

Hence, solving the ARE (23) can be done performing the following steps:

1. Form the Hamiltonian matrix H corresponding to the ARE.

2. Apply the QR iteration to H in order to obtain the eigenvalues of H without accumulating the transformations.

3. Compute the n eigenvectors of H corresponding to stable eigenvalues via inverse iteration.

4. Compute X_* via (35).

This algorithm requires approximately $160n^3$ flops.

At this point it should be emphasized that this algorithm can only be used safely if all eigenvalues of H are non-defective and well-separated from each other. In case H is close to a defective matrix, the eigenvector matrix of H will become ill-conditioned, usually causing a severe loss of accuracy in the computed eigenvectors and hence in the ARE solution X_*. Moreover, it becomes very difficult to decide on the geometric multiplicity of the eigenvalues in case they are close to defective eigenvalues. In case that there are defective eigenvalues, the above approach can also be used by employing the principal vectors of H corresponding to the stable defective eigenvalues. Due to roundoff errors during the QR iteration, it is usually hard to decide whether multiple eigenvalues have geometric multiplicity one or greater. This shows that in general, the above approach cannot be considered as being numerically stable and should be used only with very much care.

Moreover, the QR algorithm (already in the initial reduction part) destroys the Hamiltonian structure of H immediately and therefore does not satisfy the aim of a strong backward stable algorithm.

4.1.2 The Schur vector approach

The above considerations led Laub to suggest the following method [49]. Under the assumptions of Theorem 3.4 the stable eigenvalues are separated from the unstable ones. Hence, in the real Schur form M_* of the Hamiltonian matrix H computed by the QR algorithm, it is possible to reorder the eigenvalues in such a way that the stable eigenvalues appear in the leading $n \times n$ block of M_*. This can be achieved by a finite sequence of orthogonal similarity transformations; for details see [35] and the references therein. If these transformations are accumulated in an orthogonal matrix $\hat{T} \in \mathbb{R}^{2n \times 2n}$, we obtain an ordered real Schur form of M as

$$\hat{M} := \hat{T}^T M_* \hat{T} = \hat{T}^T T^T M T \hat{T} =: \begin{bmatrix} \hat{M}_{11} & \hat{M}_{12} \\ 0 & \hat{M}_{22} \end{bmatrix}, \tag{36}$$

where $\lambda(\hat{M}_{11}) = \Delta$. Hence, the first n columns of $T\hat{T}$ span the stable H-invariant subspace and the ARE solution can be computed via Theorem 3.4c).

As the columns of the orthogonal matrices T or $T\hat{T}$ transforming H to a real Schur form are called *Schur vectors*, this approach is called the *Schur vector method*. It can be summarized as follows.

1. Form the Hamiltonian matrix H corresponding to the ARE.

2. Apply the QR iteration to H in order to obtain the real Schur form of H and accumulate all similarity transformations into T.

3. Re-order the real Schur form of H as in (36).
4. Let the first n columns of $T\hat{T}$ be given by $[U^T, V^T]^T$, $U, V \in \mathbb{R}^{n\times n}$. Compute the stabilizing ARE solution as $X_* = -VU^{-1}$.

This algorithm requires about $205n^3$ flops (using flop counts from [35]) and is numerically backward stable in case the ARE is scaled such that $\|X_*\|_2 \approx 1$ [42]. Though $\|X_*\|_2$ is usually not known in advance, such a scaling can usually be achieved by heuristic methods; for a comparison of different strategies see [11].

But as the general QR algorithm is used, the Hamiltonian structure is destroyed during this algorithm and hence the method is not strongly stable. In case eigenvalues are close to the imaginary axis, this may cause failure of the method as eigenvalues may cross the imaginary axis due to roundoff errors and hence, the re-ordering in (36) becomes impossible.

Nevertheless, this method can be applied safely in most circumstances in linear-quadratic optimal control and has been a major progress in the course of introducing numerical reliable methods into control theory. But note that in the modern robust controller design methods, often AREs have to be solved where the corresponding Hamiltonian matrix has eigenvalues on or close to the imaginary axis. In these situations, the use of the Schur vector method is not advisable.

4.1.3 Symplectic methods

We have noted that in order to have a strong backward stable method, the Hamiltonian structure of H from (24) has to be preserved throughout the algorithm. It can be shown that in general, similarity transformations that preserve the Hamiltonian structure essentially have to be *symplectic* [20].

Definition 4.2 *A matrix $S \in \mathbb{R}^{2n\times 2n}$ is* symplectic *iff $SJS^T = J$ (or, equivalently, $S^TJS = J$), where J is defined in (29).*

Proposition 4.3 *If $H \in \mathbb{R}^{2n\times 2n}$ is Hamiltonian and $S \in \mathbb{R}^{2n\times 2n}$ is symplectic, then $S^{-1}HS$ is Hamiltonian.*

Some remarks are in order.

Remark 4.4 *a) Symplectic matrices form a* Lie group *which operates on the corresponding* Lie algebra *of Hamiltonian matrices.*

b) If the Hamiltonian matrix H has additional structure, then other similarity transformations may preserve this structure as well. For instance, if $H = \begin{bmatrix} 0 & G \\ F & 0 \end{bmatrix}$, then any similarity transformation with $U = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}$, where $U_1, U_2 \in \mathbb{R}^{n\times n}$ are both orthogonal, preserves the Hamiltonian structure.

c) From Definition 4.2 it is straightforward to verify $S^{-1} = -JS^TJ$ and $\det S = 1$ *if S is symplectic. Unfortunately, the norm of a symplectic matrix is not bounded so that for symplectic similarity transformations, usually only the following backward error bound can be given: if $\tilde{H}$ is the computed result of a similarity transformation of a Hamiltonian matrix H by a symplectic matrix S, then*

there exists a Hamiltonian matrix $E \in \mathbb{R}^{2n\times 2n}$ *such that* $S^{-1}(H+E)S = \tilde{H}$ *and* $\|E\| \leq const. \cdot \varepsilon \cdot \|H\| \cdot \|S\| \cdot \|S^T\|$. *From this it is clear that backward stability can not be guaranteed only by using symplectic similarity transformations.*

If an algorithm preserves the Hamiltonian structure, the spectral information and in particular the symmetry of the eigenvalues with respect to the imaginary axis is preserved. Hence, the problems arising in the re-ordering step of the Schur vector method are avoided. Moreover, structure-preserving algorithms can usually be implemented using fewer arithmetic operations and work space than standard algorithms for general non-symmetric matrices. Hamiltonian matrices are determined by $2n^2+n$ parameters rather than the $4n^2$ parameters of a general $2n \times 2n$ matrix. Only these parameters have to be stored and re-computed during each transformation of the Hamiltonian matrix. (Note that this does not necessarily lead to shorter execution times as it requires some overhead with respect to index calculations/access of matrix elements. Moreover, block algorithms as used in most modern numerical linear algebra applications can not be used as efficiently as for general matrices.)

In order to have a strong backward stable method, we have observed in Remark 4.4c) that requiring similarity transformations to be symplectic is not sufficient. They also ought to be orthogonal. It is easy to see that such matrices also have a special structure.

Lemma 4.5 *[59] If* $U \in \mathbb{R}^{2n\times 2n}$ *is orthogonal and symplectic, then*

$$U = \begin{bmatrix} U_1 & U_2 \\ -U_2 & U_1 \end{bmatrix}, \qquad U_1, U_2 \in \mathbb{R}^{n\times n}. \tag{37}$$

Moreover, as the intersection of two groups, orthogonal symplectic matrices form a group $\mathcal{US}_{2n}$ *with respect to matrix multiplication.*

As matrices in $\mathcal{US}_{2n}$ are determined by the $2n^2$ parameters given by the entries of U_1, U_2, only these parameters need to be stored and updated throughout a sequence of similarity transformations.

Now, the following theorem raises the hope that it is possible to find an algorithm based on symplectic *and* orthogonal similarity transformations for solving AREs. Together with an appropriate scaling (due to the $\|X_*\| \approx 1$ request), such an algorithm would be strong backward stable.

Theorem 4.6 *[59] If* H *is Hamiltonian and* $\lambda(H) \cap \imath\mathbb{R} = \emptyset$ *then there exists* $U \in \mathcal{US}_{2n}$ *such that*

$$U^T H U = \begin{bmatrix} \hat{H}_{11} & \hat{H}_{12} \\ 0 & -\hat{H}_{11}^T \end{bmatrix}, \qquad \hat{H}_{11}, \hat{H}_{12} \in \mathbb{R}^{n\times n}, \tag{38}$$

where $\hat{H}_{11}$ *is in real Schur form and* $\lambda(\hat{H}_{11}) = \Delta$ *(the stable part of* $\lambda(H)$*).*

Partitioning U from (38) as in (37), we have from Theorem 3.4 that the stabilizing solution of the ARE (23) is given by $X_* = U_2 U_1^{-1}$.

Remark 4.7 *The decomposition given in (38) is called the* Hamiltonian Schur form. *It can be shown that such a form may also exist if eigenvalues on the imaginary axis are present. They have to satisfy certain properties, the most obvious one is that their algebraic multiplicity needs to be even; see [52, 53].*

The problem of computing the Hamiltonian Schur form (38) using only $\mathcal{O}(n^3)$ flops is known in numerical linear algebra as *Van Loan's curse* — indicating that it is a nontrivial task. Though much progress has been made throughout the last years towards such an algorithm, a completely satisfactory method has not yet been found. In the remainder of this section we will highlight some of these developments.

The Hamiltonian QR algorithm

As the QR algorithm is considered to be the best method for solving the dense non-symmetric eigenproblem, it is straightforward to strive for a symplectic variant of the QR algorithm converging to the Hamiltonian Schur form given in (38). A framework for such an algorithm can easily be derived analogous to Example 4.1. Denote the iterates of such an algorithm by H_j. If we choose the QR decomposition performed in each step, i.e., $p_j(H_j) = S_{j+1}R_{j+1}$, such that all S_{j+1} are symplectic and orthogonal, then from Proposition 4.3 it follows that all iterates $H_{j+1} = S_{j+1}^T H_j S_{j+1}$ are Hamiltonian. Unfortunately, such a *symplectic QR decomposition* does not always exist. Sets of matrices in $\mathbb{R}^{2n\times 2n}$ for which it exists are described in [20]. In particular, it is also shown there (see [24] for a constructive proof) that if M is symplectic, then there exists $S \in \mathcal{US}_{2n}$ such that

$$M = SR = S\begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{11}^{-T} \end{bmatrix} = \begin{bmatrix} \ast & \ast \\ & \ast \end{bmatrix}, \tag{39}$$

where $R_{11}, R_{12} \in \mathbb{R}^{n\times n}$. Uniqueness of this decomposition can be achieved by requiring all diagonal entries of R_{11} to be positive.

As the matrix R in (39) is permutationally similar to an upper triangular matrix and the Hamiltonian Schur form is similar to the real Schur form using the same permutations, it can be shown under mild assumptions that such a Hamiltonian QR algorithm converges to Hamiltonian Schur form if it exists. Moreover, as only similarity transformations in $\mathcal{US}_{2n}$ are used, the algorithm can be shown to be strong backward stable.

Byers [24] shows that if the rational function p_j is chosen to be the *Cayley shift* $c_k(t) := (t-\mu_k)(t+\mu_k)^{-1}$, where μ_k is an approximate real eigenvalue of H, or $d_k(t) := (t-\mu_k)(t-\overline{\mu_k})(t+\mu_k)^{-1}(t+\overline{\mu_k})^{-1}$, where μ_k is an approximate complex eigenvalue of H, then $p_j(H_j)$ is symplectic, and hence, the symplectic QR decomposition of $p_j(H_j)$ exists. In case $\pm\mu_k$ are exact eigenvalues of H and hence of H_j, then deflation is possible, and we can proceed with the deflated problem of smaller dimension without ever being forced to invert a singular matrix.

Unfortunately, the so derived algorithm is of complexity $\mathcal{O}(n^4)$ as each symplectic QR decomposition requires $\mathcal{O}(n^3)$ flops and usually $\mathcal{O}(n)$ iterations are required (based on the experience that for each eigenvalue, 1–2 iterations are needed). The missing part that would bring the computational cost down to $\mathcal{O}(n^3)$ is an initial reduction analogous to the Hessenberg reduction in the QR algorithm that

- is invariant under the similarity transformation performed in each step of the Hamiltonian QR algorithm (the *Hamiltonian QR step*);
- admits an implementation of the Hamiltonian QR step using only $\mathcal{O}(n^2)$ flops.

In [24] Byers shows that such a form exists.

Definition 4.8 *A Hamiltonian matrix $H \in \mathbb{R}^{2n\times 2n}$ is in* Hamiltonian Hessenberg form *if*

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & -H_{11}^T \end{bmatrix} = \begin{bmatrix} \diagdown & \square \\ * & \diagdown \end{bmatrix}, \tag{40}$$

where $H_{ij} \in \mathbb{R}^{n\times n}$, $i,j = 1,2$, H_{11} is upper Hessenberg, and $H_{21} = \varphi e_n e_n^T$ with $\varphi \in \mathbb{R}$ and e_n being the nth unit vector. The Hamiltonian Hessenberg matrix H is unreduced if $h_{i+1,i} \neq 0$, $i = 1,\ldots,n-1$, and $\varphi \neq 0$.

Byers [24] shows that if H_j is in Hamiltonian Hessenberg form and the rational function p_j is chosen as a Cayley shift, then H_{j+1} is in Hamiltonian Hessenberg form again and the Hamiltonian QR step can be implemented in $\mathcal{O}(n^2)$ flops.

The crux of this algorithm is the initial reduction of a Hamiltonian matrix to Hamiltonian Hessenberg form. Byers shows how this can be achieved if one of the off-diagonal blocks of the Hamiltonian matrix H in (24) has rank 1. (This is related to control systems of the form (2)–(3) having only one input, i.e., *single input systems* and/or only one output, i.e., *single output systems.*) But unfortunately no algorithm is known for reducing a general Hamiltonian matrix to Hamiltonian Hessenberg form. But the situation is even worse. Analogous to the standard QR algorithm where the QR step is performed on unreduced Hessenberg matrices (possibly zeros on the subdiagonal are used for deflation, i.e., splitting the problem in two or more subproblems consisting of unreduced Hessenberg matrices), the Hamiltonian QR algorithm works for unreduced Hamiltonian Hessenberg matrices. The following theorem due to Ammar and Mehrmann [2] shows that the situation is in general hopeless with respect to the existence of the unreduced Hamiltonian Hessenberg form.

Theorem 4.9 *If $H \in \mathbb{R}^{2n\times 2n}$ is Hamiltonian, then there exists an orthogonal and symplectic matrix transforming H to unreduced Hamiltonian Hessenberg form if and only if the nonlinear set of equations*

$$x^T x = 1 \qquad \text{and} \qquad x^T J H^{2k-1} x = 0 \quad \text{for} \quad k = 1,\ldots,n-1,$$

has a solution that is not *contained in an H-invariant subspace of dimension n or less.*

Obviously, if JH is positive definite, such a vector cannot exist, showing that there really exist situations in which the unreduced Hamiltonian Hessenberg form does not exist. Therefore, other approaches have been investigated during the last decade.

The Hamiltonian SR algorithm

As observed in the last sections, a QR-type algorithm satisfying all three requests given at the beginning of Section 4 could so far only be given for special cases of Hamiltonian matrices. In the Schur vector approach, strong stability was given up due to the use of non-symplectic similarity transformations. The general Hamiltonian QR algorithm can in general not be implemented in $\mathcal{O}(n^3)$ flops. A third approach is to force symplecticity and efficient implementation by giving up orthogonality of the similarity transformations. This was pursued in [22]. The algorithm presented there follows the framework outlined in (32)–(33). All similarity transformations are chosen to be symplectic. Non-orthogonal *symplectic Gaussian transformations* (see, e.g., [22, 58]) are allowed. Using these transformations, an initial reduction to a very condensed form, consisting of only $4n-1$ nonzero parameters can be computed.

Definition 4.10 *A Hamiltonian matrix $H \in \mathbb{R}^{2n\times 2n}$ is* J-tridiagonal *iff it has the form*

$$H = \left[\begin{array}{cccc|cccc} a_{11} & & & & g_{11} & g_{21} & & \\ & a_{22} & & & g_{21} & g_{22} & \ddots & \\ & & \ddots & & & \ddots & \ddots & g_{n,n-1} \\ & & & a_{n,n} & & & g_{n,n-1} & g_{n,n} \\ \hline f_{11} & & & & -a_{11} & & & \\ & f_{22} & & & & -a_{22} & & \\ & & \ddots & & & & \ddots & \\ & & & f_{n,n} & & & & -a_{n,n} \end{array}\right].$$

A Hamiltonian matrix of the above form is also said to be in *Hamiltonian J-Hessenberg form.*

It is shown in [20] that for any Hamiltonian matrix and almost any vector $s_1 \in \mathbb{R}^{2n\times 2n}$, a symplectic matrix with first column s_1 exists such that $S^{-1}HS$ is J-tridiagonal. In case it does not exist, one may perform a random similarity transformation on H and try again to compute the transformation to this reduced form. As this transformation exists almost always, this will not lead to an infinite process and the reduction to J-tridiagonal form can serve as the initial reduction step in the general framework given by (32)–(33).

For the iteration step, we need the decomposition of $p_j(H_j)$, where p_j is again some rational function, into a symplectic matrix S_{j+1} and a matrix R_{j+1} such that $H_{j+1} = S_{j+1}^{-1} H_j S_{j+1}$ is again J-tridiagonal.

Definition 4.11 *Let $M \in \mathbb{R}^{2n\times 2n}$. A decomposition $M = SR$ is called an SR* decomposition *of M iff $S \in \mathbb{R}^{2n\times 2n}$ is symplectic and R is J-*triangular*, i.e.,*

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} \diagdown & \diagdown \\ 0 & \diagdown \end{bmatrix},$$

where $R_{ij} \in \mathbb{R}^{n\times n}$, $i,j = 1,2$, are upper triangular and the main diagonal of R_{21} is zero.

It can be shown that the set of matrices having an SR decomposition is dense in $\mathbb{R}^{2n\times 2n}$ (but not in $\mathbb{C}^{2n\times 2n}$!) and hence, this decomposition can be used to derive an SR step analogous to the QR step given in Example 4.1. In order to circumvent breakdowns due to the non-existence of an SR decomposition or ill-conditioned transformation matrices when being close to such a situation, an exceptional shift strategy has been proposed in [22].

The non-orthogonal Gaussian transformations used in both the initial reduction to J-tridiagonal form and during the SR decomposition in each SR step are chosen to be optimally conditioned in the class of transformations that satisfy the same purpose. Nevertheless, ill-conditioned transformations can not always be avoided. Hence, the condition should be monitored during the iteration (this can easily be achieved; see [22]). But still the method is not backward stable due to the use of non-orthogonal transformations.

The choice of rational functions driving the SR step is guided by the symmetry of the spectrum of Hamiltonian matrices. So usually, a double shift $p_j(t) = (t - \mu_j)(t + \mu_j)$ for real approximate eigenvalues $\pm\mu_j$ or a quadruple shift $p_j(t) = (t - \mu_j)(t - \overline{\mu_j})(t + \mu_j)(t + \overline{\mu_j})$ for complex approximate eigenvalues $\pm\mu_j, \pm\overline{\mu_j}$ is chosen. Keeping p_j real is not only an issue of implementation but is also due to the fact that the class of complex matrices having no SR decomposition is not dense in $\mathbb{C}^{2n\times 2n}$. Exceptional steps can be derived using the SR decomposition of $p_j(H_j) = H_j - \alpha_j I_{2n}$ where α_j is some randomly chosen real scalar.

It can be shown that if $p_j(H_j) = S_{j+1}R_{j+1}$ is an SR decomposition, then $H_{j+1} = S_{j+1}^{-1} H_j S_{j+1}$ is again J-tridiagonal.

In summary, the Hamiltonian SR algorithm consists of an initial reduction to J-tridiagonal form and SR steps derived from the SR decomposition of double and quadruple shift polynomials $p_j(H_j)$. The SR steps can be implemented implicitly analogous to the implicit QR step at a computational cost of $\mathcal{O}(n)$ flops. This SR

iteration converges to a Hamiltonian matrix

$$H_\infty = \left[\begin{array}{ccc|ccc} A_{11} & & & G_{11} & & \\ & \ddots & & & \ddots & \\ & & A_{rr} & & & G_{rr} \\ \hline F_{11} & & & -A_{11} & & \\ & \ddots & & & \ddots & \\ & & F_{rr} & & & -A_{rr} \end{array}\right]$$

where $n/2 \leq r \leq n$ and all A_{jj}, G_{jj}, F_{jj}, $j = 1, \ldots, r$, are either 1×1 or 2×2. Moreover, $\lambda(H) = \cup_{j=1}^{r} \lambda(H_{jj})$, where $H_{jj} = \left[\begin{smallmatrix} A_{jj} & G_{jj} \\ F_{jj} & -A_{jj}^T \end{smallmatrix}\right]$, $j = 1, \ldots, r$, are Hamiltonian submatrices of H_∞. These submatrices are then transformed to real Hamiltonian Schur form (if it exists). Also note that convergence of the SR algorithm is usually *cubic* [71]. Assuming all H_{jj} are transformed to Hamiltonian Schur form, these transformations can be combined to yield a symplectic similarity transformation such that H_∞ is transformed to Hamiltonian Schur form and the leading $n \times n$ principal submatrix corresponds to the stable part of $\lambda(H)$.

Due to its potential numerical instability, this algorithm is only used in certain circumstances (e.g., in the context of Ritz approximations to $\lambda(H)$ obtained from a symplectic Lanczos process; see [14]) or to compute an initial estimate of an ARE solution that is refined, e.g., by Newton's method; see Section 4.3.

For the details of the method and its implementation see [22, 58, 69].

The multishift algorithm

From Theorem 4.9 we know that the reduction to Hamiltonian Hessenberg form which is necessary to efficiently implement the Hamiltonian QR algorithm is in general not possible. What can be achieved by orthogonal symplectic similarity transformations is the following reduction due to Paige and Van Loan [59].

Theorem 4.12 *Let $H \in \mathbb{R}^{2n\times2n}$. Then there exists $U \in \mathcal{US}_{2n}$ such that*

$$U^T H U = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} = \begin{bmatrix} \diagdown & \square \\ \diagdown & \square \end{bmatrix}, \tag{41}$$

where $H_{11} \in \mathbb{R}^{n\times n}$ is upper Hessenberg and $H_{21} \in \mathbb{R}^{n\times n}$ is upper triangular. The transformation matrix U can be chosen such that

$$U = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & \tilde{U}_1 & 0 & \tilde{U}_2 \\ \hline 0 & 0 & 1 & 0 \\ 0 & -\tilde{U}_2 & 0 & U_1 \end{array}\right]. \tag{42}$$

If in addition, H is Hamiltonian, then

$$U^T H U = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & -H_{11}^T \end{bmatrix} = \begin{bmatrix} \diagdown\!\square & \square \\ \diagdown & \diagdown\!\square \end{bmatrix}, \tag{43}$$

i.e., H_{21} is diagonal and H_{12} is symmetric.

The reduced form (43) of a Hamiltonian matrix will be called *PVL form* in the following. An algorithm for computing the transformation given in (43) is derived in [59]. It can be implemented using a finite number of similarity transformations and requires $\mathcal{O}(n^3)$ flops. Unfortunately, the PVL form is not preserved under the Hamiltonian QR iteration and can therefore not serve for the initial reduction step of the Hamiltonian QR algorithm. In the following, we will see that the PVL form can be used in what can be considered as a Hamiltonian *multishift* QR algorithm.

First, we need some more theory.

Definition 4.13 *A real subspace $\mathcal{Q} \subset \mathbb{R}^{2n}$ is* isotropic *iff $x^T J y = 0$ for all $x, y \in \mathcal{Q}$ and J as in (29). If $\mathcal{Q}$ is maximal, i.e., not contained in an isotropic subspace of larger dimension, then $\mathcal{Q}$ is a* Lagrangian *subspace.*

Lemma 4.14 *Let $S \in \mathbb{R}^{2n \times 2n}$ be symplectic. Then the first r columns of S, $1 \leq r \leq n$, span an isotropic subspace of $\mathbb{R}^{2n}$. For $r = n$, this subspace is Lagrangian.*

Proof: This property immediately follows from the definition of symplectic matrices, i.e., $S^T J S = J$. The case $r = n$ is a consequence of the maximality of Lagrangian subspaces. □

The basis for the multishift algorithm is contained in the following result.

Proposition 4.15 *[2] Let $H \in \mathbb{R}^{2n \times 2n}$ be Hamiltonian with $\lambda(H) = \Delta_n \cup (-\Delta_n)$, $\Delta_n \cap (-\Delta_n) = \emptyset$, and $\Delta_n = \overline{\Delta_n} = \{\lambda_1, \ldots, \lambda_n\}$. Then the* multishift vector

$$x = \alpha (H - \lambda_1 I_{2n}) \cdots (H - \lambda_n I_{2n}) e_1, \quad \alpha \in \mathbb{R}, \tag{44}$$

where $e_1 \in \mathbb{R}^{2n}$ is the first unit vector, is contained in the n-dimensional H-invariant subspace corresponding to $-\Delta_n$. Moreover, this subspace is Lagrangian. In particular, if $\Delta_n \subset \mathbb{C}^+ := \{z \in \mathbb{C} \mid Re(z) > 0\}$, then this Lagrangian subspace is the stable H-invariant subspace.

So once we know the spectrum of H we can compute *one* vector that is contained in the subspace required for solving the corresponding ARE. This observation can be combined with the computation of the PVL form in order to derive a *multishift* step as follows — assuming for simplicity that H has no eigenvalues on the imaginary axis.

Algorithm 4.16 *[Multishift step]*

1. Compute the multishift vector as in (44) with $\lambda_j \in \mathbb{C}^+$, $j = 1, \ldots, n$. Choose α in (44) such that $||x||_2 = 1$. (If this is not possible, i.e., $x = 0$, then exit.)
2. Compute $U_1 \in \mathcal{US}_{2n}$ such that $U_1^T x = \pm e_1$.
3. Set $H_1 = U_1^T H U_1$.
4. Compute the PVL form of H_1, i.e., compute $U_2 \in \mathcal{US}_{2n}$ such that $H_2 = U_2^T H_1 U_2 = (U_1 U_2)^T H (U_1 U_2)$ is in PVL form.

Using this approach, it is possible to get the whole stable H-invariant subspace. The following theorem will indicate how Algorithm 4.16 can be used to achieve this.

Theorem 4.17 *Let $H \in \mathbb{R}^{2n \times 2n}$ be Hamiltonian and let $\mathcal{V}_n$ be an n-dimensional H-invariant Lagrangian subspace corresponding to $\Delta_n \subset \lambda(H)$ with Δ_n as in Proposition 4.15. Further, let the multishift vector x from (44) be computed using $-\Delta_n = \{\lambda_1, \ldots, \lambda_n\}$ as the shifts. If $1 \leq p \leq n$ is the dimension of the minimal isotropic H-invariant subspace $\mathcal{V}_p$ containing x, then after Step 4 of the multishift step, H_2 has the form*

$$H_2 = \left[\begin{array}{cc|cc} A_{11} & A_{12} & G_{11} & G_{21} \\ 0 & A_{22} & G_{21}^T & G_{22} \\ \hline 0 & 0 & -A_{11}^T & 0 \\ 0 & F_{22} & -A_{12}^T & -A_{22}^T \end{array}\right] \begin{array}{l} \}p \\ \}n-p \\ \}p \\ \}n-p \end{array}, \tag{45}$$

where $A_{11} \in \mathbb{R}^{p \times p}$, $\lambda(A_{11}) \subset \Delta_n$, and the Hamiltonian submatrix

$$H_{22} := \left[\begin{array}{cc} A_{22} & G_{22} \\ F_{22} & -A_{22}^T \end{array}\right] \in \mathbb{R}^{2(n-p) \times 2(n-p)}$$

is in PVL form.

Furthermore, for $U_1, U_2 \in \mathcal{US}_{2n}$ from the multishift step we have

$$U := U_1 U_2 = [u_1, \ldots, u_p, u_{p+1}, \ldots, u_{2n}] \in \mathcal{US}_{2n}, \quad u_j \in \mathbb{R}^{2n} \text{ for } j = 1, \ldots, 2n,$$

and $\operatorname{span}\{u_1, \ldots, u_p\} = \mathcal{V}_p \subset \mathcal{V}_n$.

We will provide the proof of the above theorem for the sake of completeness as it is not given in the open literature.

Proof: If $x = 0$, then $p = 0$ and nothing needs to be shown. Therefore, w.l.o.g. we assume $x \neq 0$.

Let $U_j = [u_1^{(j)}, \ldots, u_{2n}^{(j)}]$ and $H_j = [h_{ik}^{(j)}]_{i,k=1}^{2n}$ for $j = 1, 2$.

First of all, note that $\mathcal{V}_p \subset \mathcal{V}_n$ as $x \in \mathcal{V}_n$ by Proposition 4.15.

From (37) it is clear that $u_{n+k}^{(j)} = J^T u_k^{(j)}$ for $j = 1, 2$ and $k = 1, \ldots, n$. As $\mathcal{V}_n$ is Lagrangian and hence isotropic, we have that $x^T J y = 0$ for all $y \in \mathcal{V}_n$. Moreover, $Hx \in \mathcal{V}_n$ as $\mathcal{V}_n$ is H-invariant. Hence, $x^T J H x = 0$.

The proof will be given by induction on the columns of H_2. From (42) and Step 2 of the multishift step we have

$$\begin{aligned} h^{(2)}_{n+1,1} &= e^T_{n+1}U_2^T H_1 U_2 e_1 = e^T_{n+1}U_1^T H U_1 e_1 = \pm (u^{(1)}_{n+1})^T Hx \\ &= \pm (u^{(1)}_1)^T JHx = \pm e_1^T U_1^T JHx = x^T JHx = 0. \end{aligned}$$

Partition $U_1U_2 = [u_1, \ldots, u_{2n}]$ and define $\mathcal{U}_q := \operatorname{span}\{u_1, \ldots, u_q\}$, $1 \le q \le n$. After Step 4 of the multishift step we have

$$Hu_1 = h^{(2)}_{11}u_1 + h^{(2)}_{21}u_2.$$

If $h^{(2)}_{21} = 0$, then $\mathcal{U}_1 = \operatorname{span}\{u_1\}$ is a 1–dimensional H-invariant subspace. This subspace contains x as from (42) it follows that $u_1 = u^{(1)}_1 = \pm x$. Furthermore, $\mathcal{U}_1$ is isotropic because of Lemma 4.14. Hence the assertion holds with $p = 1$ and $\mathcal{V}_1 = \mathcal{U}_1$.

Now assume $h^{(2)}_{21} \neq 0$. Then $u_2 \in \operatorname{span}\{u_1, Hu_1\} = \operatorname{span}\{x, Hx\} \subset \mathcal{V}_p$. As u_1 and u_2 are linear independent and $\mathcal{U}_2 \subset \mathcal{V}_p$, it follows that $p > 1$.

Suppose $h^{(2)}_{n+j-1,j-1} = 0$, $h^{(2)}_{j,j-1} \neq 0$, and $U_j \subset \mathcal{V}_p$ for $2 \le j \le n$.

First, assume $j < n$ and consider $HUe_j = UH_2e_j$. This identity can be written as

$$h^{(2)}_{j+1,j}u_{j+1} + h^{(2)}_{n+j,j}u_{n+j} = (H - h^{(2)}_{j,j}I_{2n})u_j - \sum_{i=1}^{j-1} h^{(2)}_{ij}u_i. \tag{46}$$

The right-hand side of (46) is contained in $\mathcal{V}_p$ as $\mathcal{U}_j \subset \mathcal{V}_p$ and $\mathcal{V}_p$ is H-invariant. As U is orthogonal and symplectic and $\mathcal{V}_p$ is isotropic, we can multiply (46) from the left by $u_j^T J$ in order to obtain $h^{(2)}_{n+j,j} = 0$.

If $h^{(2)}_{j+1,j} = 0$, then $\mathcal{U}_j$ is an j-dimensional, isotropic H-invariant subspace. As $x \in \mathcal{U}_j \subset \mathcal{V}_p$ and $\mathcal{V}_p$ is the minimal isotropic H-invariant subspace containing x, it follows that $\mathcal{V}_p = \mathcal{U}_j$ and $p = j$.

On the other hand, if $h^{(2)}_{j+1,j} \neq 0$, then $u_{j+1} \in \mathcal{V}_p$. Hence we obtain $\mathcal{U}_{j+1} \subset \mathcal{V}_p$ and $p \ge j + 1$. We then continue until j is the maximal index $j < n$ such that $h^{(2)}_{n+j-1,j-1} = 0$, $h^{(2)}_{j,j-1} \neq 0$, and $U_j \subset \mathcal{V}_p$. By the above considerations it then follows that $p = j$ and $\mathcal{V}_p = \mathcal{U}_j$.

In case $j = n$, the identity $HUe_j = UH_2e_j$ yields

$$h^{(2)}_{2n,n}u_{2n} = (H - h^{(2)}_{n,n}I_{2n})u_n - \sum_{i=1}^{n-1} h^{(2)}_{i,n}u_i.$$

Multiplying from the left by $u_n^T J$ and noting that the right-hand side is contained in $\mathcal{V}_p$, we get $h^{(2)}_{2n,n} = 0$. Hence, $\mathcal{U}_n \subset \mathcal{V}_p$ is an n-dimensional, isotropic H-invariant subspace containing x. Therefore, the assertion follows as above with $p = n$.

The fact that H_{22} in in PVL form directly follows from the PVL reduction in Step 4 of the multishift step. □

The theorem shows that if the multishift vector x from (44) has components in *all* directions of a Lagrangian H-invariant subspace, then after one multishift step, a basis for this invariant subspace is given by the first n columns of U_1U_2. Otherwise, the first p columns of U_1U_2 span a p-dimensional H-invariant subspace contained in this subspace and the problem decouples into two subproblems. Algorithm 4.16 can then repeatedly be applied to the resulting Hamiltonian submatrix $H_{22} \in \mathbb{R}^{2(n-p)\times 2(n-p)}$ until $p = n$. The implementation of this algorithm is described in detail in [1].

Remark 4.18 *The proof of Theorem 4.17 has not exploited the orthogonality of U but only its symplecticity. Hence, the proof as given above also applies to any multishift step using (non-orthogonal) symplectic similarity transformations that yield the same reductions needed in the multishift step as described in [65].*

As only orthogonal symplectic similarity transformations are used, a multishift step is strongly backward stable. The computational cost of one multishift step for $p = 0$ is around 15% of the Schur vector method. The complete computational cost depends on the number of iteration steps necessary. In a worst case scenario, i.e., in each step only one basis vector of $\mathcal{U}_n$ is found, the complexity of this algorithm becomes basically $\mathcal{O}(n^4)$. This is rarely observed in praxis, though. On the other hand, rounding errors during the computation, in particular while forming the multishift vector, and the fact that the eigenvalues are usually only known approximately, make it practically impossible that deflation occurs exactly. Often, some iteration steps are necessary to detect deflation when using finite precision arithmetic. Generally speaking, as long as the size of the problem is modest ($n \leq 100$), the method is feasible and the number of required iterations is acceptable.

When solving AREs, usually the stable H-invariant subspace is required. In that case, Δ_n in Proposition 4.15 has to be chosen such that $\mathrm{Re}(\lambda_j) < 0$ for all $j = 1, \ldots, n$. Note that the stable H-invariant subspace is Lagrangian; see, e.g., [2, 58]. But observe that in principle, the multishift algorithm can be used to compute the ARE solution corresponding to any Lagrangian H-invariant subspace. This is of particular importance in some applications, e.g., in some $\mathcal{H}_\infty$-control problems, ARE solutions exist and have to be computed if H has eigenvalues on the imaginary axis. As long as these eigenvalues permit a Lagrangian invariant subspace, the corresponding ARE solutions can be computed by the multishift algorithm.

Remark 4.19 *Under certain circumstances, the multishift vector in (44) becomes zero and hence, a multishift step provides no information about the desired H-invariant subspace. One situation in which this may occur is the case of* isolated *eigenvalues, e.g., if*

$$A = \begin{bmatrix} \lambda_n & 0 \\ 0 & \tilde{A} \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{G} \end{bmatrix}, \quad F = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{F} \end{bmatrix}, \quad H = \begin{bmatrix} A & G \\ F & -A^T \end{bmatrix}.$$

Hence, isolated eigenvalues should be deflated before computing a multishift step, using, e.g., the balancing procedure described in [11]. But for $\lambda(H) \cap i\mathbb{R} \neq \emptyset$, this

may also happen without isolated eigenvalues present. So far, it is not clear how to proceed in such a situation. This is under current investigation.

The computation of the multishift vector in (44) requires the knowledge of $\lambda(H)$. Hence, what remains to show is how to obtain the eigenvalues of a Hamiltonian matrix H. One possibility is to run the QR algorithm without accumulating the transformations. But then the same problems with eigenvalues close to the imaginary axis as mentioned when discussing the Schur vector method have to be expected. A different approach, which costs only one third of the QR algorithm and takes the symmetry of $\lambda(H)$ into account was suggested by Van Loan [70]. Consider $K := H^2$. Obviously, if $\lambda \in \lambda(H)$, then $\lambda_K := \lambda^2 \in \lambda(K)$. If $\mathrm{Re}(\lambda) \neq 0$, then λ_K is a double eigenvalue of K due to the symmetry of $\lambda(H)$. Squared Hamiltonian matrices are *skew-Hamiltonian*, that is, they satisfy $KJ = -(KJ)^T$ and therefore have the explicit block structure

$$K = \begin{bmatrix} K_1 & K_2 \\ K_3 & K_1^T \end{bmatrix}, \qquad K_2 = -K_2^T, \quad K_3 = -K_3^T. \tag{47}$$

The skew-Hamiltonian structure is preserved under symplectic similarity transformations [70]. Hence, computing the PVL form (41) for skew-Hamiltonian matrices yields

$$U^T H^2 U = U^T K U = \begin{bmatrix} \tilde{K}_1 & \tilde{K}_2 \\ 0 & \tilde{K}_1^T \end{bmatrix} = \begin{bmatrix} \diagdown & \square \\ & \diagdown \end{bmatrix}. \tag{48}$$

Hence, $\lambda(K)$ can be obtained by computing the eigenvalues of the upper Hessenberg matrix $\tilde{K}_1$, e.g., by applying the QR iteration to $\tilde{K}_1$. Let $\lambda(\tilde{K}_1) = \{\mu_1, \ldots, \mu_n\}$, then $\lambda(H) = \{\pm\sqrt{\mu_1}, \ldots, \pm\sqrt{\mu_n}\}$. Note that no information of eigenvectors or invariant subspaces of H is obtained.

The resulting method is strong backward stable for K and preserves the symmetry structures of $\lambda(K)$ and $\lambda(H)$. An implicit version of this algorithm is also suggested in [70]; U from (48) is applied directly to the Hamiltonian matrix such that $\tilde{H} := U^T H U$ is *square-reduced*, i.e., $\tilde{H}^2$ has the form given in (48). The disadvantage of Van Loan's method is that a loss of accuracy up to half the number of significant digits of the computed eigenvalues of H is possible. An error analysis in [70] shows that for a computed simple eigenvalue $\tilde{\lambda}$ corresponding to $\lambda \in \lambda(H)$ we have

$$|\lambda - \tilde{\lambda}| \approx \min\left\{\frac{\varepsilon\|H\|_2^2}{s(\lambda)|\lambda|}, \frac{\sqrt{\varepsilon}\|H\|_2}{s(\lambda)}\right\} = \varepsilon\frac{\|H\|_2}{s(\lambda)} \times \min\left\{\frac{\|H\|_2}{|\lambda|}, \frac{1}{\sqrt{\varepsilon}}\right\}, \tag{49}$$

where $s(\lambda)$, the reciprocal condition number of λ, is the cosine of the acute angle between the left and right eigenvectors of H corresponding to λ. Basically, this error estimate indicates that eigenvalues computed by Van Loan's method are as accurate as those computed by a numerically backward stable method provided that $\lambda \approx \|H\|_2$ while for $\lambda \ll \|H\|_2$, the error grows with the ratio $\|H\|_2/|\lambda|$.

Usually, eigenvalues computed by Van Loan's method are satisfactory as shifts for the multishift algorithm and in most other practical circumstances. On the other hand, removing the possible $1/\sqrt{\varepsilon}$ loss of accuracy provides the motivation of the algorithms presented in the next section.

A METHOD BASED ON TWO-SIDED DECOMPOSITIONS

The central problem of Van Loan's method is that squaring the Hamiltonian matrix leads to a possible loss of half of the accuracy. For products of general matrices, this possible loss of accuracy caused by forming the product can be circumvented by employing the *periodic* or *cyclic QR algorithm* [37, 38, 19]. If $A = A_1 \cdot A_2 \cdots A_p$, where $A_j \in \mathbb{R}^{n\times n}$, $j = 1, \ldots, p$, then this algorithm computes the real Schur form of A without forming A explicitly. This is achieved by cyclically reducing the factors A_j to (quasi-)upper triangular form:

$$U^T A U = (U_1^T A_1 U_2)(U_2^T A_2 U_3) \cdots (U_p^T A_p U_1) = \left[\diagdown\right] \cdot \left[\diagdown\right] \cdots \left[\diagdown\right]. \tag{50}$$

That is, $U_1^T A_1 U_2$ is in real Schur form while $U_j^T A_j U_{(j+1) \bmod p}$, $j = 2, \ldots, p$, are upper triangular such that the product is in real Schur form. The eigenvalues are then obtained from computing the eigenvalues of the 1×1 and 2×2 blocks on the diagonal of the product in (50). This method is numerically backward stable and avoids the loss of accuracy in the eigenvalues as the product A is never formed explicitly.

The idea is now to employ this approach to H^2 by replacing the reduction of H^2 to PVL form by $U^T H^2 U = (U^T H V)(V^T H U)$, where $U, V \in \mathcal{US}_{2n}$. This can be achieved by the *symplectic URV-like decomposition* given in [18].

Proposition 4.20 *For $H \in \mathbb{R}^{2n\times 2n}$ there exist $U, V \in \mathcal{US}_{2n}$ such that*

$$V^T H U = \begin{bmatrix} H_1 & H_3 \\ 0 & -H_2^T \end{bmatrix} = \begin{bmatrix} \diagdown & \square \\ 0 & \diagdown\!\!\diagdown \end{bmatrix}, \tag{51}$$

i.e., H_1 is upper triangular and H_2 is upper Hessenberg. If, in addition, H is Hamiltonian, then

$$U^T H^2 U = \begin{bmatrix} H_2 H_1 & H_2 H_3 - (H_2 H_3)^T \\ 0 & (H_2 H_1)^T \end{bmatrix} = \begin{bmatrix} \diagdown\!\!\diagdown & \square \\ 0 & \diagdown\!\!\diagdown \end{bmatrix} \tag{52}$$

and the eigenvalues of H are the positive and negative square roots of the eigenvalues of the upper Hessenberg matrix $H_2 H_1$.

That is, using the decomposition given in (51) we obtain the PVL form of H^2 without explicitly squaring H. In order to obtain the eigenvalues of H we then apply the periodic QR algorithm to $H_2 H_1$.

In [18] an algorithm for computing the decomposition given in (51) is presented. It requires a finite number of transformations. The combined cost of computing the decomposition (51) and applying the periodic QR algorithm to H_2H_1 is about $48n^3$ flops — this is 1.5 $\times$ the computational cost of Van Loan's method and about 60% of the cost of the QR algorithm applied to a non-symmetric $2n \times 2n$ matrix. The method is numerically backward stable as only orthogonal transformations are used. The symmetry property of $\lambda(H)$ is preserved and in this sense the method is strongly backward stable. But note that for the computed eigenvalues $\tilde{\lambda}$ we only get $\tilde{\lambda} \in \lambda(H+E)$ for a nearby matrix E. So far there is no proof available that E is Hamiltonian and hence that the method is strongly backward stable in the usual sense.

A detailed error analysis of the above method yields the following result [18]. Essentially (under mild assumptions), for a nonzero and simple eigenvalue λ of a Hamiltonian matrix $H \in \mathbb{R}^{2n\times 2n}$, the algorithm based on the symplectic URV-like decomposition followed by applying the periodic QR algorithm to H_2H_1 from (51) yields a computed eigenvalue $\tilde{\lambda}$ satisfying

$$|\tilde{\lambda} - \lambda| \leq \frac{2\|H\|\varepsilon}{s(\lambda)} + \mathcal{O}(\varepsilon^2).$$

This is the accuracy to be expected from any backward stable method like, e.g., the QR algorithm and shows that by avoiding to square H we get the full possible accuracy.

Nevertheless, as Van Loan's method, the approach presented above does not provide the H-invariant subspaces. But based on (51) it is possible to derive an algorithm that can be used to compute the stable H-invariant subspace and the solution of the ARE (23) [17]. The basis for this algorithm is the following theorem.

Theorem 4.21 *[17] Let $A \in \mathbb{R}^{n\times n}$ and define $B = \left[\begin{smallmatrix} 0 & A \\ A & 0 \end{smallmatrix}\right]$. Then $\lambda(B) = \lambda(A) \cup (-\lambda(A))$. Further, let $\lambda(A) \cap i\mathbb{R} = \emptyset$. If the columns of $\left[U_1^T, U_2^T\right]^T \in \mathbb{R}^{2n\times n}$ span an orthogonal basis for a B-invariant subspace such that*

$$B \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} R, \qquad \lambda(R) \subset \mathbb{C}^+ \cap \lambda(B),$$

then range$(U_1 + U_2)$ *is the A-invariant subspace corresponding to $\lambda(A) \cap \mathbb{C}^+$ and* range$(U_1 - U_2)$ *is the stable A-invariant subspace.*

An orthogonal basis for the subspace defined by range$(U_1 - U_2)$ can be obtained, e.g., from a rank-revealing QR decomposition of $U_1 - U_2$; see, e.g., [35].

In general it is of course not advisable to use the above result in order to obtain the stable invariant subspace of a matrix A as one would have to double the dimension and thereby increase the computational cost and required workspace significantly as compared to applying the QR algorithm to A. But we will see that for Hamiltonian Matrices, the given structure makes this approach very attractive.

Let $H \in \mathbb{R}^{2n\times 2n}$ be Hamiltonian with $\lambda(H) \cap \imath\mathbb{R} = \emptyset$. Define a permutation matrix $P \in \mathbb{R}^{4n\times 4n}$ by

$$P = \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & 0 & I_n & 0 \\ 0 & I_n & 0 & 0 \\ 0 & 0 & 0 & I_n \end{bmatrix}.$$

Then $P^T \left[\begin{smallmatrix} 0 & H \\ H & 0 \end{smallmatrix}\right] P$ is a Hamiltonian matrix in $\mathbb{R}^{4n\times 4n}$. The basic idea is now to employ the decomposition (51) in order to make P^THP block-upper triangular. Let $\hat{U}, \hat{V} \in \mathcal{US}_{2n}$ be as in Proposition 4.20 such that $\hat{V}^TH\hat{U}$ has the form given in (51). Then we apply the periodic QR algorithm to H_2H_1. From this we obtain orthogonal matrices $V_1, V_2 \in \mathbb{R}^{n\times n}$ such that both, the product

$$(V_1^TH_2V_2)(V_2^TH_1V_1) =: \hat{H}_2\hat{H}_1$$

and $\hat{H}_2$, are in upper real Schur form while $\hat{H}_1$ is upper triangular. Define

$$U_1 := \hat{U}\begin{bmatrix} V_1 & 0 \\ 0 & V_1 \end{bmatrix}, \quad U_2 := \hat{V}\begin{bmatrix} V_2 & 0 \\ 0 & V_2 \end{bmatrix}, \quad \text{and} \quad U := \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}.$$

Then

$$B := P^TU^T\begin{bmatrix} 0 & H \\ H & 0 \end{bmatrix}UP = \begin{bmatrix} 0 & \hat{H}_2 & 0 & \hat{H}_3^T \\ \hat{H}_1 & 0 & \hat{H}_3 & 0 \\ 0 & 0 & 0 & -\hat{H}_1^T \\ 0 & 0 & -\hat{H}_2^T & 0 \end{bmatrix}$$

is Hamiltonian and block upper triangular with $\hat{H}_1$ upper triangular, $\hat{H}_2$ in real Schur form, and $\hat{H}_3 = V_2^T(H_2H_3 - H_3^TH_2^T)V_1$.

Now let U_3 be orthogonal such that

$$U_3^T\begin{bmatrix} 0 & \hat{H}_2 \\ \hat{H}_1 & 0 \end{bmatrix}U_3 = \begin{bmatrix} T_1 & T_3 \\ 0 & -T_2 \end{bmatrix} \tag{53}$$

is in upper real Schur form with $T_j \in \mathbb{R}^{n\times n}$, $j = 1,2,3$, and $\lambda(T_1) = \lambda(T_2) \subset \mathbb{C}^+$. Note that this is possible as the eigenvalues of $\left[\begin{smallmatrix} 0 & \hat{H}_2 \\ \hat{H}_1 & 0 \end{smallmatrix}\right]$ are exactly those of H^2 and $\lambda(H) \cap \imath\mathbb{R} = \emptyset$. Hence,

$$\tilde{B} := \begin{bmatrix} U_3^T & 0 \\ 0 & U_3^T \end{bmatrix}B\begin{bmatrix} U_3 & 0 \\ 0 & U_3 \end{bmatrix} = \begin{bmatrix} T_1 & T_3 & R_1 & R_2 \\ 0 & -T_2 & R_2^T & R_3 \\ 0 & 0 & -T_1^T & 0 \\ 0 & 0 & -T_3^T & T_2^T \end{bmatrix}$$

is in Hamiltonian Schur form. In order to apply Theorem 4.21, we need to reorder the eigenvalues in the Hamiltonian Schur form such that all eigenvalues in the upper left $2n \times 2n$ block are in the open right half plane. This can be achieved,

e.g., by the symplectic re-ordering algorithm due to Byers [23, 24]. With this algorithm it is possible to determine $\tilde{U} \in \mathcal{US}_{2n}$ such that

$$\tilde{U}^T \tilde{B} \tilde{U} = \begin{bmatrix} T_1 & \tilde{T}_3 & R_1 & \tilde{R}_2 \\ 0 & \tilde{T}_2 & \tilde{R}_2^T & R_3 \\ 0 & 0 & -T_1^T & 0 \\ 0 & 0 & -\tilde{T}_3^T & -\tilde{T}_2^T \end{bmatrix}, \qquad \lambda(\tilde{T}_2) = \lambda(T_2).$$

Now define

$$S := P^T \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} P \begin{bmatrix} U_3 & 0 \\ 0 & U_3 \end{bmatrix} \tilde{U}. \tag{54}$$

Then $S \in \mathcal{US}_{4n}$ and

$$T := S^T P^T \begin{bmatrix} 0 & H \\ H & 0 \end{bmatrix} PS =: \begin{bmatrix} T_{11} & T_{12} \\ 0 & -T_{11}^T \end{bmatrix} \tag{55}$$

is in Hamiltonian Schur form with $\lambda(T_{11}) \subset \mathbb{C}^+$. Now we can apply Theorem 4.21 with A replaced by H and $R := T_{11}$.

Corollary 4.22 *Let $H \in \mathbb{R}^{2n \times 2n}$ be Hamiltonian with $\lambda(H) \cap \imath\mathbb{R} = \emptyset$ and let S be as in (54) such that (55) holds. If $PS := \left[\begin{smallmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{smallmatrix}\right]$, with $S_{ij} \in \mathbb{R}^{2n \times 2n}$, then the n-dimensional, stable H-invariant subspace is given by* $\operatorname{range}(S_{11} - S_{21})$.

The above transformations yielding S are described in more detail in [17]. The solution of the ARE can be obtained from an orthogonal basis of $\operatorname{range}(S_{11} - S_{21})$ computed by a rank-revealing QR decomposition or directly from $S_{11} - S_{21}$; for details see [17]. The latter approach saves a significant amount of work such that the cost of the algorithm described above for computing the stabilizing solution of the ARE (23) is approximately 60% of the cost of the Schur vector method.

Remark 4.23 *The transformation of $\left[\begin{smallmatrix} 0 & \hat{H}_2 \\ \hat{H}_1 & 0 \end{smallmatrix}\right]$ to real Schur form and the computation of the matrix U_3 in (53) can be efficiently implemented employing the available structure. An algorithm for this is given in [17].*

Remark 4.24 *The algorithm described here can in principle also be applied to Hamiltonian matrices with eigenvalues on the imaginary axis. In some cases (depending on multiplicities of the pure imaginary eigenvalues) it is not yet clear how to pick the "right" Lagrangian subspace yielding the desired solution of the ARE. This topic is under current investigation.*

It is shown in [17] that the algorithm presented above is strongly backward stable in $\mathbb{R}^{4n \times 4n}$. That is, if $\tilde{S}$ is the analogue to S from (54) computed in finite precision arithmetic, then

$$\tilde{S}^T P^T \begin{bmatrix} 0 & H \\ H & 0 \end{bmatrix} P\tilde{S} = T + E,$$

with T as in (55), $||E||_2 \leq c\varepsilon||H||_2$ for a small constant c and $E \in \mathbb{R}^{4n\times 4n}$ is Hamiltonian. Moreover it is shown in [17] that the computed invariant subspace is as accurate as the maximum of its condition number and the condition number of its complimentary (*antistable*) H-invariant subspace permit. This is to be expected from the fact that at the same time we compute the stable H-invariant subspace, by Theorem 4.21 we also compute the antistable H-invariant subspace. In that sense the algorithm is not optimal as we would like the accuracy of the computed subspace to be limited only by its own condition number.

Concluding the section about symplectic methods, the multishift algorithm as well as the method based on the symplectic URV-like decomposition (51) are big steps towards a strong backward stable method of cubic complexity. The ideal method has not yet been found, though.

4.1.4 Comparison of methods based on the Hamiltonian eigenproblem

In this subsection we will give a comparison of some of the methods presented in Section 4.1. The computations were done in MATLAB[2] Version 5.1 with machine precision $\varepsilon \approx 2.2204 \times 10^{-16}$. The following methods are tested:

- **EV**: the eigenvector approach outlined in Section 4.1.1 and implemented in the MATLAB Control Toolbox Version 3.0b function `lqr` [57];
- **SCHUR**: the Schur vector method as implemented in the MATLAB Control Toolbox Version 3.0b function `are` [57];
- **MSH**: an implementation of the multishift algorithm as described in [10];
- **SURV**: an implementation of the method based on the two-sided, symplectic URV-like decomposition.

The methods are tested for the Examples 1–19 of the benchmark collection for continuous-time algebraic Riccati equations [15, 16]. (Note that Example 20 is missing as this example could not be solved on the author's machine due to insufficient memory.) Table 1 shows the 2-norms of the normalized residuals, i.e., $||\mathcal{R}(X)||_2/||X||_2$, where X are the solutions computed by the above methods. In Table 2 we list the relative errors for those examples where the exact stabilizing solution is known.

From both tables it is obvious that none of the methods is superior to all other methods. But note that for almost all examples, either one of the structure preserving methods (MSH and SURV) produces the smallest residuals and relative errors. In particular for almost all small size examples, the multishift method yields the best solutions. Note that this is also true for Examples 11 and 14 though the reported residuals for (some of) the other methods are smaller. For Example 11, this can be clearly seen from the relative error. Here, the Hamiltonian matrix has eigenvalues on the imaginary axis causing the other methods to loose half the

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

Ex. no.	n	EV	SCHUR	MSH	SURV
1	2	1.4×10^{-8}	2.1×10^{-15}	2.4×10^{-16}	0.0
2	2	1.4×10^{-14}	5.8×10^{-15}	4.2×10^{-15}	1.2×10^{-14}
3	4	2.8×10^{-15}	3.9×10^{-15}	1.7×10^{-15}	2.3×10^{-15}
4	8	1.3×10^{-15}	8.5×10^{-16}	6.2×10^{-16}	1.9×10^{-15}
5	9	2.0×10^{-13}	7.6×10^{-14}	8.4×10^{-15}	2.7×10^{-14}
6	30	2.0×10^{-12}	3.2×10^{-11}	4.2×10^{-10}	3.6×10^{-8}
7	2	1.5×10^{-28}	4.4×10^{-5}	1.2×10^{-4}	1.7×10^{-4}
8	2	1.8×10^{-8}	4.6×10^{-9}	2.5×10^{-8}	1.6×10^{-8}
9	2	3.5×10^{-12}	3.9×10^{-10}	3.3×10^{-13}	5.8×10^{-11}
10	2	1.2×10^{-15}	3.3×10^{-15}	4.4×10^{-16}	4.5×10^{-16}
11	2	1.2×10^{-15}	5.5×10^{-16}	1.1×10^{-15}	9.5×10^{-10}
12	3	3.6×10^{3}	2.5×10^{3}	5.7×10^{3}	3.3×10^{3}
13	4	4.4×10^{-11}	6.8×10^{-10}	4.1×10^{-12}	2.2×10^{-5}
14	4	6.0×10^{-15}	2.5×10^{-15}	3.8×10^{-13}	3.8×10^{-15}
15	39	9.3×10^{-15}	8.6×10^{-15}	6.6×10^{-13}	3.4×10^{-15}
16	64	1.0×10^{-14}	1.2×10^{-14}	1.6×10^{-13}	7.3×10^{-15}
17	21	8.1×10^{-7}	9.7×10^{-7}	2.8×10^{-8}	8.7×10^{-7}
18	100	3.4×10^{-9}	3.0×10^{-9}	1.6×10^{-5}	1.0×10^{-12}
19	60	2.3×10^{-14}	2.6×10^{-14}	2.0×10^{-11}	4.0×10^{-15}

Table 1: $||\mathcal{R}(\tilde{X})||_2/||\tilde{X}||_2$ for tested methods.

Ex. no.	EV	SCHUR	MSH	SURV
1	6.9×10^{-9}	7.0×10^{-16}	7.4×10^{-17}	7.0×10^{-16}
2	6.9×10^{-15}	1.4×10^{-15}	1.3×10^{-15}	4.7×10^{-15}
7	8.3×10^{-29}	2.2×10^{-5}	5.9×10^{-5}	8.3×10^{-5}
9	1.8×10^{-15}	1.2×10^{-14}	1.6×10^{-16}	4.1×10^{-14}
10	5.2×10^{-16}	7.5×10^{-16}	1.2×10^{-11}	1.6×10^{-16}
11	1.0×10^{-8}	1.6×10^{-8}	6.3×10^{-16}	2.1×10^{-8}
12	6.4×10^{-4}	7.0×10^{-4}	9.5×10^{-4}	5.7×10^{-4}
16	3.0×10^{-15}	3.4×10^{-15}	2.9×10^{-14}	1.9×10^{-15}
17^3	1.0×10^{-6}	1.1×10^{-6}	6.6×10^{-9}	8.3×10^{-7}

Table 2: $||X^* - \tilde{X}||_2/||X^*||_2$ for tested methods.

number of significant digits while the multishift method computes the solution to full accuracy. (From this example it can be seen that sometimes the residual gives misleading information about the quality of the computed solution; see also [43].) In Example 14, MSH yields the largest residual, but the other methods do compute a non-symmetric solution matrix which can by theory not be the desired (and existing) stabilizing solution. For Example 10, the larger relative error of the solution computed by MSH can be explained from the loss of accuracy of the eigenvalues computed by the square-reduced method. Here, two eigenvalues $\pm\lambda$ are close to zero and a loss of almost half the digits must be expected from (49) and the fact that $\|H\|_2/|\lambda| = \mathcal{O}(10^7)$.

For the problems of larger dimension (Examples 15, 16, 18, 19), the method based on the symplectic URV-like decomposition produces the best results while the multishift method suffers from convergence problems and looses 1 to 3 orders of magnitude compared to SURV.

Note that in Examples 6 and 7 which are the only ones for which the eigenvector approach performs best, there are isolated eigenvalues present. Hence, EV benefits from balancing the matrix while this is not employed by SCHUR as implemented in `are` and the structure preserving methods MSH and SURV. The use of the symplectic balancing procedure described in [11] will resolve this disadvantage of the structured methods as preliminary numerical tests indicate.

The large residuals in Examples 7, 12, and 17 are due to badly scaled algebraic Riccati equations. The relative errors obtained in these examples are in accordance with the condition of the matrix U_{11} which has to be factored in order to solve for X.

From the examples with eigenvalues close to the imaginary axis it seems that the multishift algorithm can handle this problem a little better than SURV (which can be explained by the fact that it is not affected by the conditioning of the anti-stable H-invariant subspace). On the other hand, SURV overcomes the problems of the multishift method for growing dimensions while still being substantially faster than the Schur vector method.

Remark 4.25 *The numerical solution of AREs by any of the methods described in this section should always be followed by at least one step of iterative refinement using Newton's method (see Section 4.3).*

4.2 Spectral Projection Methods

Given a projector $\mathcal{P}$ onto the stable H-invariant subspace $\mathcal{S}$, i.e., range$(\mathcal{P}) = \mathcal{S}$ and $\mathcal{P}^2 = \mathcal{P}$, the solution of the ARE can be obtained as in (28) by computing an orthogonal basis for range$(\mathcal{P})$, e.g., by a rank-revealing QR decomposition.

One of the most popular spectral projection methods is the *matrix sign function method*. This method was first introduced 1971 by Roberts [67] in order to solve the ARE as given in (23). The *matrix sign function* of a matrix $Z \in \mathbb{R}^{n\times n}$ can

[3] In this example, the only known component of the solution is $x_{1,n} = x_{n,1} = 1$. The relative errors reported are the relative errors for this single matrix entry.

be defined as follows. Let $\lambda(Z) \cap i\mathbb{R} = \emptyset$ and let the Jordan decomposition of Z be given as

$$Z = S \begin{bmatrix} J^- & 0 \\ 0 & J^+ \end{bmatrix} S^{-1}, \tag{56}$$

where Jordan blocks corresponding to the, say, k eigenvalues in the open left half plane are collected in J^- and Jordan blocks corresponding to the remaining $n-k$ eigenvalues in the open right half plane are collected in J^+. Then

$$\operatorname{sign}(Z) := S \begin{bmatrix} -I_k & 0 \\ 0 & I_{n-k} \end{bmatrix} S^{-1}. \tag{57}$$

From this definition we immediately obtain an important property of the sign function: $\mathcal{P}^- = \frac{1}{2}(I_n - \operatorname{sign}(Z))$ defines a skew projector onto the stable Z-invariant subspace parallel to the antistable Z-invariant subspace whereas $\mathcal{P}^+ = \frac{1}{2}(I_n + \operatorname{sign}(Z))$ defines a skew projector onto the antistable Z-invariant subspace parallel to the stable one.

The sign function can be computed via the Newton iteration for the equation $S^2 = I$ where the starting point is chosen as Z, i.e.,

$$Z_0 \leftarrow Z, \qquad \text{for} \quad j = 1, 2, \ldots, \quad Z_{j+1} \leftarrow \frac{1}{2}\left(Z_j + Z_j^{-1}\right). \tag{58}$$

It is shown in [67] that $\lim_{j\to\infty} Z_j = \operatorname{sign}(Z)$.

Although convergence of the Newton iteration (58) is globally quadratic, the initial convergence may be slow. There have been several proposals to accelerate the convergence of this iteration by scaling each iterate Z_j by a factor γ_j, see, e.g., [25, 41, 67]. Several other iterative schemes have been developed for computing the sign function of a matrix. For a summary see the recent review paper [41].

As the sign function method computes a splitting of the spectrum along the imaginary axis, it has a natural relationship to the problem of computing stable invariant subspaces. Recalling the results of Section 3 it is now easy to see that the linear-quadratic optimization problem (1)–(3) and the ARE (23) can be solved by applying the sign function method to the corresponding Hamiltonian matrix H. This approach was introduced in [67] and studied in plenty of subsequent papers; see [41] and the references given therein. In particular, the application to Hamiltonian matrices and exploiting their special structure was investigated in [25]. Moreover, it was observed in [67, 25] that the solution of AREs can be computed directly from $\operatorname{sign}(H)$. We will briefly describe the basis for this observation here, using a slightly different approach. Let

$$Z_\infty = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} = \operatorname{sign}(H)$$

be the limit of the Newton iteration (58) applied to H. From the considerations in Section 3 we know that the columns of $[I_n, -X_*]^T$ span a basis for the stable H-invariant subspace. Hence, they are contained in the null space of any projector

onto the antistable H-invariant subspace and thus particularly in the null space of $Z_\infty + I_{2n}$. Thus,

$$(Z_\infty + I_{2n}) \begin{bmatrix} I \\ X \end{bmatrix} = 0.$$

Hence, X_* satisfies the overdetermined system of linear equations

$$\begin{bmatrix} Z_{12} \\ Z_{22} + I_{2n} \end{bmatrix} X_* = - \begin{bmatrix} Z_{11} + I_{2n} \\ Z_{21} \end{bmatrix}. \tag{59}$$

It can be shown (see, e.g., [48, Section 22] and the references given there) that the coefficient matrix of the left-hand side of (59) has full rank (and hence the linear system is consistent) if and only if the stabilizing solution X_* of the corresponding ARE exists.

The sign function method has the advantage that it is still feasible for medium sized linear-quadratic optimization problems and AREs with dense coefficient matrices (up to order $n \approx 5000$) whereas most of the methods mentioned in the last section for problems of this size are much slower and/or suffer from convergence problems. It can also easily be adapted for parallel computation and serves therefore as basis for most parallel algorithms for solving AREs, see, e.g., [32, 64].

The disadvantage of the sign function method is that $\operatorname{sign}(Z)$ is not defined if $\lambda(H) \cap i\mathbb{R} \neq \emptyset$ and the iterative schemes for computing $\operatorname{sign}(Z)$ perform poorly or may fail when there are eigenvalues close to the imaginary axis. There are approaches to resolve this problem; see [31].

Recently, another iterative method was suggested that computes a projector onto invariant or deflating subspaces of matrices or matrix pencils corresponding to eigenvalues inside or outside the unit disk [55, 6]. The application of this method, following [11, 12] called the *disk function method*, to AREs and linear-quadratic optimal control problems is investigated in [55, 7, 11, 12]. Though by the computed spectral splitting it has a natural relationship to discrete-time optimal control problems, it can be applied to the continuous-time problems considered here by using an appropriate spectral transformation. The method has similar advantages and disadvantages as the sign function method: it can easily be parallelized but gets into trouble if the spectrum of the matrix or matrix pencil is poorly separated with respect to the computed spectral splitting.

4.3 Newton's Method

The methods presented so far have addressed the ARE by its relation to the Hamiltonian eigenproblem. By nature, the ARE (23) represents a system of nonlinear equations. It is therefore straightforward to apply methods for solving nonlinear equations to the ARE. In [45], Kleinman shows that Newton's method, applied to the ARE and properly initialized, converges to the desired stabilizing solution of the ARE (see Theorem 4.27 below). The resulting algorithm can be stated in different ways. We have chosen here the variant that is most robust with respect to accumulation of rounding errors.

Algorithm 4.26 (Newton's method for the generalized CARE).

Input: *$A, G, F \in \mathbb{R}^{n\times n}$, $G = G^T$, $F = F^T$, $X_0 = X_0^T$ – an initial guess.*
Output: *Approximate solution $X_{j+1} \in \mathbb{R}^{n\times n}$ of (23).*
FOR $j = 0, 1, 2, \ldots$ "until convergence"

1. $A_j = A - GX_j$.
2. Solve for N_j in the *Lyapunov equation* $0 = \mathcal{R}(X_j) + A_j^T N_j + N_j A_j$.
3. $X_{j+1} = X_j + N_j$.

END FOR

We have the following results for Algorithm 4.26 [48].

Theorem 4.27 *If $G \geq 0$, (A, G) is stabilizable, the unique stabilizing solution X_* of the ARE exists, and X_0 is stabilizing, then for the iterates produced by Algorithm 4.26 we have:*

(i) *All iterates X_j are stabilizing, i.e., $\lambda(A - GX_j) \subset \mathbb{C}^-$ for all $j \in \mathbb{N}_0$.*
(ii) $X_* \leq \ldots \leq X_{j+1} \leq X_j \leq \ldots \leq X_1$.
(iii) $\lim_{j\to\infty} X_j = X_*$.
(iv) *There exists a constant $\gamma > 0$ such that*

$$\|X_{j+1} - X_*\| \leq \gamma \|X_j - X_*\|^2, \qquad j \geq 1,$$

i.e., the X_j converge globally quadratic to X_*.

Older versions of this theorem [45, 58] usually need stronger assumptions than those used here.

Finding a stabilizing X_0 usually is a difficult task and requires a computational cost equivalent to one iteration step of Newton's method; see, e.g., [69] and the references therein. Moreover, X_0 determined by a stabilization procedure may lie far from X_*. Though ultimately quadratic convergent, Newton's method may initially converge slowly. This can be due to a large error $\|X_0 - X_*\|$ or to a disastrously bad first step, leading to a large error $\|X_1 - X_*\|$; see, e.g., [42, 11, 13]. The computational cost for solving the ARE with the Schur vector method is roughly equivalent to 5–7 iterations of Algorithm 4.26. Due to the initial slow convergence, Newton's method often requires more than 7 iterations. Therefore it is most frequently only used to refine an approximate ARE solution computed by any other method.

Recently an exact line search procedure was suggested that accelerates the initial convergence and avoids "bad" first steps [11, 13]. Specifically, Step 3. of Algorithm 4.26 is modified to $X_{j+1} = X_j + t_j N_j$, where t_j is chosen in order to minimize the Frobenius norm of the residual $\mathcal{R}(X_j + tN_j)$. As computing the exact minimizer is very cheap compared to a Newton step and usually accelerates the initial convergence significantly while benefiting from the quadratic convergence

of Newton's method close to the solution, this method becomes attractive, even as a solver for AREs (at least in some cases), see [11, 9, 13] for details. Moreover, for some ill-conditioned AREs, exact line search improves Newton's method also when used only for iterative refinement.

5 Concluding Remarks

The linear-quadratic optimization problem arising in optimal control can be tackled by many different solution methods and plenty of numerical algorithms for its solution have been investigated. The approaches based on the solution of the corresponding Riccati equations turn out to be the most efficient and reliable ones. Again there are many different algorithms for solving these symmetric and quadratic, algebraic or differential matrix equations. For the most frequently considered case, that is, the infinite time horizon case, an algebraic Riccati equation has to be solved. This can be done by addressing the ARE as a set of nonlinear equations or via the corresponding Hamiltonian eigenproblem. Though very efficient methods from numerical linear algebra are available for this structured eigenproblem, an ideal method has still not been found. We have presented some of these methods and discussed their advantages and disadvantages. Any of these methods can be used to compute an approximate solution which should then be refined using Newton's method. With this approach, the ARE can be solved in a very efficient and reliable way. In some circumstances, Newton's method itself, endowed with exact line search, can also be used as solution method of the ARE.

Acknowledgments

I would like to thank Emanuele Galligani who initiated the writing of this survey by inviting me to give tutorial talks about this subject. Moreover, I would like to thank my co-authors Greg Ammar, Ralph Byers, Heike Faßbender, Volker Mehrmann, and Hongguo Xu — our joint research made this article possible. My thanks also go to Anne tom Suden and Tanja Brüggemann for typesetting parts of the manuscript.

References

[1] G.S. Ammar, P. Benner, and V. Mehrmann. A multishift algorithm for the numerical solution of algebraic Riccati equations. *Electr. Trans. Num. Anal.*, 1:33–48, 1993.

[2] G.S. Ammar and V. Mehrmann. On Hamiltonian and symplectic Hessenberg forms. *Linear Algebra Appl.*, 149:55–72, 1991.

[3] B.D.O. Anderson and J.B. Moore. *Linear Optimal Control.* Prentice-Hall, Englewood Cliffs, NJ, 1971.

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

[5] M. Athans and P.L. Falb. *Optimal Control.* McGraw-Hill, New York, 1966.

[6] Z. Bai, J. Demmel, and M. Gu. An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. *Numer. Math.*, 76(3):279–308, 1997. *See also:* Tech. Report LBL-34969, Lawrence Berkeley Laboratory, University of California, Berkeley, CA 94720.

[7] Z. Bai and Q. Qian. Inverse free parallel method for the numerical solution of algebraic Riccati equations. In J.G. Lewis, editor, *Proc. Fifth SIAM Conf. Appl. Lin. Alg., Snowbird, UT, June 1994*, pages 167–171. SIAM, Philadelphia, PA, 1994.

[8] A.N. Beavers and E.D. Denman. A new solution method for quadratic matrix equations. *Mathematical Biosciences*, 20:135–143, 1974.

[9] P. Benner. Numerical solution of special algebraic Riccati equations via an exact line search method. In *Proc. European Control Conf. ECC 97,* Paper 786. BELWARE Information Technology, Waterloo, Belgium, 1997. CD-ROM.

[10] P. Benner. Ein orthogonal symplektischer Multishift Algorithmus zur Lösung der algebraischen Riccatigleichung. Diplomarbeit, RWTH Aachen, Institut für Geometrie und Praktische Mathematik, Aachen, FRG, March 1993. In German.

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

[12] P. Benner and R. Byers. Disk functions and their relationship to the matrix sign function. In *Proc. European Control Conf. ECC 97,* Paper 936. BELWARE Information Technology, Waterloo, Belgium, 1997. CD-ROM.

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

[14] P. Benner and H. Faßbender. An implicitly restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem. *Linear Algebra Appl.*, 263:75–111, 1997.

[15] P. Benner, A.J. Laub, and V. Mehrmann. A collection of benchmark examples for the numerical solution of algebraic Riccati equations I: Continuous-time case. Technical Report SPC 95_22, Fakultät für Mathematik, TU Chemnitz–Zwickau, 09107 Chemnitz, FRG, 1995. Available from `http://www.tu-chemnitz.de/sfb393/spc95pr.html`.

[16] P. Benner, A.J. Laub, and V. Mehrmann. Benchmarks for the numerical solution of algebraic Riccati equations. *IEEE Control Systems Magazine*, 7(5):18–28, 1997.

[17] P. Benner, V. Mehrmann, and H. Xu. A new method for computing the stable invariant subspace of a real Hamiltonian matrix. *J. Comput. Appl. Math.*, 86:17–43, 1997.

[18] 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(3):329–358, 1998.

[19] A. Bojanczyk, G.H. Golub, and P. Van Dooren. The periodic Schur decomposition; algorithms and applications. In *Proc. SPIE Conference, vol. 1770*, pages 31–42, 1992.

[20] A. Bunse-Gerstner. Matrix factorization for symplectic QR-like methods. *Linear Algebra Appl.*, 83:49–77, 1986.

[21] A. Bunse-Gerstner and H. Faßbender. A Jacobi-like method for solving algebraic Riccati equations on parallel computers. *IEEE Trans. Automat. Control*, 42(8):1071–1084, 1997.

[22] A. Bunse-Gerstner and V. Mehrmann. A symplectic QR-like algorithm for the solution of the real algebraic Riccati equation. *IEEE Trans. Automat. Control*, AC-31:1104–1113, 1986.

[23] R. Byers. *Hamiltonian and Symplectic Algorithms for the Algebraic Riccati Equation*. PhD thesis, Cornell University, Dept. Comp. Sci., Ithaca, NY, 1983.

[24] R. Byers. A Hamiltonian QR-algorithm. *SIAM J. Sci. Statist. Comput.*, 7:212–229, 1986.

[25] R. Byers. Solving the algebraic Riccati equation with the matrix sign function. *Linear Algebra Appl.*, 85:267–279, 1987.

[26] R. Byers. A Hamiltonian-Jacobi algorithm. *IEEE Trans. Automat. Control*, 35:566–570, 1990.

[27] J.L. Casti. *Linear Dynamical Systems*. Mathematics in Science and Engineering. Academic Press, New York, 1987.

[28] C. Choi and A.J. Laub. Efficient matrix-valued algorithms for solving stiff Riccati differential equations. *IEEE Trans. Automat. Control*, 35:770–776, 1990.

[29] L. Dieci, Y.M. Lee, and R.D. Russel. Iterative methods for solving algebraic Riccati equations. Report, Dept. Math. & Stat., Simon Fraser University, Burnaby, Canada, 1988.

[30] B.A. Francis. *A Course In $\mathcal{H}_\infty$ Control Theory*, volume 88 of *Lecture Notes in Control and Information Sciences*. Springer–Verlag, Berlin, 1987.

[31] J. D. Gardiner. A stabilized matrix sign function algorithm for solving algebraic Riccati equations. *SIAM J. Sci. Comput.*, 18:1393–1411, 1997.

[32] J.D. Gardiner and A.J. Laub. Parallel algorithms for algebraic Riccati equations. *Internat. J. Control*, 54:1317–1333, 1991.

[33] A.R. Ghavimi, C. Kenney, and A.J. Laub. Local convergence analysis of conjugate gradient methods for solving algebraic Riccati equations. *IEEE Trans. Automat. Control*, 37:1062–1067, 1992.

[34] I. Gohberg, P. Lancaster, and L. Rodman. *Matrices and Indefinite Scalar Products*. Birkhäuser, Basel, 1983.

[35] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[36] M. Green and D.J.N Limebeer. *Linear Robust Control*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

[37] J.J. Hench and A.J. Laub. An extension of the QR algorithm for a sequence of matrices. Technical Report CCEC-92-0829, ECE Dept., University of California, Santa Barbara, CA, 1992.

[38] J.J. Hench and A.J. Laub. Numerical solution of the discrete-time periodic Riccati equation. *IEEE Trans. Automat. Control*, 39:1197–1210, 1994.

[39] R.E. Kalman. Contributions to the theory of optimal control. *Boletin Sociedad Matematica Mexicana*, 5:102–119, 1960.

[40] R.E. Kalman and R.S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME, Series D*, 83:95–108, 1961.

[41] C. Kenney and A.J. Laub. The matrix sign function. *IEEE Trans. Automat. Control*, 40(8):1330–1348, 1995.

[42] C. Kenney, A.J. Laub, and M. Wette. A stability-enhancing scaling procedure for Schur-Riccati solvers. *Sys. Control Lett.*, 12:241–250, 1989.

[43] C. Kenney, A.J. Laub, and M. Wette. Error bounds for Newton refinement of solutions to algebraic Riccati equations. *Math. Control, Signals, Sys.*, 3:211–224, 1990.

[44] C. Kenney and R.B. Leipnik. Numerical integration of the differential matrix Riccati equation. *IEEE Trans. Automat. Control*, AC-30:962–970, 1985.

[45] D. L. Kleinman. On an iterative technique for Riccati equation computations. *IEEE Trans. Automat. Control*, AC-13:114–115, 1968.

[46] V. Kučera. A contribution to matrix quadratic equations. *IEEE Trans. Automat. Control*, AC-17:344–347, 1972.

[47] P. Kunkel and V. Mehrmann. Numerical solution of Riccati differential algebraic equations. *Linear Algebra Appl.*, 137/138:39–66, 1990.

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

[49] A.J. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Automat. Control*, AC-24:913–921, 1979.

[50] A.J. Laub. Invariant subspace methods for the numerical solution of Riccati equations. In S. Bittanti, A.J. Laub, and J.C. Willems, editors, *The Riccati Equation*, pages 163–196. Springer-Verlag, Berlin, 1991.

[51] F. Lin. *Robust Control Design: An Optimal Control Approach.* AFI Press, 1997.

[52] W.-W. Lin and T.-C. Ho. On Schur type decompositions for Hamiltonian and symplectic pencils. Technical report, Institute of Applied Mathematics, National Tsing Hua University, Taiwan, 1990.

[53] W.-W. Lin, V. Mehrmann, and H. Xu. Canonical forms for Hamiltonian and symplectic matrices and pencils, preliminary version by last two authors available as. Technical Report SFB393/98-7, Fakultät für Mathematik, TU Chemnitz, 09107 Chemnitz, FRG, 1998. Available from `http://www.tu-chemnitz.de/sfb393/sfb98pr.html`.

[54] J. Macki and A. Strauss. *Introduction to Optimal Control Theory.* Springer-Verlag, 1982.

[55] A.N. Malyshev. Parallel algorithm for solving some spectral problems of linear algebra. *Linear Algebra Appl.*, 188/189:489–520, 1993.

[56] F.T. Man. The Davidon method of solution of the algebraic matrix Riccati equation. *Internat. J. Control*, 10(6):713–719, 1969.

[57] The MathWorks, Inc., Cochituate Place, 24 Prime Park Way, Natick, Mass, 01760. *The MATLAB Control Toolbox, Version 3.0b*, 1993.

[58] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution.* Number 163 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Heidelberg, July 1991.

[59] C.C. Paige and C.F. Van Loan. A Schur decomposition for Hamiltonian matrices. *Linear Algebra Appl.*, 14:11–32, 1981.

[60] P. Pandey. Quasi–Newton methods for solving algebraic Riccati equations. In *Proc. American Control Conf.*, pages 654–658, Chicago, IL, June 1992.

[61] P.H. Petkov, N.D. Christov, and M.M. Konstantinov. *Computational Methods for Linear Control Systems.* Prentice-Hall, Hertfordshire, UK, 1991.

[62] E.R. Pinch. *Optimal Control and the Calculus of Variations.* Oxford University Press, Oxford, UK, 1993.

[63] J.E. Potter. Matrix quadratic solutions. *SIAM J. Appl. Math.*, 14:496–501, 1966.

[64] E.S. Quintana-Ortí and V. Hernández. Parallel algorithms for solving the algebraic Riccati equation via the matrix sign function. *Automatica*, 34:151–156, 1998.

[65] A.C. Raines and D.S. Watkins. A class of Hamiltonian–symplectic methods for solving the algebraic Riccati equation. *Linear Algebra Appl.*, 205/206:1045–1060, 1994.

[66] W.T. Reid. *Riccati Differential Equations.* Academic Press, New York, 1972.

[67] J.D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *Internat. J. Control*, 32:677–687, 1980. (Reprint of Technical Report No. TR-13, CUED/B-Control, Cambridge University, Engineering Department, 1971).

[68] A. Saberi, P. Sannuti, and B.M. Chen. H_2 *Optimal Control.* Prentice-Hall, Hertfordshire, UK, 1995.

[69] V. Sima. *Algorithms for Linear-Quadratic Optimization*, volume 200 of *Pure and Applied Mathematics.* Marcel Dekker, Inc., New York, NY, 1996.

[70] C.F. Van Loan. A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix. *Linear Algebra Appl.*, 61:233–251, 1984.

[71] D.S. Watkins and L. Elsner. Convergence of algorithms of decomposition type for the eigenvalue problem. *Linear Algebra Appl.*, 143:19–47, 1991.

[72] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice-Hall, Upper Saddle River, NJ, 1996.

Berichte aus der Technomathematik ISSN 1435-7968

http://www.math.uni-bremen.de/zetem/berichte

— Vertrieb durch den Autor —

Reports **Stand: 2. September 1998**

98–01. Peter Benner, Heike Faßbender:
An Implicitly Restarted Symplectic Lanczos Method for the Symplectic Eigenvalue Problem, Juli 1998.

98–02. Heike Faßbender:
Sliding Window Schemes for Discrete Least-Squares Approximation by Trigonometric Polynomials, Juli 1998.

98–03. Peter Benner, Maribel Castillo, Enrique S. Quintana-Ortí:
Parallel Partial Stabilizing Algorithms for Large Linear Control Systems, Juli 1998.

98–04. Peter Benner:
Computational Methods for Linear–Quadratic Optimization, August 1998.

98–05. Peter Benner, Ralph Byers, Enrique S. Quintana-Ortí, Gregorio Quintana-Ortí:
Solving Algebraic Riccati Equations on Parallel Computers Using Newton's Method with Exact Line Search, August 1998.