MAX-PLANCK-GESELLSCHAFT

Max Planck Institute Magdeburg
Preprints

Peter Benner Jens Saak

Numerical Solution of Large and Sparse Continuous Time Algebraic Matrix Riccati and Lyapunov Equations: A State of the Art Survey

MAX–PLANCK–INSTITUT
FÜR DYNAMIK KOMPLEXER
TECHNISCHER SYSTEME
MAGDEBURG

MPIMD/13-07 June 24, 2013

Abstract

Efficient numerical algorithms for the solution of large and sparse matrix Riccati and Lyapunov equations based on the low rank alternating directions implicit (ADI) iteration have become available around the year 2000. Over the decade that passed since then, additional methods based on extended and rational Krylov subspace projection have entered the field and proved to be competitive alternatives. In this survey we sketch both types of methods and discuss their advantages and drawbacks. We focus on the continuous time case here, but corresponding results for discrete time problems can for most results be found in the available literature and will be referred to throughout the paper.

Contents

Author's addresses:

Peter Benner and Jens Saak
Computational Methods in Systems and Control Theory, Max Planck Institute
for Dynamics of Complex Technical Systems,
Sandtorstr. 1,
39106 Magdeburg

Germany,

benner@mpi-magdeburg.mpg.de,saak@mpi-magdeburg.mpg.de

1 Introduction

Throughout this paper we will consider algebraic matrix equations related to the linear time invariant dynamical system in generalized state space form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t). \end{aligned} \tag{1}$$

We assume E, $A \in \mathbb{R}^{n\times n}$ sparse, and non singular, as well as $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$. Though we assume real matrices here, most of the algorithms discussed in the following can be applied in the complex case with minor, if at all, modifications. In order to allow for low rank approximations of the solutions we further assume $p \ll n$ and $m \ll n$.

In the case where we are interested in the solutions P and Q of the Lyapunov matrix equations

$$APE^T + EPA^T = -BB^T, \qquad A^TQE + E^TQA = -C^TC, \tag{2}$$

we additionally assume the system (1) to be asymptotically stable, i.e., $\Lambda(A, E) \subset \mathbb{C}_{<0} := \{ s \in \mathbb{C} \,|\, \mathrm{Re}\,(s) < 0 \}$, such that the equations in (2) have unique solutions. The solutions of these two equations are the main ingredient in Balanced Truncation based model order reduction for linear time invariant systems (1).

For the sake of simplicity in the derivations we also consider the formulation

$$FX + XF^T = -GG^T, \tag{3}$$

where, e.g., F is either A or A^T and G is B or C^T and $E = I$ is the identity matrix to match (2). Alternatively for theoretic considerations we can simply replace A by $E^{-1}A$ and B by $E^{-1}B$ to retrieve a system of the form (1) with $E = I$.

For the algebraic Riccati equation

$$C^TC + A^TXE + E^TXA - E^TXBB^TXE = 0, \tag{4}$$

the set of solutions is in general large due to the quadratic nature of the equation. To make a solution unique it needs to have additional properties. One is usually interested in one specific solution among all possible solutions. In the context of optimal control this is the unique maximal positive semidefinite symmetric solution that stabilizes the system (1). The stabilization is then performed in the sense that

$$E\dot{x}(t) = (A - BB^TXE)x(t) \tag{5}$$

is asymptotically stable, i.e. all eigenvalues of the pencil are located in the open left half plane. The distinguished solution of (4) is then simply called the stabilizing solution. Additional requirements to allow for a stabilizing solution can be expressed as stabilizability and detectability of the system (1) (see, e.g., [35]).

The key ingredient towards an efficient handling of the above matrix equations is the observation that the solution can be represented in forms other than the dense square matrix form. Throughout this paper, we will focus on the low rank representation of solutions in the form $X \approx ZZ^H$ for a possibly complex factor Z with $k \ll n$ columns. Several contributions [46, 25, 2, 51, 59] have investigated the singular value decay in X, especially in the Lyapunov case, in order to derive conditions on when a good approximation by low rank factors can be achieved. Other approaches use LDL^T type representations with thin L and small square D, data sparse representations based on block low rank factorizations, such as $\mathcal{H}$-matrices, or even more sophisticated tensor structured forms like, e.g., tensor trains. We will briefly get back to these in Section 8.

An important question common to all iterative solver approaches is that of stopping the iteration. Usually the norm of the residual is the property of choice here. Concrete bounds for the smallness of the residual should be chosen carefully taking the data in the equation into account. When the right hand side has moderate size its norm may be used to normalize the residual norm. In case it is very small one should better take the norms of F and Z_i, or A, E and Z into account in backward error style instead, in the Lyapunov equation case. The same considerations should be undertaken with respect to the constant term in the Riccati equation case.

The remainder of this paper is structured as follows. We review early approaches to projection based solution of equations (2) and (3) in Section 3. Section 4 discusses the important class of extended Arnoldi based Krylov subspace projection methods. In Section 5 we introduce low rank ADI based solvers for both Lyapunov and Riccati equations. We dedicate Section 8 to some recent additions to the field and extensions of the methods discussed here to more general problem settings.

2 Low Rank Approximation of Solutions

The key to the successful solution of large scale Lyapunov and algebraic Riccati equations is to avoid forming the full solution matrix, as this is a usually dense $n \times n$ matrix. Though symmetric, for $n > 1000$, it becomes a challenging task to even store such a matrix, and, even worse, computing all $n(n+1)/2$ entries needs at least $\mathcal{O}(n^2)$ operations even if the coefficient matrices are sparse. Most current approaches rely on the low rank representation of solutions in the form $X \approx ZZ^H$ for a possibly complex factor Z with $k \ll n$ columns. Other possibilities have also been suggested and as already mentioned above, will briefly be discussed later in the paper.

In several contributions [46, 25, 2, 51, 59], it is shown that under certain assumptions, the eigenvalues of the solution to Lyapunov equations decay fast. This allows to approximate the solution by

$$X \approx X_k := \sum_{j=1}^{k} \lambda_j z_j z_j^H,$$

where $Xz_j = \lambda_j z_j$ with ordered eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n \geq 0$ and low-rank factor $Z = [\,\sqrt{\lambda_1} z_1, \ldots, \sqrt{\lambda_k} z_k\,]$ constructed using the dominant eigenvectors of X, scaled by the square roots of the corresponding eigenvalues. If k is small compared to n, then this is a very good low-rank approximation $X_k = ZZ^H$ satisfying the obvious error bound

$$\|X - X_k\|_2 \leq \lambda_{k+1},$$

if the decay is fast enough such that $\lambda_{k+1} \leq \tau$ for an acceptable error tolerance τ. Then the desired low-rank approximation with storage requirements of only nk words of memory is obtained. Of course, this is not a practical alternative as we would need to compute X first to determine its eigenvectors. Therefore, we will discuss in the next sections methods that compute low-rank factors Z by iterative processes, avoiding the forming of X. Here, we want to show one way to understand why good low-rank approximations can be expected. This yields an intuition more than a quantitative way of determining a practical estimate of a good low rank, but in contrast to some other approaches discussed in the literature, it extends without much ado to generalized situations as in Section 7.

We basically follow the approach discussed in [25]. The starting point, considering for ease of derivation a standard Lyapunov equation

$$AX + XA^T + BB^T = 0,$$

with A asymptotically stable and $B \in \mathbb{R}^{n \times m}$, is the solution formula (see, e.g., [3])

$$X = \int_0^\infty e^{At} BB^T e^{A^T t} \, dt.$$

Applying a suitable quadrature formula leads to the approximation

$$X \approx \sum_{j=1}^{k} \omega_j e^{At_j} BB^T e^{A^T t_j},$$

with quadrature points t_j and weights ω_j, where only quadrature formulas with positive weights should be chosen in order to obtain a real low-rank approximation ZZ^T. This approximation is obviously of rank km at most.

It is all but clear that this is in general a good approximation. But in [25] it is shown that if one chooses $k = 2K + 1$ sinc quadrature points with appropriate weights [55], one obtains an approximation satisfying

$$\|X - X_{(2K+1)m}\|_2 \lesssim \exp(-\pi\sqrt{K}) \tag{6}$$

with the square root of K replaced by K for symmetric A. We omit the quite technical exact statement from [25], yielding exact expressions for the involved constants. Here we only note that under mild assumptions, we can expect a good low-rank approximation if $m \ll n$.

3 Projection Methods for Solving Large Scale Matrix Equations

In this section we review the basic ideas of projection based solution of large scale matrix equations. We base the presentation on a general lower dimensional subspace of $\mathbb{R}^n$ and get into more detail for two special classes of subspaces in the next section.

Let $U \in \mathbb{R}^{n\times k}$ with $U^TU = I_k$ the identity matrix of dimension $k \times k$. Then the columns of U span a k dimensional subspace $\mathcal{U} \subset \mathbb{R}^n$ and $P_U = UU^T$ is the canonical orthogonal projection onto $\mathcal{U}$. The basic idea of projecting Lyapunov equations to such subspaces to get an approximation to the solution goes back to Saad [49]. It was picked up by Jaimoukah and Kasenally to formulate their Krylov subspace methods for solving large Lyapunov equations[30]. The idea of choosing a Krylov subspace was then further extended by Simoncini and co-authors to using extended and rational Krylov subspaces. The common approach in all these contributions is to solve a projected Lyapunov equation

$$U^TFUY + YU^TF^TU = -U^TGG^TU \tag{7}$$

instead of (3). Then Compute $Y = C^TC$ via the Cholesky or eigendecomposition of the projected solution Y and consider $Z = UC^T$ as the approximation of the Cholesky factor of the solution. In case the solution is not accurate enough, e.g., judging from the residual computed from inserting ZZ^T into (3), then an extension and/or update of the subspace $\mathcal{U}$ needs to be found to increase the quality of the approximate solution.

A common limitation to all projection based solvers is the requirement for the projected matrix U^TFU to remain Hurwitz (i.e., all eigenvalues lie in the open left half-plane) to guarantee solubility of (7). This is usually guaranteed by assuming that the matrix F fulfills $F + F^T < 0$ which employing Bendixon's theorem [41] is a sufficient condition for U^TFU being Hurwitz for any U as defined above.

Reminiscent of the analysis of the GMRES method for standard linear systems Mikkelsen[44] shows that the projection method resulting from the choice of the Arnoldi subspace for $\mathcal{U}$ may converge arbitrarily bad. In practice these methods have not been competitive with the low rank ADI or Smith type iterations presented in Section 5 until the investigation of the extended Arnoldi based approach by Simoncini [52].

The corresponding results for the algebraic Riccati equation have been worked out by Jbilou and co-authors in a series of papers since 2003[31, 32, 28], but the basic idea was also already treated in the paper by Jaimoukah and Kasenally [30]. The projected Riccati equation corresponding to (4) in complete analogy to (7) can be expressed as

$$\tilde{C}^T\tilde{C} + \tilde{A}^TY\tilde{E} + \tilde{E}^TY\tilde{A} - \tilde{E}^TY\tilde{B}\tilde{B}^TY\tilde{E} = 0, \tag{8}$$

where the coefficient and data matrices are defined as $\tilde{C} = CU$, $\tilde{B} = U^TB$, $\tilde{A} = U^TAU$, and $\tilde{E} = U^TEU$. However, the condition $F + F^T < 0$ relaxes here just as asymptotic stability is replaced by stabilizability for the solvability conditions in Section 1. That means we do not require the Hurwitz property for the symmetric part of $E^{-1}A$ itself, but for the corresponding stabilized closed loop matrices.

4 Extended and Rational Krylov Subspace Methods

Druskin and Knizherman [19] introduced extended Krylov subspaces – a combination of the Krylov subspaces $\mathcal{K}_m(F, G)$ and $K_m(F^{-1}, G)$ generated with respect to F and F^{-1} – as a new class of subspaces for the approximation of matrix function. The close relation of the solution X of (3) to the matrix exponential motivates the use of

Algorithm 1 Extended Krylov Subspace Method (EKSM)

```
Input: $E$, $A$, $B$ as in (2) with $E^{-1}A + A^TE^{-T} < 0$
Output: $Z \in \mathbb{R}^{n\times k}$ with $X \approx ZZ^T$ in (2)
if $E = I$ then
    Set $F = A$, $G = B$
else
    Compute Cholesky decomposition $LL^T = E$
    Set $F = L^{-1}AL^{-T}$, $G = L^{-1}B$
end if
$V_1 = \mathrm{orth}([G,\ F^{-1}G])$
$i = 2$, $U = V_1$
while ($i <$maxiter) do
    $F_i = U^TFU$ and $G_i = U^TG$
    Solve $F_iY_i + Y_iF_i^T = -G_i^TG_i$ for $Y_i$
    if (converged) then
        if ($E = I$) then
            $Z = U\,\mathrm{chol}(Y_i)$ and STOP
        else
            $Z = L^{-T}U\,\mathrm{chol}(Y_i)$ and STOP
        end if
    end if
    $V_{i+1} = \left[F\,U(:,2j-1),\ \ F^{-1}\,U(:,2j)\right]$
    Orthogonalize $V_{i+1}$ with respect to $U$
    Orthogonalize $V_{i+1}$ internally
    $U = [U,\ V_{i+1}]$
    i=i+1
end while
```

this combined subspace as the subspace $\mathcal{U}$ in the projection methods introduced in the previous section. Around 2006 Simoncini[52] came up with the idea of applying this exact subspace in her method that was initially known as KpiK (Krylov plus inverse Krylov) observing that the above subspace is equivalent to the space $\mathcal{K}_{2m}(F, F^{-m}G)$. For the case $G \in \mathbb{R}^n$, i.e., Lyapunov equations related to systems with a single input or output, it is summarized in Algorithm 1. In case Algorithm 1 is fed with a nontrivial symmetric positive definite (spd) E matrix, it performs a system transformation of (1), employing a Cholesky factor L of E, into the form

$$\begin{aligned}\dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{B}u(t),\\ y(t) &= \tilde{C}\tilde{x}(t) + Du(t),\end{aligned}$$

where $\tilde{x}(t) = Lx(t)$, $\tilde{A} = L^{-1}AL^{-T}$, $\tilde{B} = L^{-1}B$ and $\tilde{C} = CL^{-T}$. For the corresponding Lyapunov equations (2) we then find

$$\tilde{A}\tilde{P} + \tilde{P}\tilde{A}^T = -\tilde{B}\tilde{B}^T, \qquad \tilde{A}^T\tilde{Q} + \tilde{Q}\tilde{A} = \tilde{C}^T\tilde{C}, \tag{9}$$

where

$$\tilde{P} = L^T PL \text{ and } \tilde{Q} = L^T QL. \tag{10}$$

Algorithm 1 then in fact solves the first equation in (9) and performs the inverse transformation according to (10). Note that the spd restriction on E is taken only for ease of representation. For more general non-singular E it is simply replaced by an LU decomposition in the above. Note further that the transformation of the A matrix should never be performed explicitly forming $\tilde{A}$, or F, in the algorithm. Instead, the inversions of R should be performed as forward or backward solves employing a precomputed (and stored) LU decomposition, whenever the matrix is applied. As an alternative to the decomposition approach one can try to formulate the algorithm in terms of $E^{-1}A$ solving with E whenever the inverse would be required. This way one can avoid the extra memory consumption caused by Cholesky factors at the cost of loosing the easy residual recurrence.

Today the method is more often referred to as extended Krylov subspace method (EKSM) reflecting the origin of the spaces in [19]. The same name was also used in the corresponding article on the projection based solution of matrix Riccati equations (4) employing the same subspace [28]. That means equation (8) is treated with a matrix U whose columns span a subspace generated by an extended (block) Arnoldi process applied to the pair (A^T, C^T), i.e. $\mathcal{U} = \mathcal{K}_m(A^T, C^T)$ in the k-th step.

In the area of matrix functions (where [19] belongs) the extended Krylov subspace relates to a series expansion in frequency domain at frequencies 0 and ∞. The natural idea to increase the accuracy from that point of view is to add or use expansion points at intermediate frequencies. This relates to the projection onto rational Krylov subspaces, which is also investigated and compared to the ADI based approach in [20].

Stopping Criteria. One key ingredient for the efficiency of the EKSM method is the observation[52] that the residual norm in the k-th step can be computed via

$$\left|\left|FZZ^T + ZZ^TF^T + GG^T\right|\right| = \left|\left|G_k^TF_kY_k\right|\right|, \tag{11}$$

i.e., based only on small projected data and avoiding the explicit forming of the full residual, or even the factor Z. Unfortunately this formulation only applies to Lyapunov equations of the form (3) and can not be extended to those in (2). Thus the Cholesky decomposition of E in Algorithm 1 together with the additional memory requirements for R can not be avoided. Note further that R is used in the entire algorithm to avoid explicit forming of F, which would easily become dense. The treatment of non invertible E matrices can be found in [58]. However, only computable estimates to the residual norm have been proved so far in that case.

5 Low Rank Cholesky Factor ADI and Newton ADI

The second class of solvers we discuss is that of alternating directions implicit (ADI) based iterative methods. The core ADI iteration for an equation of the form (3) that

Algorithm 2 Low-rank Cholesky factor ADI iteration (LRCF-ADI)

```
Input: $E$, $A$, $B$ as in (2) and ADI shifts $p_i$, $i = 1, \dots,$ maxiter.
Output: $Z \in \mathbb{R}^{n \times k}$ with $P \approx ZZ^T$ in (2)
$Z_0 = []$
$i = 1$
while (not converged) and ($i <$maxiter) do
    if $i = 1$ then
        Solve $(A + p_1 E)V_1 = B$ for $V_1$.
    else
        Solve $(A + p_i E)\tilde{V} = EV_{i-1}$ for $\tilde{V}$.
        $V_i = V_{i-1} - (p_i + \overline{p_{i-1}})\tilde{V}$.
    end if
    if $p_i \in \mathbb{R}$ then
        $V_i = \mathrm{Re}\,(V_i)$.
        Update LRCF $Z_i = [Z_{i-1}\ \sqrt{-2p_i}V_i]$.
    else
        $\alpha = 2\sqrt{-\mathrm{Re}\,(p_i)}$, $\beta = \frac{\mathrm{Re}\,(p_i)}{\mathrm{Im}\,(p_i)}$.
        $V_{i+1} = \overline{V_i} + 2\beta\,\mathrm{Im}\,(V_i)$.
        Update LRCF $Z_{i+1} = \left[Z_{i-1},\ \alpha\left(\mathrm{Re}\,(V_i) + \beta\,\mathrm{Im}\,(V_i)\right),\ \alpha\sqrt{(\beta^2+1)}\cdot\mathrm{Im}\,(V_i)\right]$
        $i = i + 1$
    end if
    $i = i + 1$
end while
```

underlies all these methods is

$$\begin{aligned} X_0 &= 0 \\ (F + p_i I)X_{i-\frac{1}{2}} &= -GG^T - X_{i-1}(F^T - p_i I), \\ (F + \overline{p_i} I)X_i^T &= -GG^T - X_{i-\frac{1}{2}}^T(F^T - \overline{p_i} I). \end{aligned} \tag{12}$$

The parameters p_i here are the so called ADI shifts that have to be determined prior to the execution to accelerate the convergence. Some details on the choice are given below. The low rank Cholesky factor ADI (LRCF-ADI) iteration computes a symmetric rectangular factorization of the solution X. It is described briefly in Section 5.1. Algebraic Riccati equations like (4) are often solved using a Newton like iterative method due to Kleinman [33]. There in every step of the iteration, a Lyapunov equation is solved. The procedure employing the LRCF-ADI in these iteration steps and thus computing a low rank approximation of the solution is abbreviated NM-ADI and presented in Section 5.2. In both cases we discuss stopping criteria and recent variants of the algorithms trying to accelerate the solution process. Isolation of $X_{i-\frac{1}{2}}$ in the first equation and inserting it into the second leads to the one step update formula

$$\begin{aligned} X_i =& (F-\overline{p_i}I)(F+p_iI)^{-1}X_{i-1}(F-p_iI)^T(F+\overline{p_i}I)^{-T} \\ &- 2\operatorname{Re}(p_i)(F+\overline{p_i}I)^{-1}GG^T(F+p_iI)^{-T}. \end{aligned} \tag{13}$$

Thus, if X_{i-1} is real and symmetric so will be X_i. Especially, the symmetry of the update can be used to derive a low rank update formula [45, 12] by inserting $X_{i-1} = Z_{i-1}Z_{i-1}^H$ and $X_i = Z_iZ_i^H$

$$Z_i = \left[(F-\overline{p_i}I)(F+p_iI)^{-1}Z_{i-1},\ \sqrt{-2\operatorname{Re}(p_i)}(F+\overline{p_i}I)^{-1}G\right] \tag{14}$$

This update for the factor Z_i is the foundation of the algorithm discussed in Section 5.1.

Convergence of the Iteration. The iteration in (12) can be viewed as a double relaxation of a splitting method applied to the Lyapunov equation. Therefore it is not very surprising that the convergence result here comes in the form of a fixed point argument as well. The error reduction here basically takes place with respect to the matrix $W_J = \prod_{i=1}^J R_{p_i}$ for a shift vector $p \in \mathbb{R}^J$ and

$$R_{p_i} = (F-\overline{p_i}I)(F+p_iI)^{-1}(F+\overline{p_i}I)^{-T}(F-p_iI)^T.$$

The acceleration of the worst case convergence of the iteration can thus be expressed in terms of the spectral radius of W_J and minimized by a clever choice of the elements of p, i.e., the ADI shifts p_i in the rational min-max-problem[61, 62]

$$\min_{\substack{p_i \in \mathbb{C}_{<0},\\ i=1,\dots,J}} \max_{\lambda\in\Lambda(F)} \prod_{i=1}^J \frac{|p_i-\lambda|^2}{|p_i+\lambda|^2}.$$

5.1 LRCF-ADI

As mentioned above, we are interested in low rank factored representations of the solution and we would like to compute the factors successively. After some further manipulation of equation (14) [36], one finds that in fact only the new columns in the low rank factor update have to be processed, instead of all columns as in the naive approach. The extension of the resulting algorithm to the case of systems (1) with $E \neq I$ but regular is then straightforward (see, e.g., [5, 50]) by applying the aforementioned steps to $F = E^{-1}A$ and avoiding the inverses in the steps of the algorithm. This procedure together with a recent strategy to guarantee real low rank factors [10] results in Algorithm 2. The case of singular E is discussed in Section 7.1.

Variants. Over the recent years some variants of the above algorithm have been proposed. Most of them are slight modifications to the above algorithm to exploit special problem structures or improve performance where necessary. A performance increase can for example be found in some cases, when a column compression step for the factor Z_i is added prior to the evaluation of the stopping criteria. This especially helpful in the context of Lyapunov equations that arise in each time step of Rosenbrock solvers applied to differential Riccati equations (e.g., [43]), since there already the right hand side factor may contain linearly dependent columns. Note that also the EKSM or RKSM Methods easily allow for such a compression replacing the Cholesky decomposition of Y_i by an SVD or eigendecomposition approach and truncation by the magnitude or the singular or eigenvalues respectively. A method that takes the ADI iteration into the context of those solvers was described in [14]. There the rational Krylov subspace formed by the columns of the solution factor during the ADI iteration is used to perform a projection step as in the projection methods to improve the solution. In that sense this method should be regarded as a projection method employing a very special rational Krylov subspace. However, in their performance analysis Simoncini and co-authors [20] prove that RKSM always performs at least as good as this algorithmic variant of LRCF-ADI.

Stopping Criteria. The two most common stopping criteria for the iteration in Algorithm 2 are based on monitoring either the relative change of the factor Z, i.e.,

$$\begin{aligned} p_i \in \mathbb{R} : \mathsf{rc}_i &= \frac{\left|\left|\sqrt{-2p_i}V_i\right|\right|_F}{||Z_i||_F}, \\ p_i \in \mathbb{C}\backslash\mathbb{R} : \mathsf{rc}_i &= \alpha\frac{\left|\left|\left[\mathrm{Re}\,(V_i) + \beta\,\mathrm{Im}\,(V_i),\ \sqrt{(\beta^2+1)}\cdot \mathrm{Im}\,(V_i)\right]\right|\right|_F}{||Z_i||_F}, \end{aligned}$$

or the residual of the current iterate $\mathcal{L}(Z_i) := FZ_iZ_i^TE^T + EZ_iZ_i^TA^T + GG^T$ for smallness.

The relative change is advisable to be measured in the Frobenius norm since there $||Z_i||_F$ can be accumulated from the enumerators.

Only recently a low rank representation of the residual has been derived in [11]. Exploiting the equivalence of the Lyapunov equation to a Stein equation (compare Section 6.1) in the same way it is done in [29] to prove the definiteness of the residual for all iterates in the process, it can be observed to be of the following low rank structure:

$$\mathcal{L}(Z_i) = (F - \overline{p_i}E)V_iV_i^H(F - \overline{p_i}E) =: \hat{V}_i\hat{V}_i^H. \tag{15}$$

Here the shifted multiplications are remaining from the reverse transformation from Stein equation form. It can further be shown that

$$\hat{V}_i = \hat{V}_{i-1} - 2\,\mathrm{Re}\,(\mu_i)EV_i$$

providing an easy update formula which saves the additional shifted matrix vector product.

From this representation for both the spectral and Frobenius norms it immediately follows

$$||\mathcal{L}(Z_i)|| = \left|\left|\hat{V}_i\hat{V}_i^H\right|\right| = \left|\left|\hat{V}_i^H\hat{V}_i\right|\right| = \left|\left|\hat{V}_i\right|\right|^2.$$

Note that in case projections are used to accelerate the iteration, the residual formula (15) is no longer valid. In that case $\hat{V}_i$ would have to be projected onto the orthogonal complement of the space to which the projection was performed. Unfortunately that is not possible efficiently. A Frobenius norm computation based on QR-factorization updates was proposed in the LyaPack software package by Penzl [47].

Alternatively one can employ the spectral norm. Due to symmetry and definiteness (see, e.g., [29]) of the residual it coincides with the largest magnitude eigenvalue and one may use a Lanczos method to get a good approximation quickly.

5.2 NM-ADI

The Newton iteration applied to solving the algebraic Riccati equation received only minor attention until around the year 2000. Until then it was mainly considered an iterative refinement technique used to increase the accuracy of a solution acquired by a direct, invariant subspace based method for solving the equation with dense coefficient matrices. With the ability to solve large and sparse Lyapunov equations, however, it became the method of choice for the solution of large scale Riccati equations. The version of the Newton iteration that is used in the NM-ADI is due to Kleinman [33]. Its main advantage over the classic Newton method is the greatly simplified right hand side in the Lyapunov equation that has to be solved in each step.

The following paragraph summarizes the origins of the iteration and derives the structure of the Lyapunov equations resulting from the Kleinman reformulation. Often the solution of the Riccati equation is solved in order to compute the optimal feedback in a linear quadratic optimal control problem. Then the ADI based solution of the Lyapunov equations in the Newton steps allows for a reformulation that avoids the computation of solution factors and instead only implicitly uses them to form successively improved approximations of the optimal feedback. This procedure will be sketched, together with other variants of the basic iteration, thereafter. The same reformulation trick can be employed to implement an inexact Kleinman-Newton method following the theory developed in [29, 21], which will also be reviewed shortly. In the final paragraph of this section we get back to the problem of stopping the iteration.

Kleinman-Newton-Formulation. Consider the ARE (4) and define the left hand side as $\mathfrak{R}(X)$. Then the ℓ-th basic Newton iteration step can be formulated as

$$\mathfrak{R}'|_X(N_\ell) = -\mathfrak{R}(X_\ell), \qquad X_{\ell+1} = X_\ell + N_\ell. \tag{16}$$

where

$$\mathfrak{R}'|_X : \quad N \mapsto (A - BR^{-1}B^TX)^TNE + E^TN(A - BR^{-1}B^TX), \tag{17}$$

is the Frechét derivative of $\mathfrak{R}$. Kleinman's contribution now was the reformulation of the step such that it does not provide the update N_ℓ of the iterate, but the new iterate itself. In other words he uses $\mathfrak{R}'(N_\ell) = \mathfrak{R}'(X_{\ell+1} - X_\ell) = \mathfrak{R}'(X_{\ell+1}) - \mathfrak{R}'(X_\ell)$ to derive

$$\begin{aligned}(A^T - K_{\ell-1}B^T)X_{\ell+1}E + E^TX_{\ell+1}(A - BK_\ell{}^T) &= -C^TC - K_\ell K_\ell{}^T \\ &= -[C^T, K_\ell][C^T, K_\ell]^T.\end{aligned} \tag{18}$$

Algorithm 3 Low-rank Kleinman-Newton-ADI iteration (NMADI)

```
Input: $E$, $A$, $B$, $C$ as in (4) and an initial guess $K_0$ for the feedback.
Output: $X_\infty$ solving (4) and the optimal state feedback $K_\infty$ (or approximations when
    stopped before convergence).
for $k = 1, 2, \ldots$ do
    $F_k = A - BK_{k-1}^T$
    $G_k = [C^T,\ K_{k-1}]$
    Choose a set of ADI shift parameters with respect to $F_k$.
    Determine the solution factor $Z_k$ of the solution $X_k$ of
        $$F_k^T X_k E + E^T X_k F_k = G_k G_k^T$$
    by Algorithm 2.
    $K_k = (E^T Z_k)(Z_k^T B)$.
end for
```

This obviously has the additional advantage of the simplified right hand side, which especially can be written in low rank format. The latter observation shows that in fact the step equation is of the form (3) with a low rank updated sparse matrix as the coefficient F, which we call splr (for *sparse plus low rank* following [50, Definition 4.2]). This equation can now be solved by any of the methods for large and sparse Lyapunov equations described above. Here the ADI has certain advantages when only the feedback gain matrix and not the actual solution of the ARE is of interest, as we will see in the next paragraph. The application of Krylov subspace based solvers is investigated especially for the inexact Kleinman-Newton case recently in a technical report by Simoncini, Szyld and Monsalve in [53].

Whenever linear systems with an splr F or a shifted splr F need to be solved, the Sherman-Morrison-Woodburry (SMW) formula (e.g. [24])

$$(M + UV^T)^{-1} = M^{-1} - M^{-1}U(I + V^T M^{-1}U)^{-1}V^T M^{-1}, \tag{19}$$

for a sparse matrix M and thin rectangular blocks U and V, is applied to avoid explicit forming of the dyadic product. In the special case of $F = A^T - K_{(\ell-1)}B^T$ this means solving a linear system with F (or $F + p_i I$) requires two solves with A (or $A - p_i I$) and an additional small linear solve with an $m \times m$ matrix.

A major difficulty of the Newton procedure for initially unstable systems is the need for an initial stabilizing feedback K_0, such that $(A^T - K_0 B^T)$ is Hurwitz, in order to guarantee solvability of the Lyapunov equation (18). The task of computing this stabilizing initial feedback K_0 is numerically challenging itself. For dense problems (partial) stabilization methods based on pole placement or solving certain Lyapunov equations have been existing in the literature for years. Their extension to the large and sparse case is considered, e.g., in [1, 23, 48, 4]. An alternative approach that has been used for distributed parameter systems is given by the Chandrasekhar iteration[16]. The Bernoulli equation based partial stabilization technique in [6] is especially attractive when the unstable eigenvalues are known, or easy to find, together with their eigenspaces and when their number is very small in comparison to n.

The basic Kleinman-Newton-ADI procedure is summarized in Algorithm 3. In the next paragraph we will discuss some variants of this iteration that can save some computation time in certain situations.

Variants. One of the most important observations when using ADI as the inner iteration for the Kleinman-Newton process is that due to the way the solution factor is formed in the low rank ADI iteration, the feedback approximation can be accumulated without ever storing the entire solution factor. Recall that the low rank ADI iteration successively adds new column blocks to the factor (e.g., $Z_k^{(i)} = [Z_k^{(i-1)}, \sqrt{-2p_i}V_i]$ as in Step 12 of Algorithm 2). Then for the feedback update we have

$$\begin{aligned} K_k^{(i)} &= E^T Z_k^{(i)} {Z_k^{(i)}}^T B = E^T \left(Z_k^{(i-1)} {Z_k^{(i-1)}}^T - 2p_i V_i V_i^T \right) B \\ &= E^T Z_k^{(i-1)} {Z_k^{(i-1)}}^T B - 2p_i E^T V_i V_i^T B \\ &= K_k^{(i-1)} - 2p_i E V_i V_i^T B. \end{aligned}$$

This observation does not only lead to an implicit Kleinman-Newton-ADI iteration directly iterating on the feedback, but also helps formulating an inexact Kleinman-Newton-ADI iteration controlling the accuracy of the inner iteration to further reduce the execution time.

Since the update of the Frechét derivative is mainly given for free in the context of Kleinman-Newton, the way of forming a simplified Newton type iteration is achieved by freezing the ADI shifts for a couple of steps. If the closed loop matrix F_k did not change very much, the loss in convergence speed for the ADI is easily compensated by the time saved in skipping the parameter computation. However this approach should be used with care since it can in the worst case increase the total execution time when F_k changes a lot from step to step.

An idea that is often used to optimize execution times in Newton type iterations is that of a line search for determining the best step length. In the large scale Newton-ADI setting this idea showed to be too expensive to provide any gains in execution time however. On the other hand, getting away form a preprocessed optimization with respect to a one dimensional subspace and instead applying a post processing Galerkin projection step as described in Section 3 for the Lyapunov case, one can extend the optimization to an even higher dimensional subspace. As the projection basis one can use the span of the current solution factor as described before. Numerical results in [14] show that this can easily lead to the Newton iteration being stopped after just one step.

Inexact Kleinman-Newton. As a consequence of what we have seen before (18) the inexact Kleinman-Newton step can be written in the form

$$R_\ell = \mathfrak{R}'|_{X_\ell}(X_{\ell+1}) - \mathfrak{R}'|_{X_\ell}(X_\ell) + \mathfrak{R}(X_\ell) = \mathfrak{R}'|_{X_\ell}(X_{\ell+1} - X_\ell) + \mathfrak{R}(X_\ell), \qquad (20)$$

where R_ℓ denotes the inner residual (i.e., the Lyapunov/ADI residual) representing the inexactness allowed for the solution of the inner iteration. Exploiting the expansion (see, e.g., [29])

$$\mathfrak{R}(Y) = \mathfrak{R}(X) + \mathfrak{R}'|_X(Y-X) + \frac{1}{2}\mathfrak{R}''|_X(Y-X, Y-X),$$

and comparing terms with (20) we can derive an expression for the Riccati residual

$$\begin{aligned}\mathfrak{R}(X_{\ell+1}) &= R_\ell + \frac{1}{2}\mathfrak{R}''|_{X_\ell}(X_{\ell+1}-X_\ell, X_{\ell+1}-X_\ell)\\ &= R_\ell + \frac{1}{2}\mathfrak{R}''|_{X_\ell}(N_\ell, N_\ell) = R_\ell - \frac{1}{2}E^TN_\ell BB^TN_\ell E,\end{aligned} \tag{21}$$

in terms of the inner residual and the change of the feedback gain matrix

$$\begin{aligned}E^TN_\ell BB^TN_\ell E &= E^T(X_{\ell+1}-X_\ell)BB^T(X_{\ell+1}-X_\ell)E\\ &= X_{\ell+1}BB^TX_{\ell+1} + X_\ell BB^TX_\ell\\ &\quad - X_\ell BB^TX_{\ell+1} - X_{\ell+1}BB^TX_\ell\\ &= K_{\ell+1}^TK_{\ell+1} + K_\ell^TK_\ell - K_{\ell+1}^TK_\ell - K_\ell^TK_{\ell+1}\\ &= (K_{\ell+1}-K_\ell)^T(K_{\ell+1}-K_\ell).\end{aligned}$$

Since the inner residual is monitored in the inner iteration anyway, and the feedback gain matrix can be successively accumulated, this allows us to steer the accuracy of the inner iteration. Note that (21) especially shows us that the Riccati residual is of rank at most $(m+p)+m = 2m+p$ following from the rank of the right hand side in (18) (together with the result in (15)) and the observation above. Similar results for the Riccati residual have been derived from the Krylov subspace projection perspective in [53].

Stopping Criteria. The arguments regarding equation (15) for the Lyapunov case together with the expression in (21) allow us to directly extend the cheap evaluation of residuals to the Riccati case. Again, this can only be used when Galerkin projection is not applied. Also here, however, the residual can cheaply be approximated by a few steps of Lanczos algorithm due to symmetry.

In the case of the inexact Kleinman-Newton approach equation (21) can be used to guarantee the validity of the conditions [29, 21]

$$0 \leq R_\ell \leq C^TC \qquad \text{and} \qquad 0 \leq R_\ell \leq E^TN_\ell BB^TN_\ell E,$$

that ensure convergence towards the stabilizing solution. Here the semi-definiteness of R_ℓ is a direct consequence of the derivation of (15). Alternatively the quadratic convergence can be enforced via

$$\begin{aligned}\|\tilde{\mathfrak{R}}(X_{\ell+1})\|_2 &\leq \gamma\left(\|R_\ell\|_2 + \frac{1}{2}\|[K_{\ell+1}-K_\ell]^T[K_{\ell+1}-K_\ell]\|_2\right)\\ &\leq \varepsilon_\ell := \alpha\tilde{\mathfrak{R}}(X_\ell)^2\end{aligned}$$

for an $\alpha < 1$, $\gamma = \frac{1}{\|C^TC\|_2}$ and $\tilde{\mathfrak{R}}(.) = \gamma\mathfrak{R}(.)$ the normalized residual.

6 Doubling Based Approaches

In a sequence of papers Chu and co-workers have introduced doubling based algorithms for all kinds of large scale matrix equations. Their contributions for the continuous time Lyapunov and Riccati equations can be found in [39] and [37]. We briefly recall their main ideas here.

6.1 Doubling for Large Scale Lyapunov Equations

Basically the doubling approach for large scale Lyapunov equations boils down to a variant of the Smith method[54] for solving the equivalent Stein equation generated through Cayley transformation for a positive, real shift μ,

$$X = F_\mu^T X F_\mu + G_\mu T_\mu G_\mu^T, \tag{22}$$

where $F_\mu = (F + \mu I)(F - \mu I)^{-1} = I + 2\mu F$, $G_\mu = (F - \mu I)^{-T} G$, and $T_\mu = -2\mu I$. Note that, due to the assumptions on F, here F_μ is d-stable, i.e., for the spectral radius of F_μ we have $\varrho(F_\mu) < 1$. Successively inserting the right hand side in (22) into itself and exploiting that $F_\mu^{2^k} \to 0$ quadratically as $k \to \infty$, one finds (see [39] for details) that

$$X = \lim_{k\to\infty} H_k, \qquad \text{where} \qquad H_k = \sum_{i=0}^{2^k-1} (F_\mu^i)^T G_\mu T_\mu G_\mu^T F_\mu^i. \tag{23}$$

The basic recurrence for the iteration then is

$$\begin{aligned} H_{k+1} &= H_k + F_{\mu,k}^T H_k F_{\mu,k} = G_{\mu,k+1} T_{k+1} G_{\mu,k+1}^T, \\ G_{\mu,k+1} &= [G_{\mu,k},\, F_{\mu,k}^T G_{\mu,k}], \\ T_{k+1} &= T_k \oplus T_k = \begin{bmatrix} T_k & 0 \\ 0 & T_k \end{bmatrix}, \end{aligned} \tag{24}$$

with initial values

$$F_{\mu,0} = F_\mu, \quad G_{\mu,0} = G_\mu, \quad T_0 = T_\mu, \quad H_0 = G_{\mu,0} T_0 G_{\mu,0}^T = G_\mu T_\mu G_\mu^T.$$

Obviously the number of columns in $G_{\mu,j}$ and thus the size of T_j doubles in every iteration step. To limit the memory demand and keep the iteration computationally efficient, the authors introduce an additional rank truncation strategy, which they claim to be the major advantage as compared to older Smith type iterations, as discussed, e.g. in [45].

6.2 Doubling for Large Scale Riccati Equations

The general idea in the case of continuous time algebraic Riccati equations (4) is essentially the same. First a transformation into a discrete time algebraic Riccati equation

$$X = A^T X A - A^T X B (I + B^T X B)^{-1} B^T X A + H \tag{25}$$

or equivalently (employing the SMW formula (19) with $G = BB^T$)

$$X = A^T X (I + GH)^{-1} A + H \tag{26}$$

is performed via Cayley transformation, as in the Lyapunov case. Then the actual doubling algorithm is performed on the resulting discrete time matrix equation as above. We refer to the original papers [38, 37] for the details, since their derivation is to involved to include a compact version in the presentation at hand.

7 Generalized Linear Matrix Equations

In this section we discuss two important generalizations of the linear matrix equations above that can be found in the literature. On the one hand we treat systems (1) with rank deficiency in the E matrix, i.e., differential algebraic equation (DAE) systems, on the other hand, we touch the case of linear stochastic and bilinear systems which both lead to the same generalization of (3).

7.1 Projected Lyapunov Equations Related to DAE systems

The main contributions to this area are due to Stykel. They can be found, e.g., in [42] and references therein. Following the presentation there, we assume that even if the matrix E is singular the pencil $(\lambda E - A)$ is regular. That means we can always find a $\lambda \in \mathbb{C}$ such that $\det(\lambda E - A) \neq 0$. Then (see, e.g.[56]), the pencil can be written in *Weierstrass canonical form*

$$E = U \begin{bmatrix} I_{n_f} & 0 \\ 0 & N \end{bmatrix} V, \qquad \text{and} \qquad A = U \begin{bmatrix} J & 0 \\ 0 & I_{n_\infty} \end{bmatrix} V, \tag{27}$$

where J is in *Jordan canonical form* and N is nilpotent. The nilpotency index of N can be used to define the *nilpotency index* of the DAE. In the case of linear systems with constant coefficients this concept coincides with the *differentiation index* describing the number of times the system, or parts of it, need to be differentiated to result in an ordinary differential equation system. The numbers n_∞ and n_f describe the dimensions of the deflating subspaces corresponding to the infinite or finite eigenvalues of the pencil. The transformation matrices in (27) can now be used to define the left and right spectral projection matrices

$$\Pi_l = U^{-1} \begin{bmatrix} I_{n_f} & 0 \\ 0 & 0 \end{bmatrix} U \qquad \text{and} \qquad \Pi_r = V \begin{bmatrix} I_{n_f} & 0 \\ 0 & 0 \end{bmatrix} V^{-1} \tag{28}$$

mapping to the left and right deflating subspaces corresponding to the finite eigenvalues. The two Lyapunov equations of interest when extending Balanced Truncation model order reduction to the DAE system, together with the appropriate invariance conditions for their solutions, can then be expressed as

$$\begin{aligned} AP_pE^T + EP_pA^T &= -\Pi_l BB^T \Pi_l^T, & P_p &= \Pi_r P_p \Pi_r^T, \\ A^T Q_p E + E^T Q_p A &= -\Pi_r^T C^T C \Pi_r, & Q_p &= \Pi_l^T Q_p \Pi_l. \end{aligned} \tag{29}$$

Spectral Projection Based Low Rank ADI for Projected Lyapunov Equations. For the types of equations introduced above Stykel [57], roughly speaking, formulates Algorithm 2 for $F = A^{-1}E$ with shifts $\mu_i = \frac{1}{p_i}$ assuming that A is regular when E is not. The positive observation for the resulting algorithm is that it automatically fulfills the additional invariance conditions for each iterate (and thus the final factor) during the process once the initial right hand sides have been projected properly. The major drawback is the necessity for the spectral projectors to the finite spectrum of the pencil. These are in general not easy to obtain. Therefore, other authors have developed variants of the LRCF-ADI that avoid these projections.

Alternative Approaches Avoiding the Spectral Projections. For simple index-1 systems the authors of [22] show that the index reduction can be performed implicitly. The key ingredient in their idea is that the index-1 system can always be written in the form

$$\begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix} \dot{x}(t) = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} x(t) + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t),$$

with an invertible A_{22} matrix. Then they formulate the ADI iteration with respect to the Schur complement in A and show that the inversion of A_{22} can in fact be avoided when solving the shifted linear systems by undoing the Schur complement. This way the whole algorithm can be formulated in terms of the original matrices and the index reduction performed by the Schur complement is never executed explicitly. A similar approach, implicitly projecting to the *hidden manifold* describing the solution set of the DAE system, is pursued for index-2 systems of Stokes-like block structure

$$\begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix} \dot{x}(t) = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & 0 \end{bmatrix} x(t) + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t),$$

in [27]. Although technically more involved, the basic structure of the approach is very similar. First the special structure of the Stokes-like equation is exploited to form the oblique projection matrix

$$\Pi^T = I - E_{11}^{-1} A_{12} (A_{12}^T E_{11}^{-1} A_{12})^{-1} A_{12}^T,$$

onto the aforementioned hidden manifold, which in the Stokes case coincides with the discrete Leray projection onto the space of divergence-free functions, and the corresponding projected system. Note that Π is in fact symmetric in the E_{11}-inner product.

Then the algorithm is formulated based on the projected, index reduced (ordinary differential equation) system and finally the equivalence of the projected shifted linear systems that need to be solved in the ADI step to certain saddle point systems involving the original problem structure is shown. Exploiting this equivalence to avoid forming of the projected systems and projectors in the final algorithm enables the implicit index reduction here as well.

EKSM for the Projected Lyapunov Equations. In the case of the EKSM for projected Lyapunov equations [58], the most critical question is which construction is replacing the matrix F in Algorithm 1. Due to non-invertibility of E neither E^{-1}, nor the inverses of its Cholesky factor can be formed. Also the approach employing $F = A^{-1}E$ used by Stykel in the ADI case is not straight forward. in the EKSM one needs to invert F which is still not possible. Fortunately the inverse can be replaced by a proper pseudo inverse to make the algorithm work again. The pseudo inverse of choice here is the Drazin inverse [17] $F^D = E^-A$ employing the reflexive generalized inverse

$$E^- = V^{-1}\begin{bmatrix} I_{n_f} & 0 \\ 0 & 0 \end{bmatrix} U^{-1},$$

with U, V, and I_{n_f} from (27). For the stopping criteria the method exploits the equivalence of the first equation in (29) to

$$E^-AP_p + P_p(E^-A)^T = -E^-BB^T(E^-)^T, \qquad P_p = \Pi_r P_p \Pi_r^T,$$

terminating when the normalized equivalent residual

$$\frac{\|E^-R_k(E^-)^T\|_F}{\|E^-BB^T(E^-)^T\|_F} \leq tol.$$

Here R_k represents the original residual inserting the current iterate in (29). The equivalence is exploited since $\|E^-R_k(E^-)^T\|_F$ can be represented in a similar way as (11) and thus R_k can be estimated by the same cheaply computable expression with $\|E\|_F^2$ as the proportionality factor.

7.2 Lyapunov-plus-positive Equations

A different generalization of the Lyapunov equation (3) consists in adding one or more positive terms $\sum_{k=1}^{\ell} N_k X N_k^T$ (with $N_k \in \mathbb{R}^{n\times n}$) so that one obtains

$$FX + XF^T + \sum_{k=1}^{\ell} N_k X N_k^T = -GG^T. \tag{30}$$

This linear matrix equation is called *Lyapunov-plus-positive equation* for obvious reasons. Positivity here means that the operator $X \to N_k X N_k^T$ preserves positive semidefiniteness. This equation arises in control and model reduction of bilinear systems of the form

$$\begin{aligned} \dot{x}(t) &= Fx(t) + \sum_{k=1}^{m} N_k x(t) u_k(t) + Bu(t), \\ y(t) &= Cx(t), \end{aligned} \tag{31}$$

with $F \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $u(t) = [u_1(t), \ldots, u_m(t)]^T \in \mathbb{R}^m$, as well as in linear stochastic Itô-type systems driven by Wiener processes, see [9] and references therein. In the bilinear case (31), where $G \equiv B$, the number of terms in the sum in (30) obviously equals the number of inputs and therefore $\ell = m$, while in the stochastic setting, ℓ denotes the number of independent Wiener processes driving the system. In both cases, under certain assumptions, the reachability and observability Gramians of the systems are given as the solutions of equations of the form (30). It is therefore natural to ask how these equations can be solved in a large-scale setting as considered here.

Direct and iterative procedures to solve (30) for the full X are discussed in [18], while in [8] it is shown that for many applications, one can expect good low-rank approximations to X by adapting the bound (6) to the situation considered here. Though the assumptions made for proving the existence of low-rank approximations to X (low-rank structure of all N_k or commutativity of the N_k and F) are restrictive, solutions to (30) often exhibit numerical low-rank properties even if these assumptions are violated. The latter fact requires further investigation. Based on the assumed low-rank property, several strategies how such low-rank approximations can be computed are discussed in [8]. This includes variants of the ADI method discussed in Section 5 as well as the extended and rational Krylov subspace methods reviewed in Section 4. For (30), it appears to be more effective, though, to use adapted variants of the preconditioned Krylov subspace solvers with truncation discussed in [34]. The experiments reported in [8] indicate that a promising preconditioner is derived based on using a low, fixed number of ADI steps (using the ADI variant adapted to (30)). Certainly, other methods like the ones discussed below can be applied to Lyapunov-plus-positive equations as well.

The generalization of the available approaches for solving (30) to cases including a nonsingular mass matrix, resulting in a generalized Lyapunov part $AXE^T + EXA^T$ instead of $FX + XF^T$, is straightforward and can be treated similarly as in the generalized Lyapunov case. The treatment of such an equation with singular E remains an open problem, though.

8 Other Recent Approaches and Extensions

In this final section of the paper we want to point out some related methods that have appeared over the recent years. In their *SIAM Outstanding Paper Prize* awarded contribution [60] Vandereycken and Vandewalle propose an optimization on manifolds based approach to finding a low rank solution of (3). Although their method is often outperformed by the projection and ADI based methods it opens an interesting new perspective. This in turn enabled the authors in [7] to prove the optimality of the final IRKA poles as ADI shifts in view of the optimization on manifolds. In fact it was shown that applying the final IRKA poles (also called H_2-shifts) the ADI iteration, the RKSM and the optimization on manifolds approach are equivalent in the sense that they are computing the same solution factor Z.

Eppler and Bollhöfer in a series of conference papers derived a flexible generalized minimal residual (FGMRES) type iteration with ADI preconditioning [15]. The main feature that motivates the usage of FGMRES is its flexibility with respect to the preconditioner that is allowed to change in every step. The method replaces the standard matrix-vector and vector-update operations as well as inner products by their equivalents for LDL^T type low rank factorizations. A rank truncation framework completes the picture. In this view it is clear that their approach has to be seen as close relative of the Krylov subspace methods applied to more generally tensor structured equations as for example treated by Kressner and Tobler [34] that have shown in [8] to be effective also in the Lyapunov-plus-positive case discussed above. It is noteworthy that with [34], yet another paper in the area of numerical methods for matrix equations won a *SIAM Outstanding Paper Prize* (awarded 2013).

An extension of the ADI framework to tensor structured equations was discussed in [40]. Grasedyck [26] contributed a nonlinear multigrid based approach that computes approximations to the solution of Riccati equations in either low rank or $\mathcal{H}$-matrix format.

References

[1] L. Amodei and J.-M. Buchot, *A stabilization algorithm of the Navier-Stokes equations based on algebraic Bernoulli equation*, Numer. Lin. Alg. Appl., 19 (2012), pp. 700–727.

[2] A. Antoulas, D. Sorensen, and Y. Zhou, *On the decay rate of Hankel singular values and related issues*, Sys. Control Lett., 46 (2002), pp. 323–342.

[3] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM Publications, Philadelphia, PA, 2005.

[4] H. Banks and K. Ito, *A numerical algorithm for optimal feedback gains in high dimensional linear quadratic regulator problems*, SIAM J. Cont. Optim., 29 (1991), pp. 499–515.

[5] P. Benner, *Solving large-scale control problems*, IEEE Control Systems Magazine, 14 (2004), pp. 44–59.

[6] P. Benner, *Partial stabilization of descriptor systems using spectral projectors*, in Numerical Linear Algebra in Signals, Systems and Control, P. Van Dooren, S. P. Bhattacharyya, R. H. Chan, V. Olshevsky, and A. Routray, eds., vol. 80 of Lecture Notes in Electrical Engineering, Springer Netherlands, 2011, pp. 55–76.

[7] P. Benner and T. Breiten, *On optimality of interpolation-based low-rank approximations of large-scale matrix equations*, Preprint MPIMD/11-10, Max Planck Institute Magdeburg, December 2011. Available from http://www.mpi-magdeburg.mpg.de/preprints/.

[8] ——, *Low rank methods for a class of generalized Lyapunov equations and related issues*, Numer. Math., (2013, to appear).

[9] P. Benner and T. Damm, *Lyapunov equations, energy functionals, and model order reduction of bilinear and stochastic systems*, SIAM J. Cont. Optim., 49 (2011), pp. 686–711.

[10] P. Benner, P. Kürschner, and J. Saak, *Efficient Handling of Complex Shift Parameters in the Low-Rank Cholesky Factor ADI Method*, Numerical Algorithms, 62 (2013), pp. 225–251. 10.1007/s11075-012-9569-7.

[11] ——, *An improved numerical method for balanced truncation for symmetric second order systems*, Math. Comput. Model. Dyn. Sys., (online since 01 May 2013). DOI: 10.1080/13873954.2013.794363.

[12] P. Benner, J.-R. Li, and T. Penzl, *Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems*, Numer. Lin. Alg. Appl., 15 (2008), pp. 755–777.

[13] P. Benner, V. Mehrmann, and D. Sorensen, eds., *Dimension Reduction of Large-Scale Systems*, vol. 45 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

[14] P. Benner and J. Saak, *A Galerkin-Newton-ADI Method for Solving Large-Scale Algebraic Riccati Equations*, Preprint SPP1253-090, DFG Priority Programme 1253 Optimization with Partial Differential Equations, 2010. Available from http://www.am.uni-erlangen.de/home/spp1253/wiki/images/2/28/Preprint-SPP1253-090.pdf.

[15] M. Bollhöfer and A. K. Eppler, *A structure preserving FGMRES method for solving large Lyapunov equations*, in Progress in Industrial Mathematics at ECMI 2010, M. Günther, A. Bartel, M. Brunk, S. Schöps, and M. Striebel, eds., , Mathematics in Industry, Springer Berlin Heidelberg, 2012, pp. 131–136.

[16] J. A. Burns, K. Ito, and R. K. Powers, *Chandrasekhar equations and computational algorithms for distributed parameter systems*, in Proc. 23rd IEEE Conference on Decision and Control., Las Vegas, NV, Dec. 1984.

[17] S. L. Campbell and C. D. Meyer, *Generalized Inverses of Linear Transformations*, no. 56 in Classic in Applied Mathematics, SIAM Publications, 2009.

[18] T. Damm, *Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations*, Numer. Lin. Alg. Appl., 15 (2008), pp. 853–871.

[19] V. Druskin and L. Knizhnerman, *Extended Krylov subspaces: Approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 755–771. DOI: 10.1137/S0895479895292400.

[20] V. Druskin, L. Knizhnerman, and V. Simoncini, *Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation*, SIAM J. Numer. Anal., 49 (2011), pp. 1875–1898.

[21] F. Feitzinger, T. Hylla, and E. W. Sachs, *Inexact Kleinman-Newton Method for Riccati Equations*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 272–288.

[22] F. Freitas, J. Rommes, and N. Martins, *Gramian-Based Reduction Method Applied to Large Sparse Power System Descriptor Models*, IEEE Trans. Power Systems, 23 (2008), pp. 1258–1270.

[23] K. Gallivan, X. Rao, and P. VanDooren, *Singular riccati equations stabilizing large-scale systems*, Linear Algebra Appl., 415 (2006), pp. 359–372.

[24] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, third ed., 1996.

[25] L. Grasedyck, *Existence of a low rank or H-matrix approximant to the solution of a Sylvester equation*, Numer. Lin. Alg. Appl., 11 (2004), pp. 371–389.

[26] ——, *Nonlinear multigrid for the solution of large-scale Riccati equations in low-rank and $\mathcal{H}$-matrix format*, Numer. Lin. Alg. Appl., 15 (2008), pp. 779–807.

[27] M. Heinkenschloss, D. Sorensen, and K. Sun, *Balanced truncation model reduction for a class of descriptor systems with applications to the oseen equations*, SIAM J. Sci. Comput., 30 (2008), pp. 1038–1063.

[28] M. Heyouni and K. Jbilou, *An extended block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation*, Electr. Trans. Num. Anal., 33 (2009), pp. 53–62.

[29] T. Hylla, *Extension of inexact Kleinman-Newton methods to a general monotonicity preserving convergence theory*, PhD thesis, Universität Trier, 2010.

[30] I. Jaimoukha and E. Kasenally, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., 31 (1994), pp. 227–251.

[31] K. Jbilou, *Block Krylov subspace methods for large algebraic Riccati equations*, Numer. Algorithms, 34 (2003), pp. 339–353.

[32] K. Jbilou, *An Arnoldi based algorithm for large algebraic Riccati equations*, Applied Mathematics Letters, 19 (2006), pp. 437–444.

[33] D. Kleinman, *On an iterative technique for Riccati equation computations*, IEEE Trans. Automat. Control, AC-13 (1968), pp. 114–115.

[34] D. Kressner and C. Tobler, *Krylov subspace methods for linear systems with tensor product structure*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1688–1714.

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

[36] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.

[37] T. Li, E. K. Chu, W. Lin, and P. Weng, *Solving large-scale continuous-time algebraic Riccati equations by doubling*, J. Comput. Appl. Math., 237 (2013), pp. 373–383.

[38] T. Li, E. K. W. Chu, and W. W. Lin, *Solving large-scale discrete-time algebraic Riccati equations by doubling*, Tech. Rep. 2012-05-002, National Center for Theoretical Science, Hsinchu, Taiwan, 2012. http://www.math.cts.nthu.edu.tw/download.php?filename=680_bf78120b.pdf&dir=publish&title=prep2012-05-002.

[39] T. Li, P. C. Y. Weng, E. K. Chu, and W. W. Lin, *Large-scale Stein and Lyapunov equations, Smith method, and applications*, Numer. Algorithms, (Published online: 13 October 2012). DOI: 10.1007/s11075-012-9650-2.

[40] T. Mach and J. Saak, *How competitive is the ADI for tensor structured equations?*, Proc. Appl. Math. Mech., 12 (2012), pp. 635–636. DOI: 10.1002/pamm.201210306.

[41] M. Marcus and H. Minc, *A Survey of Matrix Theory and Matrix Inequalities*, Allyn and Bacon, Boston, 1964.

[42] V. Mehrmann and T. Stykel, *Balanced truncation model reduction for large-scale systems in descripter form*, 2005. Chapter 3 (pages 83–115) of [13].

[43] H. Mena and P. Benner, *Rosenbrock methods for solving differential Riccati equations*, IEEE Trans. Automat. Control, (2013, to appear).

[44] C. C. K. Mikkelsen, *Any positive residual curve is possible for the Arnoldi method for Lyapunov matrix equations*, Tech. Rep. UMINF 10.03, Department of Computing Science and HPC2N, UmeåUniversity, 2010.

[45] T. Penzl, *A cyclic low rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (2000), pp. 1401–1418.

[46] ——, *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Sys. Control Lett., 40 (2000), pp. 139–144.

[47] ——, Lyapack *Users Guide*, Tech. Rep. SFB393/00-33, Sonderforschungsbereich 393 *Numerische Simulation auf massiv parallelen Rechnern*, TU Chemnitz, 09107 Chemnitz, Germany, 2000. Available from http://www.tu-chemnitz.de/sfb393/sfb00pr.html.

[48] X. Rao, *Large scale stabilization with linear feedback*, Master's thesis, Florida State University, 1999.

[49] Y. Saad, *Numerical solution of large Lyapunov equation*, in Signal Processing, Scattering, Operator Theory and Numerical Methods, M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, eds., Birkhäuser, 1990, pp. 503–511.

[50] J. Saak, *Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction*, PhD thesis, TU Chemnitz, July 2009. available from http://nbn-resolving.de/urn:nbn:de:bsz:ch1-200901642.

[51] J. Sabino, *Solution of Large-Scale Lyapunov Equations via the Block Modified Smith Method*, PhD thesis, Rice University, Houston, Texas, June 2007. available from: http://www.caam.rice.edu/tech_reports/2006/TR06-08.pdf.

[52] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.

[53] V. Simoncini, D. B. Szyld, and M. Monsalve, *On the numerical solution of large-scale Riccati equations*, IMA J. Numer. Anal., (2013, to appear).

[54] R. Smith, *Matrix equation $XA + BX = C$*, SIAM J. Appl. Math., 16 (1968), pp. 198–201.

[55] F. Stenger, *Handbook of Sinc Numerical Methods*, Numerical Analysis and Scientific Computing Series, Chapman & Hall/CRC, 2010.

[56] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.

[57] T. Stykel, *Low-rank iterative methods for projected generalized Lyapunov equations*, Electr. Trans. Num. Anal., 30 (2008), pp. 187–202.

[58] T. Stykel and V. Simoncini, *Krylov subspace methods for projected Lyapunov equations*, Appl. Numer. Math., 62 (2012), pp. 35–50.

[59] N. Truhar and K. Veselić, *Bounds on the trace of a solution to the Lyapunov equation with a general stable matrix*, Syst. Control Lett., 56 (2007), pp. 493–503.

[60] B. Vandereycken and S. Vandewalle, *A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2553–2579.

[61] E. Wachspress, *The ADI model problem*, 1995. Available from the author.

[62] ——, *ADI iteration parameters for the Sylvester equation*, 2000. Available from the author.