Continuation of eigendecompositions

Luca Dieci

School of Mathematics, Georgia Tech, Atlanta, GA 30332 U.S.A.

and

Alessandra Papini

Dipartimento di Energetica, Univ. of Florence, 50134 Florence, Italy

Abstract

In this work we consider continuation of block eigendecompositions of a matrix valued function. We give new theoretical results on reduction to Hessenberg and bidiagonal forms, introduce and implement algorithms to continue eigendecompositions, and give numerical examples.

Key words: Smooth eigendecompositions, Schur and Hessenberg forms, Riccati equations, continuation

1991 MSC: 65F15, 65F99

1 Introduction

Consider the following problem:

(P) Given a matrix valued function $A \in \mathcal{C}^k([0,1], \mathbb{R}^{n\times n})$, $k \geq 1$, with p groups of eigenvalues, $\Lambda_1(t), \ldots, \Lambda_p(t)$, which are disjoint for all $t \in [0,1]$. Here, p is a fixed integer between 1 and n, each Λ_i is a set of n_i eigenvalues with n_i constant, $n_1 + \cdots + n_p = n$, and complex conjugate eigenvalues

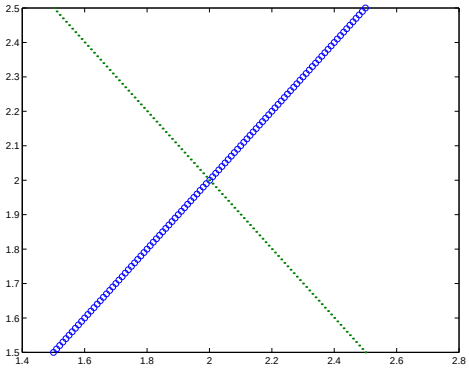
Email addresses: dieci@math.gatech.edu (Luca Dieci), papini@de.unifi.it (Alessandra Papini).

[1] Work supported in part under NSF Grant DMS-9973266 and INDAM-GNCS. The work was initiated during a visit of the first author to the University of Florence.

are grouped together. We want to smoothly transform A to real block triangular or (bi)diagonal form so that the eigenvalues of the diagonal blocks are given by the Λ_i, $i = 1, \ldots, p$.

Clearly, for each given t, $A(t)$ may be transformed to block diagonal structure with diagonal blocks associated to disjoint groups of eigenvalues. This kind of "static" block diagonalization is simple to perform with existing software, say using the methods available in `Lapack` and `Matlab`, by first taking an ordered Schur decomposition clustering together close eigenvalues, and then zeroing the off diagonal blocks by solving Sylvester equations; see [3,13,14]. However, in general, if we take two different, but arbitrarily close, values of t, say t_1 and t_2, and perform block-diagonalizations of $A(t_1)$ and $A(t_2)$ using these static algorithms, then the associated transformations can be far from one another, betraying that the overall process is not "smooth".

Example 1.1 Consider the function (adapted from [2]) $A : \; t \to A(t) = \begin{bmatrix} t & 10^{-2} \\ 10^{-4} & 4-t \end{bmatrix}$. Clearly, A is smooth (in fact, analytic), with distinct smooth eigenvalues for all t, given by $\lambda_\pm = 2 \pm \sqrt{(2-t)^2 + 10^{-6}}$, and a basis of eigenvectors can also be chosen smooth. Using either of the commands `eig` or `schur` of `Matlab` (versions 5 and 6) on 101 equispaced values of $A(t)$ for $t \in [1.5, 2.5]$ (further refining the computation around the value $t = 2$ produces no noticeable difference), and plotting the eigenvalues in the order in which they are returned by `Matlab`, gives the figure on the right (`Matlab` returns first the eigenvalue labeled "o"). Naturally, one may suspect that the eigenvalues have crossed each other, whereas obviously they have not.



The following well known Theorem tells us that under the conditions in (P) there is a smooth block eigendecomposition. For a proof, see [8, Proposition 2.6, Remark 2.6]. For analytical aspects of the general topic of smooth decompositions, we refer to [5,8,11,15].

Theorem 1.1 (Block Eigendecompositions) *Let $A \in \mathcal{C}^k([0,1], \mathbb{R}^{n\times n})$, $k \geq 1$, satisfy the conditions in* (P)*. Then, there exists a $\mathcal{C}^k$ orthogonal function Q such that, for all t, the matrix $R(t) = Q^T(t)A(t)Q(t)$ has the structure $\begin{bmatrix} R_{11} & R_{12} & \ldots & R_{1n} \\ 0 & R_{22} & \ldots & R_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & R_{pp} \end{bmatrix}$ where $R_{i,i}(t) \in \mathbb{R}^{n_i \times n_i}$, $R_{i,j}(t) \in \mathbb{R}^{n_i \times n_j}$, and $\sigma(R_{ii}(t)) = \Lambda_i(t)$, $i = 1, \ldots, p$, $j = i+1, \ldots, p-1$. Further, the function R can be smoothly brought to block-diagonal form.*

The problem of computing a smooth factorization of a matrix valued function has recently received a good deal of attention. There are several reasons for the interest in this topic. The standard one is that quite often we need to find decompositions of nearby matrices, and for the sake of efficiency we would like to exploit some of the work already done; e.g., see [6]. But another reason is that we often need to use the smooth factors in order to find changes of variables, or build smooth coordinate systems, or build smooth projections, or follow selected groups of eigenvalues, all issues of common occurrence in numerical studies of dynamical systems; e.g., see [4,7,10,18].

The first general methods for smoothly continuing matrix decompositions exploited the idea of "least variation" with respect to a reference factor in order to enforce smoothness; see [18,4,5]. Later approaches (see [17,19]) attempted direct integration of underlying differential equations in order to find the smooth path of factorizations; the key advantage, now, being that the values of t where the factorizations are computed can be chosen adaptively, but the disadvantage being that the factors are no longer exact. In [9], the authors used a combination of these previous ideas to compute a smooth block Schur form in two distinct groups of eigenvalues, by an algorithm which adaptively chooses the values of t where the decompositions are found while still obtaining exact factors.

The algorithms we propose in this paper to smoothly continue block eigendecompositions of A, for $t \in [0, 1]$, are most closely related to the approach in [9]. We will make two new algorithmic contributions: (i) First, we will extend the approach of [9] to compute smooth block Schur factorizations in case of more than two blocks of eigenvalues; (ii) Second, we will propose a new method to smoothly (block) bidiagonalize A by exploiting prior reduction of A to lower Hessenberg form. To justify the latter technique, we will give new theoretical results on smooth reduction to Hessenberg form. This will be done in the next section. In Section 3 we introduce and examine our algorithms, and in Section 4 we exemplify performance of our algorithms on some test cases.

2 Hessenberg forms

As it will become clear in the next section, to compute a Schur form based upon Theorem 1.1 is a potentially expensive process, which becomes greatly simplified if the function A gets *a priori* transformed in a form more amenable to efficient computations. An appropriate form is lower Hessenberg structure [2]. Therefore, we now give some results on smoothness of orthogonal decom-

[2] In the standard linear algebra context, it is upper Hessenberg structure which is invoked in order to simplify the computations leading to a Schur decomposition

positions to Hessenberg form, and further on decompositions to bidiagonal structure from a Hessenberg form. Besides our own algorithmic motivation for deriving these results, they are of independent interest.

Although we consider lower Hessenberg structure, obviously the results below have an immediate counterpart for upper Hessenberg structure. Finally, recall that a matrix H is in lower Hessenberg form if $H_{i,j} = 0,\ i = 1, \ldots, n-2,\ j = i+2, \ldots, n$, and it is called *unreduced* (or *proper*) if $H_{i,i+1} \neq 0, i = 1, \ldots, n-1$.

First consider transformation to unreduced Hessenberg structure.

Theorem 2.1 *Let $A \in \mathcal{C}^k([0,1], \mathbb{R}^{n\times n})$, $k \geq 1$. Let $q_1 \in \mathbb{R}^n$ be a fixed orthogonal vector ($q_1^T q_1 = 1$). Suppose that for each given $t \in [0,1]$ $A(t)$ can be brought into unreduced lower Hessenberg form by a similarity transformation with an orthogonal matrix having q_1 as its first column. Then, there exists a function $Q :\ t \in [0,1] \ \rightarrow Q(t) = [q_1\, Q_2(t)]$, such that $Q^T(t)A(t)Q(t)$ is unreduced lower Hessenberg and $Q \in \mathcal{C}^k$.*

Proof For notational convenience, let $B(t) = A^T(t)$, for all t. First, we remark (see [13, Theorem 7.4.3]) that the assumptions imply that the following matrix valued function has full rank for all t:

$$K(t) := \left[q_1\ B(t)q_1\ B^2(t)q_1\ \ldots\ B^{n-1}(t)q_1\right]. \tag{2.1}$$

On the other hand, it is known (e.g., see [8]) that a full rank $\mathcal{C}^k$ function (for us, K above) admits a $\mathcal{C}^k$ QR factorization with Q orthogonal and R upper triangular and invertible; e.g., we can choose the unique Q so that $\mathrm{diag}(R)$ is positive. Thus, we have $K(t) = Q(t)R(t)$ and necessarily $Q(t) = [q_1\, Q_2(t)]$ for all t. We now verify that with such Q we have that $H(t) := Q^T(t)B(t)Q(t)$ is in unreduced upper Hessenberg form for all t. For completeness, we give this argument, although it is the same used in [13, Theorem 7.4.3]. Rewrite $K = QR$ as $R = Q^T K$ so that, for all t:

$$\left[e_1\ H(t)e_1\ H^2(t)e_1\ \ldots\ H^{n-1}(t)e_1\right] = R(t)\,. \tag{2.2}$$

Reasoning column-wise, and using invertibility of R, we must have $R_{:,j+1} = HR_{:,j},\ j = 1, \ldots, n-1$, which implies that H must be upper Hessenberg. Since $R_{nn} = H_{21}H_{32}\cdots H_{n,n-1}$, then H must be unreduced. Finally, that H is smooth is obvious, since Q and A are. □

The proof of Theorem 2.1 fails without unreduced Hessenberg structure, since having $\mathcal{C}^k$ functions Q and R which give the QR factorization of K in (2.1) generally requires full rank of K. The result below shows that, in general, some loss of smoothness will take place if K is not full rank.

Theorem 2.2 *Let $A \in \mathcal{C}^k([0,1], \mathbb{R}^{n\times n})$, $k \geq 1$, and let K be defined as in*

(2.1). *Assume that there exists an integer d, $0 \le d \le k$, such that for every t*

$$\limsup_{\tau \to 0} \frac{1}{\tau^{2d}} \det\Big(K^T K(t+\tau)\Big) > 0\,.$$

Given an unreduced lower Hessenberg decomposition at t_0, $A(t_0) = Q^T(t_0)H(t_0)Q(t_0)$ with $Q(t_0) = [q_1\, Q_2(t_0)]$, then there exists a C^{k-d} function Q of the form $Q(t) = [q_1\, Q_2(t)]$, defined in a neighborhood of t_0 and passing through $Q(t_0)$, such that, for all t, $H(t) = Q^T(t)A(t)Q(t)$ is lower Hessenberg, and H is $\mathcal{C}^{k-d}$.

Proof The proof uses [8, Theorem 3.1]. In particular, in this cited work it is shown that under the stated assumptions there exist orthogonal Q and upper triangular R, both $\mathcal{C}^{k-d}$, such that $K(t) = Q(t)R(t)$, for all t. Further, in the proof of [8, Theorem 3.1] is also shown that the function K has full rank except at most at isolated points. Now, for all t, consider the function $H(t) = Q^T(t)A^T(t)Q(t)$ which is therefore a $\mathcal{C}^{k-d}$ function. From the relation (2.2), as in the proof of Theorem 2.1, we have that H is unreduced upper Hessenberg in intervals where R (hence K) is full rank. By continuity, H will stay upper Hessenberg (although not necessarily unreduced) also at points of lower rank, since these are isolated. □

Remark 2.1 Suppose in Theorem 2.2, $\hat{t}$ is a point where R looses rank, and let $\hat{R} = R(\hat{t})$, $\hat{H} = H(\hat{t})$. Let $\hat{R}_{r+1,r+1} = 0$, and $\hat{R}_{ii} \neq 0$, $i = 1, \ldots, r$. Then, exploiting the relation (2.2), it is easy to see that we must have $\hat{H} = [\hat{H}_1\, \hat{H}_2]$ where $\hat{H}_1 \in \mathbb{R}^{n\times r}$ is upper Hessenberg, unreduced in its first $r-1$ columns, and $\hat{H}_{r+1,r} = 0$, while $\hat{H}_2$ is undetermined from the algebraic relation (2.2). Further, we must also have $\hat{R} = \left[\begin{smallmatrix} \hat{R}_1 & \hat{R}_2 \\ 0 & 0 \end{smallmatrix}\right]$ where $\hat{R}_1 \in \mathbb{R}^{r\times r}$ is invertible. Therefore, we can conclude that if $\hat{R}_{r+1,r+1} = 0$, then $\hat{R}$ (and $\hat{K}$) have rank r.

Remark 2.2 A special case of Theorem 2.2 is when the function K in (2.1) has rank deficiency at most 1. In this case, from the previous Remark, we can easily see that $\hat{H}$ will be unreduced upper Hessenberg in its first $(n-2)$ columns, and will have $\hat{H}_{n,n-1} = 0$. In other words, necessarily, $\hat{R}$ will be invertible in its leading $(n-1, n-1)$ block, and will have $\hat{R}_{nn} = 0$. In this case, it is to be expected that $H_{n,n-1}$ will cross 0 in a generic fashion, hence it will change sign, in principle enabling us to detect a transition from unreduced to reduced Hessenberg forms.

We now discuss how to (smoothly) bidiagonalize a smooth Hessenberg function. First, we restrict attention to the unreduced case, and delay further considerations of reduced Hessenberg structure to Theorem 2.6 and to Sections 3 and 4 for the practical impact of unreduced Hessenberg structure.

Theorem 2.3 *Let $H \in \mathbb{R}^{n\times n}$ be an unreduced lower Hessenberg matrix. Let the columns of the full rank matrix $V \in \mathbb{R}^{n\times m}$ span a real invariant subspace*

of H; that is: $HV = VC$, with $C \in \mathbb{R}^{m\times m}$. Then, we can always choose V of the form $V = \left[\begin{smallmatrix} I_m \\ X \end{smallmatrix}\right]$, $X \in \mathbb{R}^{(n-m)\times m}$.

Proof The statement is equivalent to saying that V is of the form $V = \left[\begin{smallmatrix} V_1 \\ V_2 \end{smallmatrix}\right]$ with $V_1 \in \mathbb{R}^{m\times m}$ invertible. [The equivalency is clear, since if V_1 is invertible, then from $HV = VC$, we can rewrite $H \left[\begin{smallmatrix} V_1 \\ V_2 \end{smallmatrix}\right] V_1^{-1} = \left[\begin{smallmatrix} V_1 \\ V_2 \end{smallmatrix}\right] CV_1^{-1}$.] We will prove this result by contradiction. So, suppose we have full rank $V = \left[\begin{smallmatrix} V_1 \\ V_2 \end{smallmatrix}\right]$ such that (with partitioning of H inherited from that of V) $\left[\begin{smallmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{smallmatrix}\right] \left[\begin{smallmatrix} V_1 \\ V_2 \end{smallmatrix}\right] = \left[\begin{smallmatrix} V_1 \\ V_2 \end{smallmatrix}\right] C$, and that V_1 is singular. Then, there exist nonzero vectors $x = \left[\begin{smallmatrix} x_1 \\ \vdots \\ x_m \end{smallmatrix}\right]$ and $y = \left[\begin{smallmatrix} y_1 \\ \vdots \\ y_m \end{smallmatrix}\right]$ both in $\mathbb{R}^m$ such that $V_1 x = 0$ and $y^T V_1 = 0$. That is, we must have

$$\begin{bmatrix} y^T & 0 \end{bmatrix} \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} x = \begin{bmatrix} y^T & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} Cx \; . \tag{2.3}$$

Let $z = V_2 x$, and observe that z cannot be zero as otherwise V would not have full rank. Also, recall that H is unreduced lower Hessenberg, hence H_{12} is of the form $H_{12} = H_{m,m+1} \left[\begin{smallmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \end{smallmatrix}\right]$. Therefore, from (2.3), we must have

$$y^T H_{12} z = 0 \; , \quad \text{or} \quad H_{m,m+1} y_m z_1 = 0 \, .$$

Since H is unreduced, then we must have either (or both)

$$\text{(a) } y_m = 0 \; , \quad \text{or} \quad \text{(b) } z_1 = 0 \, .$$

Now we will proceed by cases.

I. First, suppose that $y_m = 0$, and that $z_1 \neq 0$. Then, from $\left[\begin{smallmatrix} y^T & 0 \end{smallmatrix}\right] \left[\begin{smallmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{smallmatrix}\right] \left[\begin{smallmatrix} V_1 \\ V_2 \end{smallmatrix}\right] = \left[\begin{smallmatrix} y^T & 0 \end{smallmatrix}\right] \left[\begin{smallmatrix} V_1 \\ V_2 \end{smallmatrix}\right] C$ we must have $(y^T H_{11}) V_1 = 0$. Now, the vectors $y^T H_{11}$ and y^T must be independent if $y \neq 0$: otherwise, we would need to have $y^T H_{11} = \alpha y^T$ with nonzero α which would force $y = 0$ since H_{11} is unreduced lower Hessenberg. Thus, we have $\left[\begin{smallmatrix} y^T & 0 \\ y^T H_{11} & 0 \end{smallmatrix}\right] \left[\begin{smallmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{smallmatrix}\right] \left[\begin{smallmatrix} 0 \\ z \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right]$, from which we must have

$$y^T H_{11} H_{12} z = 0 \; , \quad \text{or} \quad H_{m,m+1} z_1 H_{m-1,m} y_{m-1} = 0 \, ,$$

which forces $y_{m-1} = 0$. Further, as before, the vector $y^T H_{11}^2$ is independent of y^T and $y^T H_{11}$. We continue this process until eventually we obtain $y = 0$. Therefore, we cannot have $y_m = 0$ and $z_1 \neq 0$.

II. Now suppose $z_1 = 0$. From $\left[\begin{smallmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{smallmatrix}\right] \left[\begin{smallmatrix} 0 \\ z \end{smallmatrix}\right] = \left[\begin{smallmatrix} V_1 \\ V_2 \end{smallmatrix}\right] Cx$ we must have $V_1 Cx = 0$, and $H_{22} z = V_2 Cx$. Observe that the vectors z and $w := V_2 Cx$ are independent if $z \neq 0$: otherwise, we would need to have $w = \alpha z$, for nonzero α, and using the relation $H_{22} z = w = \alpha z$, we would get $z = 0$. But then, we must

have $\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \begin{bmatrix} x & Cx \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} C \begin{bmatrix} x & Cx \end{bmatrix}$, from which we then must have

$$H_{12}w = V_1C^2x \Rightarrow y^T H_{12}w = 0 \ , \quad \text{or} \quad H_{m,m+1}y_m w_1 = 0 \ .$$

If $y_m \neq 0$, then $w_1 = 0$. This fact, coupled with the relation $H_{22}z = w$ gives $z = 0$, a contradiction. If, on the other hand, $y_m = 0$, we have two possibilities to consider. The first is that $w_1 = 0$, which still leads to the contradiction $z = 0$ as above. Alternatively, if we have $y_m = 0$ and $w_1 \neq 0$, then we can repeat the proof of the case **I**, with the vector Cx replacing x there (i.e., with w replacing z). Also in this case we reach the desired contradiction. □

Corollary 2.4 *Let H be an unreduced lower Hessenberg matrix with p disjoint groups of eigenvalues $\Lambda_1, \ldots, \Lambda_p$, where complex conjugate eigenvalues are grouped together. Write H in block notation as in Theorem* 1.1*: $H = \begin{bmatrix} H_{11} & H_{12} & 0 & \ldots & 0 \\ H_{12} & H_{22} & H_{23} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ H_{p-1,1} & \ldots & H_{p-1,p-2} & H_{p-1,p-1} & H_{p-1,p} \\ H_{p1} & \ldots & H_{p,p-2} & H_{p,p-1} & H_{pp} \end{bmatrix}$. Then, there exists a matrix T of the form $T = \begin{bmatrix} I & 0 & \ldots & 0 \\ X_{21} & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ X_{p1} & \ldots & X_{p,p-1} & I \end{bmatrix}$ such that $T^{-1}HT$ is of the form $\begin{bmatrix} B_{11} & B_{12} & 0 & \ldots & 0 \\ 0 & B_{22} & B_{23} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & B_{p-1,p-1} & B_{p-1,p} \\ 0 & \ldots & 0 & 0 & B_{pp} \end{bmatrix}$ with $\sigma(B_{ii}) = \Lambda_i$, $i = 1, \ldots, p$, and $B_{i,i+1} = H_{i,i+1}$, $i = 1, \ldots, p-1$.*

Proof Using Theorem 2.3, and exploiting the fact that H is in lower Hessenberg form, we first build T_1 of the form $\begin{bmatrix} I & 0 \\ X_{2:p,1} & I \end{bmatrix}$ such that $T_1^{-1}HT_1 = \begin{bmatrix} B_{11} & H_{1,2:p} \\ 0 & H_{2:p,2:p} - X_{2:p,1}H_{1,2:p} \end{bmatrix}$, where we notice that B_{11} is unreduced lower Hessenberg and so is $H_{2:p,2:p} - X_{2:p,1}H_{1,2:p}$. So, with the relabeling $H_{2:p,2:p} \leftarrow H_{2:p,2:p} - X_{2:p,1}H_{1,2:p}$, and using again Theorem 2.3, we can build T_2 of the form $T_2 = \begin{bmatrix} I & 0 \\ 0 & \hat{T}_2 \end{bmatrix}$, with $\hat{T}_2 = \begin{bmatrix} I & 0 \\ X_{3:p,2} & I \end{bmatrix}$ such that $\hat{T}_2^{-1}H_{2:p,2:p}\hat{T}_2 = \begin{bmatrix} B_{22} & H_{2,3:p} \\ 0 & H_{3:p,3:p} - X_{3:p,2}H_{2,3:p} \end{bmatrix}$, and once more B_{22} is unreduced lower Hessenberg and so is $H_{3:p,3:p} - X_{3:p,2}H_{2,3:p}$. Continuing in this way, and setting $T = T_1T_2 \ldots T_{p-1}$ gives the result. □

With same notation as in Corollary 2.4, we can now prove the following.

Theorem 2.5 *Let $H \in \mathcal{C}^k([0,1], \mathbb{R}^{n\times n})$, $k \geq 1$, be an unreduced lower Hessenberg function. Let H have p groups of eigenvalues $\Lambda_1(t), \ldots, \Lambda_p(t)$, which stay disjoint for all $t \in [0,1]$, and contain a constant number of eigenvalues (counted with their multiplicity) $n_1, \ldots, n_p$, and with complex conjugate eigenvalues grouped together. Then, there exists a real valued $\mathcal{C}^k$ function T, of the form*

$$T = \begin{bmatrix} I & 0 & \ldots & 0 \\ X_{21} & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ X_{p1} & \ldots & X_{p,p-1} & I \end{bmatrix} \ , \tag{2.4}$$

with diagonal identity blocks of dimension n_i, $i = 1, \ldots, p$, such that, for all t,

$$T^{-1}(t)H(t)T(t) =: B(t) = \begin{bmatrix} B_{11} & B_{12} & 0 & \ldots & 0 \\ 0 & B_{22} & B_{23} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & B_{p-1,p-1} & B_{p-1,p} \\ 0 & \ldots & 0 & 0 & B_{pp} \end{bmatrix} . \tag{2.5}$$

Here, $B_{ii}(t) \in \mathbb{R}^{n_i \times n_i}$, *and* $\sigma(B_{ii}(t)) = \Lambda_i(t)$, $i = 1, \ldots, p$. *Further, each* B_{ii}, $i = 1, \ldots, p$, *is in unreduced lower Hessenberg structure, and* $B_{i,i+1} = H_{i,i+1}$, $i = 1, \ldots, p-1$, *are of the form* $\begin{bmatrix} 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \ldots & 0 \\ * & 0 & \ldots & 0 \end{bmatrix}$.

Proof Observe that, given that H is in unreduced lower Hessenberg form, if T exists, the fact that B_{ii}, $i = 1, \ldots, p$, are lower Hessenberg and that $B_{i,i+1}$, $i = 1, \ldots, p-1$, are of the stated form follows at once. To show that T exists, and $\mathcal{C}^k$, we use the implicit function theorem and derive a differential equation defining T. Using Corollary 2.4, at $t = 0$ we can choose the initial condition T_0 of the stated form. Next, we differentiate the relation $B(t) = T^{-1}(t)H(t)T(t)$ to obtain $\dot{B} = -T^{-1}\dot{T}B + T^{-1}\dot{H}T + BT^{-1}\dot{T}$. With $K := T^{-1}\dot{T}$, and $C := \dot{B} - T^{-1}\dot{H}T$, we can rewrite this relation as $BK - KB = C$. Now, observe that K is strictly block lower triangular, with blocking inherited from that of T: $K = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ K_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ K_{p1} & \ldots & K_{p,p-1} & 0 \end{bmatrix}$. Therefore, we can use the 0-structure of the strictly lower triangular part of B to recursively find the entries of K. With the agreement that entries of index "0" are 0, we can find K as follows:

$$\begin{aligned} &\text{For} \quad j = 1, \ldots, p-1, \quad \text{Solve} \\ &B_{j+1:p,j+1:p}K_{j+1:p,j} - K_{j+1:p,j}B_{jj} = C_{j+1:p,j} + K_{j+1:p,j-1}B_{j-1,j}\,. \end{aligned} \tag{2.6}$$

These Sylvester equations are uniquely solvable since $\sigma(B_{ii}) \cap \sigma(B_{jj}) = \emptyset$, $i \neq j$, so K is well defined and the result follows. □

A complete understanding in case we fail to have unreduced Hessenberg form still eludes us. However, in the special case of which in Remark 2.2, we believe we understand what can be expected. This is because of the following result, which extends Theorem 2.3.

Theorem 2.6 *Let* $H \in \mathbb{R}^{n \times n}$ *be a lower Hessenberg matrix, unreduced in its first* $(n-2)$ *rows:* $H_{k,k+1} \neq 0$, $k = 1, \ldots, n-2$, $H_{n-1,n} = 0$. *Let* Λ_1 *be any subset of* m *eigenvalues of* H, *counted with their algebraic multiplicity, and* Λ_2 *be the complementary subset of* $(n-m)$ *eigenvalues, so that* Λ_1 *and* Λ_2 *have no common eigenvalues, and let complex conjugate pairs be grouped together. Let* $V \in \mathbb{R}^{n \times m}$, $V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$ *with* $V_1 \in \mathbb{R}^{m \times m}$, *be a full rank matrix whose columns span an invariant subspace relative to* Λ_1. *Then,* V_1 *is singular if and only if* $H_{nn} \in \Lambda_1$.

Proof Clearly H_{nn} is an eigenvalue of H with eigenvector e_n (the n-th unit vector).

($\Leftarrow$) If $H_{nn} \in \Lambda_1$, then $e_n = Vc$, for some nonzero vector $c \in \mathbb{R}^m$. But then $V_1 c = 0$, and hence V_1 is singular.

($\Rightarrow$) If V_1 is singular, then there is a nonzero vector $c \in \mathbb{R}^m$ such that $V_1 c = 0$.

We now prove that Vc is in the direction of e_n, from which it will follow that $H_{nn} \in \Lambda_1$. The proof is by induction on m (the rank of V).
[$m = 1$]. Here $V = V_{1:n,1}$ and V_1 singular means $V_{11} = 0$. Using $HV = VC$ (here, C is an eigenvalue), then since $H_{k,k+1} \neq 0$, for $k = 1, \ldots, n-2$, we must have $V_{j1} = 0$, $j = 1, \ldots, n-1$, and since $V \neq 0$ then $V_{n1} \neq 0$ and thus V is in the direction of e_n.
[$m > 1$]. From $V_1 c = 0$ for some nonzero $c \in \mathbb{R}^m$, we have $V_{1:m,m} c_m = -V_{1:m,1:m-1} c_{1:m-1}$, and if $c_m = 0$ (with $c \neq 0$), we would have $V_{1:m-1,1:m-1}$ singular and so (by the induction hypothesis) we would have that $Vc = V_{1:n,1:m-1} c_{1:m-1}$ is in the direction of e_n. So, let $c_m \neq 0$, and notice that then $V_{1:m-1,m} = -V_{1:m-1,1:m-1} c_{1:m-1}/c_m$. Next, let $x = Vc$ and $b = Cc$. Since $HV = VC$, then we must have $Hx = Vb$. From the form of x, we have $x_k = 0$, $k = 1, \ldots, m$. So, we have $V_{1:m-1,1:m} b = 0$, and thus $V_{1:m-1,1:m-1} b_{1:m-1} + b_m V_{1:m-1,m} = 0$. But, using the previous expression for $V_{1:m-1,m}$, we then have $V_{1:m-1,1:m-1}\gamma = 0$, where $\gamma \in \mathbb{R}^{m-1}$ has entries $b_j - b_m c_j/c_m$, $j = 1, \ldots, m-1$. As above, if $\gamma \neq 0$, the induction hypothesis would give $V_{1:m-1,1:m-1}$ singular, and further $V_{1:n,1:m-1}\gamma$ in the direction of e_n by the induction hypothesis, and so also $V \left[\begin{smallmatrix} \gamma \\ 0 \end{smallmatrix}\right]$. Therefore, we can let $\gamma = 0$. Now, consider once more the relation $Hx = Vb$, which gives $H_{k,k+1} x_{k+1} = V_{k,1:m} b = V_{k,1:m-1}\gamma = 0$, for $k = m, \ldots, n-1$. Thus, since $H_{k,k+1} \neq 0$, we have $x_{k+1} = 0$, for $k = m, \ldots, n-2$, and thus (since V has full rank), we must have x in the direction of e_n. □

Remark 2.3 Theorem 2.6 tells us when we cannot expect to have a smooth bidiagonalization procedure as in Theorem 2.5, in case in which the Hessenberg function H of which in Theorem 2.5 is not unreduced but we have only $H_{n-1,n} = 0$. Indeed, if the eigenvalue H_{nn} which has emerged has to be moved away from the (n, n) position, then we can expect lack of smoothness (better, of existence) for the transformation T of which in Theorem 2.5.

3 Algorithms

We present two algorithms, one to obtain the Schur form of Theorem 1.1, the other to obtain the block bidiagonal form of Theorem 2.5.

3.1 Block Schur

This algorithm is an extension of the algorithm in [7,9], to which we refer for justification of some of the choices adopted. Notation is as in Theorem 1.1.

Suppose we have accomplished the Schur reduction at $t = t_0$ (initially $t_0 = 0$), that is we have $Q(t_0)$ and $R(t_0)$ such that $Q(t_0)^T A(t_0) Q(t_0) = R(t_0)$, and

suppose we have the value h of the step to be taken, that is we want to obtain the Schur factorization of $A(t_1),\ t_1 = t_0 + h$.

Algorithm 1.
(i) Form $\hat{R} := Q(t_0)^T A(t_1) Q(t_0)$ ($\hat{R}$ is close to being block upper triangular, for small h). Next, seek Q_1 such that $Q_1^T \hat{R} Q_1 = R(t_1)$.
(ii) Annihilate the lower triangular entries of $\hat{R}$ by using the lower triangular transformation $T_1 = \begin{bmatrix} I & 0 & \dots & 0 \\ X_{21} & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ X_{p1} & \dots & X_{p,p-1} & I \end{bmatrix}$. This implies that we need to solve algebraic Riccati equations. The general recursion for this task is as follows.

$$\begin{aligned} &\text{For } j = 1, 2, \dots, p-1, \quad \text{let} \quad i = j+1 \text{ .Solve } F(X_{i:p,j}) = 0\ , \quad \text{where} \\ &F(X_{i:p,j}) = \hat{R}_{i:p,j} + \hat{R}_{i:p,i:p} X_{i:p,j} - X_{i:p,j} \hat{R}_{jj} - X_{i:p,j} \hat{R}_{j,i:p} X_{i:p,j} \\ &\text{Update} \quad \hat{R}_{1:j,j} \leftarrow \hat{R}_{1:j,j} + \hat{R}_{1:j,i:p} X_{i:p,j}\ , \quad \hat{R}_{i:p,i:p} \leftarrow \hat{R}_{i:p,i:p} - X_{i:p,j} \hat{R}_{j,i:p}\ . \end{aligned} \tag{3.1}$$

The result is a new matrix $\hat{R}$ such that $\hat{R}$ is block upper triangular.
(iii) Next, recover a Schur form by performing the QR factorization of T_1: $T_1 = Q_1 U_1$, where U_1 is upper triangular with positive diagonal entries. With this Q_1, set $Q(t_1) = Q(t_0) Q_1$ and $R(t_1) = U_1 \hat{R} U_1^{-1}$.

To solve the Riccati equation in (3.1), we can use a Newton or stationary Newton's iteration. In the latter case, with notation from (3.1), for a given initial guess $X_{i:p,j}^{(0)}$ and a value `TOL` for the stopping criterion, we recursively solve the following Sylvester equations

$$\begin{aligned} &\text{For } k = 0, \dots, K_{\max} \quad \text{Solve} \\ &[\hat{R}_{i:p,i:p} - X_{i:p,j}^{(0)} \hat{R}_{j,i:p}] Y - Y [\hat{R}_{jj} + \hat{R}_{j,i:p} X_{i:p,j}^{(0)}] = -F(X_{i:p,j}^{(k)})\ , \\ &\text{Update } X_{i:p,j}^{(k+1)} \leftarrow X_{i:p,j}^{(k)} + Y\ . \quad \text{If } \frac{\|Y\|}{1 + \left\| X_{i:p,j}^{(k+1)} \right\|} \leq \texttt{TOL} \text{ Stop}\ . \end{aligned} \tag{3.2}$$

Initial Guess. As far as the initial guess $X_{i:p,j}^{(0)}$, we experimented with two choices. The first choice is the so-called **trivial** predictor: $X_{i:p,j}^{(0)} = 0$, which has the major benefit of simplicity. We also used the so-called **tangent** predictor, which usually performs better than the trivial predictor and can be obtained as follows. Recall that we seek Q_1 such that $Q_1^T \hat{R} Q_1 = R(t_1)$. Further, we know (see Theorem 1.1) that on $[t_0, t_1]$ there is a smooth orthogonal $U(t)$ such that $U(t_0) = I$ and $U^T(t)(Q(t_0) A(t) Q(t_0)) U(t) = R(t)$. If we differentiate this relation, and formally let $S(t) = U^T(t) \dot{U}$ then we get that U and R must satisfy the differential equations

$$\begin{aligned} &\text{(a)} \quad \dot{R} = U^T (Q(t_0)^T \dot{A} Q(t_0)) U - SR + RS\ ,\ R(t_0) \ \text{ given} \\ &\text{(b)} \quad \dot{U} = US\ ,\ U(t_0) = I\ . \end{aligned} \tag{3.3}$$

Observe that if we had $U(t_1)$, then we could recover (the exact) T_1, and hence all of the X_{ij}, $i = 2, \ldots, p$, $j = i-1, \ldots, p-1$, from performing a block LU-factorization of $U(t_1)$ with identity blocks on the diagonal of L: $U(t_1) = T_1 \hat{U}$. Then, as in [9], we find a second order (in h) approximation to $U(t_1)$ by taking a Euler step in the differential equation (3.3)-(b): $U_1 = I + hS(t_0) = U(t_1) + O(h^2)$. Then, the required tangent approximation for T_1 can be obtained from a block LU-factorization of U_1. To find $S(t_0)$, we can use (3.3)-(a). Indeed, evaluating (3.3)-(a) at t_0 gives $\dot{R}(t_0) = Q(t_0)^T \dot{A}(t_0) Q(t_0) - S(t_0)R(t_0) + R(t_0)S(t_0)$, and further using the (1st order) approximation $\dot{A}(t_0) \approx \frac{1}{h}(A(t_1) - A(t_0))$, then says that a second order approximation to $hS(t_0)$, which we call $\hat{S}$, must satisfy the Sylvester equation $R(t_0)\hat{S} - \hat{S}R(t_0) = \dot{R}(t_0) - Q(t_0)^T A(t_1) Q(t_0) + R(t_0)$. Now, observe that $\hat{S}$, like S, must be antisymmetric, and recall from [8,9] that its diagonal blocks can be set to 0. Therefore, we can solve the Sylvester equation for $\hat{S}$ by finding its strictly lower triangular part, exploiting the fact that both $\dot{R}(t_0)$ and $R(t_0)$ are (block) upper triangular. This gives the following recursion for the blocks of $\hat{S}$:

$$
\begin{array}{l}
\text{For } j = 1, \ldots, p-1, \quad \text{with} \quad i = j+1, \text{ Solve} \\
(R(t_0))_{i:p,i:p} \hat{S}_{i:p,j} - \hat{S}_{i:p,j} (R(t_0))_{jj} = -(Q(t_0)^T A(t_1) Q(t_0))_{i:p,j} \ , \\
\quad \text{for} \quad s = i, \ldots, p-1, \text{ Update} \quad (Q(t_0)^T A(t_1) Q(t_0))_{s+1:p,i:p-1} \\
\qquad \leftarrow (Q(t_0)^T A(t_1) Q(t_0))_{s+1:p,i:p-1} - \hat{S}_{s+1:p,j} (R(t_0))_{j,i:p-1} \ .
\end{array}
\tag{3.4}
$$

A couple of observations are in order.

(a) Of course, we may want to use a complete Newton iteration rather that the stationary Newton one. The stationary Newton iteration with tangent prediction is the common choice in the continuation literature (e.g., see [1,16]). Each iteration step of the stationary Newton is less expensive (see (b) below) than a full Newton step; however, the speed gained by a full Newton iteration often pays off (see the numerical results in Section 4).

(b) A key computational expense in the above algorithm is the need to solve the Sylvester equation (3.2) (and the one in (3.4)), rewritten compactly here as $CY - YE = G$. The standard way to do this exploits the identity $(U^T C U)(U^T Y V) - (U^T Y V)(V^T E V) = (U^T G V)$, with U and V orthogonal. We used the algorithm in [12], whereby the larger block, say C, is reduced to upper Hessenberg form, and E to real Schur form. The system is then akin to a Hessenberg one, which is inexpensive to solve (see [13]). If a complete Newton iteration is performed, then the blocks C and E above change at each Newton iteration.

Motivated by point (b) above, as well as by overall efficiency issues resulting from the required matrix/matrix multiplications, we next turn our attention to techniques which simplify "a priori" the structure of $A(\cdot)$.

The algorithm below finds the decomposition of Theorem 2.5.

Suppose we have accomplished the reduction at $t = t_0$, that is we have $Q(t_0)$ and $H(t_0)$ such that $Q(t_0)^T A(t_0) Q(t_0) = H(t_0)$, with $H(t_0)$ (unreduced) lower Hessenberg, and that we have $T(t_0)$ as in (2.4) so that $T^{-1}(t_0) H(t_0) T(t_0) = B(t_0)$, with $B(t_0)$ as in (2.5). Further, suppose we know the step h to be taken. We want to transform $A(t_1),\ t_1 = t_0 + h$, to the block bidiagonal form $B(t_1)$.

Algorithm 2.
(i) Smoothly transform $A(t_1)$ with orthogonal $Q(t_1)$ to lower Hessenberg form: $H(t_1) = Q^T(t_1) A(t_1) Q(t_1)$. Further, transform $H(t_1)$ with $T(t_0)$: set $\hat{H} = T^{-1}(t_0) H(t_1) T(t_0)$ (note: $\hat{H}$ is close to being upper bidiagonal).
(ii) Annihilate the lower triangular blocks of $\hat{H}$ by using the transformation $T = \begin{bmatrix} I & 0 & \cdots & 0 \\ X_{21} & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ X_{p1} & \cdots & X_{p,p-1} & I \end{bmatrix}$. Thus, we need to solve the following Riccati equations:

$$\begin{aligned} &\text{For } j = 1, 2, \ldots, p-1, \quad \text{let} \quad i = j + 1 \\ &\text{Solve } G(X_{i:p,j}) = 0 \quad \text{where} \\ &G(X_{i:p,j}) = \hat{H}_{i:p,j} + \hat{H}_{i:p,i:p} X_{i:p,j} - X_{i:p,j} \hat{H}_{jj} - X_{i:p,j} \hat{H}_{j,1:p} X_{i:p,j} \\ &\text{Update} \quad \hat{H}_{jj} \leftarrow \hat{H}_{jj} + \hat{H}_{j,i} X_{i,j}\,, \quad \hat{H}_{i:p,i} \leftarrow \hat{H}_{i:p,i} - X_{i:p,j} \hat{H}_{j,i}\,. \end{aligned} \tag{3.5}$$

The end result is the sought matrix $B(t_1)$.

Remark 3.1 It is very important to observe that the updates in (3.5) do not destroy the Hessenberg structure. In fact, the first update only changes the last row of $\hat{H}_{jj}$ which remains lower Hessenberg. Likewise, the second update leaves the block $\hat{H}_{i:p,i:p}$ in lower Hessenberg form. This fact produces considerable computational savings when we solve the Riccati equations.

We now discuss steps (i) and (ii) of the above Algorithm 2 in more detail.

(i). To ensure a smooth transformation of $A(t_1)$ to Hessenberg form, we can exploit the "Implicit Q Theorem", see [13, Theorem 7.4.2]. This Theorem states that " If two orthogonal matrices U_1 and U_2, with same first column, transform a given matrix into unreduced Hessenberg form, then $U_1 e_i = \pm U_2 e_i,\ i = 2, \ldots, n$". Therefore, we first find $\hat{Q}_1$ which has first column equal to the first column of $Q(t_0)$ and which transforms $A(t_1)$ into unreduced lower Hessenberg form. To find $\hat{Q}_1$ we use standard Householder transformations, as in [13]. Then, we choose a diagonal matrix $D = \mathrm{diag}(1, \pm 1, \ldots, \pm 1)$ so that $\left\| \hat{Q}_1 D - Q(t_0) \right\|_F$ is minimized, and will then let $Q(t_1) = \hat{Q}_1 D$. To solve the minimization problem is trivial: we take $+1$ on the diagonal of D if $(\hat{Q}_1^T Q(t_0))_{ii} > 0$, and -1 if $(\hat{Q}_1^T Q(t_0))_{ii} < 0$.

(ii). To solve the Riccati equation in (3.5), we may use a stationary Newton's method as for (3.1), and thus end up having to solve Sylvester equations as in (3.2), with $\hat{H}$ replacing $\hat{R}$ there. Next, we need to discuss how to choose initial guesses $X^{(0)}_{i:p,j}$ for solving (3.5). Of course, the trivial predictor $X^{(0)}_{i:p,j} = 0$ is still a possible choice. To arrive at the tangent predictor, we proceed similarly to what we did for (3.1). Recall that for $t_0 \le t \le t_1$, we can think to have a smooth Hessenberg function $H(t)$ partitioned as usual, and we seek smooth $T(t)$, of the form (2.4), such that $T^{-1}(t)H(t)T(t) = B(t)$ as in (2.5). As in the proof of Theorem 2.5, we can derive a differential equation satisfied by $T(t)$. With same notation as in Theorem 2.5, we get

$$BK - KB = \dot{B} - T^{-1}\dot{H}T\,, \quad \text{and} \quad \dot{T} = TK\ , \ \ T(t_0)\ \text{given}\ .$$

From the latter, we can obtain the second order approximation $T_1^{(0)} = T(t_0)(I + hK(t_0))$, if we have $K(t_0)$. We will proceed by using the zero structure in the relation $B(t_0)K(t_0) - K(t_0)B(t_0) = \dot{B}(t_0) - T^{-1}(t_0)\dot{H}(t_0)T(t_0)$ after performing the first order approximation $\dot{H}(t_0) \approx \frac{1}{h}(H(t_1) - H(t_0))$. Thus, we can approximate $hK(t_0)$ at second order by $\hat{K}$, which satisfies

$$\hat{K}B(t_0) - B(t_0)\hat{K} = T^{-1}(t_0)H(t_1)T(t_0) - B(t_0) - \dot{B}(t_0)\,.$$

Now, $B(t_0)$ and $\dot{B}(t_0)$ are upper bidiagonal, and $\hat{K}$ is strictly lower triangular: $\hat{K} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ K_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ K_{p1} & \cdots & K_{p,p-1} & 0 \end{bmatrix}$. Therefore, we can find $\hat{K}$ as we did in (2.6). This is essentially the same procedure used in (3.4), but the major advantage, now, is that we have a much less expensive solution process for the blocks $\hat{K}_{i:p,j}$ and for the updates, since $B(t_0)$ is upper bidiagonal.

Remark 3.2 Our construction of the block-bidiagonalization has rested on Theorem 2.5, which necessitates unreduced Hessenberg structure. In case the Hessenberg function becomes reduced, we may still be able to find (smoothly) the transformation to bidiagonal form, if the situation of which in Theorem 2.6 and Remark 2.3 applies. Otherwise, in general, failure to have unreduced Hessenberg form may lead to singularities in the transformation T, which will be betrayed by lack of convergence when trying to solve the Riccati equations (3.5); see Example 4.2 in Section 4.

3.3 Continuation

To adaptively choose the step h, we adopt a common strategy in the continuation literature (e.g., see [1,16]) whereby the step is chosen based upon the convergence behavior of the Newton iteration. In particular, for both Algorithms 1 and 2, we monitor the convergence behavior of Newton's method used for solving the Riccati equations (3.1) and (3.5), respectively. In either case,

call `Nits` the required number of iterations for convergence, where we further restrict $\texttt{Nits} \leq 7$. Then, given an old stepsize h_0, we choose the new stepsize h according to $h = 2^{(4-\texttt{Nits})/3} h_0$. Finally, we require that $h \geq h_{\min}$, where we have set $h_{\min} = 10^{-8}$ (i.e., since we work in double precision, $h_{\min} \approx \sqrt{\text{EPS}}$). The very first time, we initiate the procedure with $h = 10^{-3}$. Finally, when solving the Riccati equations (3.1) (or (3.5)), we use $\texttt{TOL} = 10^{-8}$.

4 Examples

Here we exemplify the performance of our algorithms on a few test problems. The algorithms have been implemented in `Matlab` in the way explained in Section 3.

Example 4.1 We first build a function $H \in \mathbb{R}^{8\times 8}$ in unreduced lower Hessenberg form

$$H(t) = [R^{-1}(t)C(t)R(t)]^T \ , \tag{4.1}$$

where R is upper triangular with $R_{ij}(t) = \cos(i+j)(\frac{t}{j+1} + \frac{1}{3})t^{(j-i)/2}$, $1 \leq i < j \leq 8$, and $R_{ii}(t) = \cos(i)(\frac{t}{i+1} + \frac{1}{3})e^{2-i/2}$, $1 \leq i \leq 8$. Further, $C(t)$ is the companion matrix $C(t) = \begin{bmatrix} 0 & 0 & \dots & 0 & -a_0(t) \\ 1 & 0 & \ddots & \vdots & \vdots \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots \\ 0 & \dots & 0 & 1 & -a_7(t) \end{bmatrix}$, where $a_0, \dots, a_7$, are obtained by requiring that the characteristic polynomial $P(\lambda, t) = \lambda^8 + a_7(t)\lambda^7 + \cdots + a_1(t)\lambda + a_0(t)$ of $C(t)$ be $P(\lambda, t) = p(\lambda)q(\lambda, t)$ with $p(\lambda) = (\lambda-2)(\lambda-1.5)(\lambda-1)(\lambda-0.5)$ and $q(\lambda, t) = (\lambda + a^t - 2)(\lambda + a^t - 1.5)(\lambda + a^t - 1)(\lambda + a^t - 0.5)$, $a = 2.5$.

Then, the function H in (4.1) is rotated into the function $A = Q^T H Q$, where Q is defined as the exponential (computed with the `expm` function of `Matlab`) of the antisymmetric function S: $S(t) = \left[\begin{smallmatrix} 0 & 0 \\ 0 & s(t) \end{smallmatrix}\right]$, with $s : t \to \mathbb{R}^{7\times 7}$, antisymmetric whose strictly upper triangular part is given by

$$s_{ij}(t) = (-1)^{i+j} \frac{t-1}{j+1} t^{j-i}, \quad 1 \leq i < j \ . \tag{4.2}$$

Initial time is $t = 1$, where the eigenvalues are $\pm 0.5, \pm 1, \pm 1.5, \pm 2$. We compute the eigendecompositions for $1 \leq t \leq 3$ (the negative eigenvalues slowly decrease up to $-13.625, -14.125, -14.625, -15.125$), as well as for $t \leq 1$, in which case the negative eigenvalues increase until one of them coalesces with the eigenvalue 0.5 at $t = \frac{\ln 1.5}{\ln 2.5} \approx 0.4425$.

Example 4.2 This is a test problem used in [9]. We have $A :=\ t \to \mathbb{R}^{8\times 8}$ of the form $A(t) = V^T(t)R(t)V(t)$. Here, $V(t) = e^{s(t)}$ with s skew-symmetric with strictly upper triangular entries as in (4.2), and $R(t) = \left[\begin{smallmatrix} D & C \\ 0 & -B \end{smallmatrix}\right]$, $D =$

$\mathrm{diag}(1,2,3,4)+L$, $B=\mathrm{diag}(4,3,2,1)+L^T-5^tI_4$, and $C=DX+XB$, where $X\in\mathbb{R}^{4\times 4}$ is made up by all 1's, and L is strictly lower triangular made up by all 1's. Two cases are of interest: (a) $1\le t\le 3$ whereby the eigenvalues vary from a modest size in the range $[-2,4]$ to the range $[-124,4]$; (b) $t\le 1$, until at $t=\frac{\ln 3}{\ln 5}\approx 0.68261$ we reach the double eigenvalue 1.

Results of our experiments on Examples 4.1 and 4.2 are summarized in Tables 1, 2, and have been obtained using the tangent predictor for either Algorithm 1 or 2. In the tables, we give the following. (i) The "Decomposition" we found: either a complete Schur or Bidiagonal form, `Schur` and BiDiag respectively, or a block decomposition in two blocks of size 4 ordered with respect to the initial ordering of the eigenvalues (decreasing real parts); (ii) The value t_{end}, relative to the time interval we considered; (iii) For each of the Newton or Stationary Newton iterations used to solve the Riccati equations (3.1) or (3.5), we give: `Nsteps/Nits/NF`, where `Nsteps` is the required number of steps to complete the path, `Nits` is the max number of iterations of the Newton's method for solving the Riccati equations relatively to the successful steps, and `NF` is the number of failed steps, each of which costs at least 7 iterations.

In Table 1, we report on the results for Example 4.1. The vector q_1 used for obtaining the smooth Hessenberg form is the first unit vector.

Table 1. Example 4.1.

Decomposition	t_{end}	`Newt`	`Stat-Newt`
`Block-BiDiag`	0.4	2706/10852/6	9560/38283/1
`Block-Schur`	0.4	2339/9392/3	6659/26684/1
`BiDiag`	0.4	1856/7442/11	Fail#1
`Schur`	0.4	2512/10089/2	7483/29976/2
`Block-BiDiag`	3	375/1426/17	746/2786/42
`Block-Schur`	3	Fail#2	Fail#2
`BiDiag`	3	1625/6503/0	4981/19932/0
`Schur`	3	Fail#2	Fail#2

Referring to Table 1, we notice that the bidiagonalization algorithm works very well. In case of $t_{\text{end}}=0.4$, all methods stop at $t\approx 0.4425$, where the eigenvalues coalesce. Fail#1 is due to the fact that we jump over the singularity at $t=0.44251$, and complete the path but with wrong eigenvalues' ordering. Fail#2, instead, are due to extremely slow convergence. To witness, after 5000 steps, `Block-Schur` is still at $t\approx 1.88$ with Newton and $t\approx 1.61$ with stationary Newton, whereas `Schur` is at $t\approx 1.38$ with Newton and $t\approx 1.25$ with stationary Newton.

In Table 2 we report on the results of experiments on Example 4.2. Here, the choice of an initial orthogonal vector q_1 with which to seek the smooth Hessenberg form is crucial. We generated (in `Matlab`) two "random" orthog-

onal vectors, u and v, and report on the different results in these cases. At 4 digits, these vectors are $u = [\,-0.2226\ 0.8259\ -0.0516\ 0.0431\ 0.4036\ 0.0224\ -0.0362\ -0.3148\,]$, $v = [\,-0.1611\ -0.6204\ 0.0467\ 0.1072\ -0.4270\ 0.4436\ 0.4429\ -0.0140\,]$.

Table 2. Example 4.2.

Decomposition	t_{end}	`Newt`	`Stat-Newt`
`BiDiag`: u	0.6	156/627/23	428/1740/10
`BiDiag`: v	0.6	Fail#1	Fail#1
`Schur`	0.6	Fail#2	Fail#2
`BiDiag`: u	3	Fail#1	Fail#1
`BiDiag`: v	3	12035/48157/0	Fail#3
`Schur`	3	15123/61268/0	Fail#3

Some comments on the results in Table 2. Fail#1 is due to repeated failures because of lack of unreduced Hessenberg structure: the eigenvalue given by H_{nn} is not in the group Λ_2 (see Theorem 2.6 and Remark 2.3). Fail#2, instead, is due to the fact that the Schur form jumps over the singularity at $t \approx 0.68$, whereas `BiDiag`: u has repeated failures and halts at $t \approx 0.68$. Fail#3: the procedures are extremely slow.

Example 4.3 This is Example 1.1 of the Introduction, which revealed itself quite an interesting problem. In the following list, we summarize our results for the Schur or bidiagonalization methods relative to computation for $t \in [1.5, 2.5]$, and with the general adaptive time stepping strategy of Chapter 3.

- As long as we force the methods to step exactly at $t = 2$, both methods solve the problem correctly regardless of whether we use Newton or Stationary Newton iteration and regardless of whether we use the trivial or the tangent predictors.
- If we do not force the methods to step exactly at $t = 2$, performance differs. Using the trivial predictor, the bidiagonalization method recovers always the right solution with either Newton or Stationary Newton iteration. The Schur method, instead, gives the wrong solution with Newton's method (and the correct one with Stationary Newton).
- With the Euler predictor, and either Newton or Stationary Newton iteration, both bidiagonalization and Schur methods converge to the wrong eigenvalue ordering.

Remark 4.1 Based on all our experiments above, and also others which we have done, it appears that in case of two blocks of eigenvalues the benefits offered by the prior Hessenberg reduction become minor (but see Table 1). However, for the case of complete eigendecomposition, and as long as unreduced Hessenberg structure is maintained, the new algorithm based on a combination of Hessenberg reduction and bidiagonalization appears at the very least competitive with the Schur reduction, and is considerably less expensive in many circumstances. In case in which unreduced Hessenberg structure

is lost, the bidiagonalization algorithm requires further study. In all cases, a more thorough study of the adaptive time stepping strategy is warranted.

References

[1] E. Allgower and K. Georg. Continuation and path following. *Acta Numerica*, pages 1–64, 1993.

[2] M. Baumann and U. Helmke. Diagonalization of time varying symmetric matrices. *Preprint, Univ. Wurzburg*, 2002.

[3] C. Bavely and G. W. Stewart. An algorithm for computing reducing subspaces by block–diagonalization. *SIAM J. Numer. Anal.*, 16:359–367, 1979.

[4] W. J. Beyn. The numerical computation of connecting orbits in dynamical systems. *IMA J. Numer. Analysis*, 10:379–405, 1990.

[5] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nichols. Numerical computation of an analytic singular value decomposition by a matrix valued function. *Numer. Math.*, 60:1–40, 1991.

[6] J. W. Demmel. Three methods for refining estimates of invariant subspaces. *Computing*, 38:43–57, 1987.

[7] J. W. Demmel, L. Dieci, and M. Friedman. An efficient algorithm for locating and continuing connecting orbits. *SIAM J. Scientific Computing*, 22:81–94, 2001.

[8] L. Dieci and T. Eirola. On smooth decomposition of matrices. *SIAM J. Matrix Anal. Appl.*, 20:800–819, 1999.

[9] L. Dieci and M. Friedman. Continuation of invariant subspaces. *Applied Numerical Linear Algebra*, 8:317–327, 2001.

[10] M. Friedman. An improved detection of bifurcations in large nonlinear systems via the continuation of invariant subspaces algorithm. *Int. J. Bifurc. & Chaos*, 11:2277–2285, 2001.

[11] H. Gingold and P.F. Hsieh. Globally analytic triangularization of a matrix function. *Linear Algebra Appl.*, 169:75–101, 1992.

[12] G. H. Golub, S. Nash, and C. Van Loan. A Hessenberg-Schur method for the problem $AX + XB = C$. *IEEE Trans. Auto. Control*, 24:909–913, 1979.

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 2nd edition, 1989.

[14] B. Kagström. Computation of matrix functions. Technical Report Report UMINF-58.77, Dept. Inf. Processing, Univ. of Umea, Umea, Sweden, 1977.

[15] T. Kato. *Perturbation Theory for Linear Operators*. Springer-Verlag, Berlin, 1976. 2nd edition.

[16] H. Keller. *Numerical Methods in Bifurcation Problems*. Springer Verlag, Bombay, 1987. Tata Institute of Fundamental Research.

[17] V. Mehrmann and W. Rath. Numerical methods for the computation of analytic singular value decompositions. *Electronic Trans. Numerical Analysis*, 1:72–88, 1993.

[18] W. Rheinboldt. On the computation of multi-dimensional solution manifolds of parametrized equations. *Numer. Math.*, 53:165–181, 1988.

[19] K. Wright. Differential equations for the analytical singular value decomposition of a matrix. *Numer. Math.*, 63:283, 1992.