Electronic Transactions on Numerical Analysis.
Volume 1, pp. 72-88, December 1993.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

NUMERICAL METHODS FOR THE COMPUTATION OF ANALYTIC SINGULAR VALUE DECOMPOSITIONS *

VOLKER MEHRMANN† AND WERNER RATH†

Dedicated to Wilhelm Niethammer on the occasion of his 60th birthday.

Abstract. An analytic singular value decomposition (ASVD) of a path of matrices $E(t)$ is an analytic path of factorizations $E(t) = X(t)S(t)Y(t)^T$ where $X(t)$ and $Y(t)$ are orthogonal and $S(t)$ is diagonal. The diagonal entries of $S(t)$ are allowed to be either positive or negative and to appear in any order. For an analytic path matrix $E(t)$ an ASVD exists, but this ASVD is not unique. We present two new numerical methods for the computation of unique ASVD's. One is based on a completely algebraic approach and the other on one step methods for ordinary differential equations in combination with projections into the set of orthogonal matrices.

Key words. analytic singular value decomposition, singular value decomposition.

AMS subject classification. 65F25.

1. Introduction.

The singular value decomposition (SVD) of a constant matrix $E \in \mathrm{R}^{m\times n}$, $m \geq n$, is a factorization $E = U\Sigma V^T$, where $U \in \mathrm{R}^{m\times m}$ and $V \in \mathrm{R}^{n\times n}$ are orthogonal and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_n) \in \mathrm{R}^{m\times n}$. The SVD is an important tool in numerous applications. It is well studied, and good numerical methods are available [1, 4, 6, 8, 10, 12].

In this paper we discuss the *analytic singular value decomposition (ASVD)*. For a real analytic matrix valued function $E(t) : [a, b] \to \mathrm{R}^{m\times n}$, an ASVD is a path of factorizations,

$$E(t) = X(t)S(t)Y(t)^T,$$

where $X(t) : [a, b] \to \mathrm{R}^{m\times m}$ is orthogonal, $S(t) : [a, b] \to \mathrm{R}^{m\times n}$ is diagonal, $Y(t) : [a, b] \to \mathrm{R}^{n\times n}$ is orthogonal and $X(t)$, $S(t)$ and $Y(t)$ are analytic.

In [3] Bunse–Gerstner et al. prove the existence of an ASVD and show that an ASVD is not unique. They establish uniqueness of an ASVD, $E(t) = X(t)S(t)Y(t)^T$, that minimizes the total variation (or arc length)

$$\mathrm{Vrn}\,(X(t)) := \int_a^b \|dX/dt\|_F \, dx \tag{1.1}$$

over all feasible choices $X(t)$ and minimizes $\mathrm{Vrn}\,(Y(t))$ subject to (1.1) being minimum. (Here $\|\ \|_F$ denotes the Frobenius norm.) It is shown in [3] that the left singular factor $X(t)$ of a minimum variation ASVD satisfies a set of differential equations, and how these can be used in combination with an Euler-like discretization method for the calculation of the minimum variation ASVD.

We will give a short overview over the results of [3], and then derive a new unique ASVD. This ASVD is based on algebraic transformations, and for its computation we do not have to solve differential equations.

In [14] Wright derives differential equations for the factors of the ASVD, and solves these using explicit Runge–Kutta methods. These matrix differential equations

* Research supported by DFG Project Me 790. Received October 20, 1993. Accepted for publication November 29, 1993. Communicated by L. Reichel.
† Fachbereich Mathematik, TU Chemnitz-Zwickau, PSF 964, D-09009 Chemnitz, FRG.

for the orthogonal factors have the special structure:

$$\dot{U}(t) = H(t)U(t) \tag{1.2}$$

with initial conditions $U(t_0)^T U(t_0) = I_n$, where $U(t),\ H(t) : [a,b] \to \mathrm{R}^{n\times n}$, $t_0 \in [a,b]$ and $H(t)$ is skew symmetric, i.e. $H(t)^T = -H(t)$ for all $t \in [a,b]$. The solution $U(t)$ of this type of initial value problems is an orthogonal matrix, i.e. $U(t)^T U(t) = I_n$ for all $t \in [a,b]$.

In [5] Dieci et al. study such equations and discuss two types of numerical methods which preserve the orthogonality of the approximation during the integration. The first type are the so–called automatic orthogonal integrators and in [15] Wright uses implicit Runge–Kutta methods of this type. In the second part of this paper we discuss methods of the second type, the projective orthogonal integrators. These methods are based on explicit Runge–Kutta methods as a nonorthogonal integrator and on a projection of the approximation onto the set of the orthogonal matrices.

Finally we present some numerical results and compare different methods for computing an ASVD.

2. Notation and Preliminaries. In this section we introduce some notation and give a short review of the results in [3].

2.1. Notation. Throughout the remainder of this paper, we make the simplifying assumption that $m \geq n$. The case $m < n$ is similar, using the transpose of the matrix.

We denote by

- $\mathrm{R}^{m\times n}$ the real $m \times n$ matrices;
- $\mathcal{U}_{m,n}$ the set of real $m \times n$ matrices with orthonormal columns;
- $\mathcal{A}_{m,n}([a,b])$ the set of real analytic functions on $[a,b]$ with values in $\mathrm{R}^{m\times n}$;
- $\mathcal{D}_{m,n}$ the set of diagonal matrices in $\mathrm{R}^{m\times n}$;
- $\mathcal{P}_n$ the set of permutation matrices in $\mathrm{R}^{n\times n}$;
- I_n the $n \times n$ identity matrix;
- $< A, B >= \mathrm{Trace}(A^T B)$ the Frobenius inner product for $A, B \in \mathrm{R}^{m\times n}$;
- $\|A\|$ the Frobenius norm, i.e. $\|A\| := \sqrt{< A, A >}$ for $A \in \mathrm{R}^{m\times n}$;
- $\dot{A}(t)$ the first derivative for $A(t) \in \mathcal{A}_{m,n}([a,b])$;
- $A^{(j)}(t)$ the j–th derivative for $A(t) \in \mathcal{A}_{m,n}([a,b])$.

Algorithms in this paper are presented in MATLAB notation, see [10].

2.2. Existence and Uniqueness. In [3] it is shown that if $E(t) \in \mathcal{A}_{m,n}([a,b])$, then there exists an ASVD, but this ASVD is not unique. If $E(t)$ has two ASVD's

$$E(t) = X(t)S(t)Y(t)^T = \hat{X}(t)\hat{S}(t)\hat{Y}(t)^T,$$

then the two ASVD's are called *equivalent*. If, in addition, $S(t) = \hat{S}(t)$ then they are called *parallel*. Two ASVD's are equivalent if and only if there exists a matrix $Q_L(t) \in \mathcal{U}_{m,m} \cap \mathcal{A}_{m,m}$ and a matrix $Q(t) \in \mathcal{U}_{n,n} \cap \mathcal{A}_{n,n}$ such that

$$\hat{X}(t) = X(t)Q_L(t),$$

$$\hat{S}(t) = Q_L(t)^T S(t)Q(t)$$

and

$$\hat{Y}(t) = Y(t)Q(t).$$

$Q(t)$ and $Q_L(t)$ have a special structure, since both $S(t)$ and $\hat{S}(t)$ must be diagonal, and we can build them as a product of three simpler equivalences. These are permutation–equivalence (P–equivalence), sign–equivalence (D–equivalence) and orthogonal–equivalence (Z–equivalence).

Two ASVD's, $E(t) = X(t)S(t)Y(t)^T = \hat{X}(t)\hat{S}(t)\hat{Y}(t)^T$, are P-equivalent if there exists a $P \in \mathcal{P}_n$ and a permutation collaborator $P_L \in \mathcal{P}_m$ such that

$$\hat{X}(t) = X(t)P_L,$$

$$\hat{Y}(t) = Y(t)P$$

and

$$\hat{S}(t) = P_L^T S(t)P.$$

$P_L \in \mathcal{P}_m$ is a permutation collaborator of $P \in \mathcal{P}_n$ if P_L is of the form

$$P_L = \begin{bmatrix} P & 0 \\ 0 & I_{m-n} \end{bmatrix}.$$

Two ASVD's are D-equivalent if there exists a $D \in \mathcal{D}_{n,n} \cap \mathcal{U}_{n,n}$ such that

$$\hat{Y}(t) = Y(t)D$$

and

$$\hat{S}(t) = S(t)D.$$

A diagonal factor $S(t) \in \mathcal{D}_{m,m} \cap \mathcal{A}_{m,m}$ is called *gregarious* if $S(t)$ has the form

$$S(t) = \begin{bmatrix} s_1(t)I_{m_1} & 0 & \cdots & 0 \\ 0 & s_2(t)I_{m_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_k(t)I_{m_k} \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{2.1}$$

where $s_1(t),\ s_2(t), \ldots, s_{k-1}(t)$ are distinct, nonzero, analytic functions and $s_k \equiv 0$. An ASVD $E(t) = X(t)S(t)Y(t)^T$ is *gregarious* if its diagonal factor $S(t)$ is gregarious. A point t_1 with $s_i(t_1) \neq \pm s_j(t_1)$ for all $i, j \in \{1, \ldots, n\},\ i \neq j$ is called a *generic* point.

The third equivalence is nonconstant. Two parallel ASVD's are Z-equivalent if $S(t)$ is gregarious (with structure (2.1)) and there is a $Z(t) \in \mathcal{B}_S \cap \mathcal{U}_{n,n}$ and a collaborator $Z_L \in \mathcal{U}_{m,m} \cap \mathcal{B}_{L,S}$ such that

$$\hat{X}(t) = X(t)Z_L(t)$$

and

$$\hat{Y}(t) = Y(t)Z(t).$$

Here $Z(t) \in \mathcal{B}_S$ if and only if it is of the form

$$Z(t) = \begin{bmatrix} Z_1 & 0 & \cdots & 0 & 0 \\ 0 & Z_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & Z_{k-1} & 0 \\ 0 & 0 & \cdots & 0 & T_k \end{bmatrix}, \tag{2.2}$$

ETNA
Kent State University
etna@mcs.kent.edu

with $Z_j \in \mathrm{R}^{m_j \times m_j}$ for $j = 1, 2, \ldots, k-1$ and $T_k \in \mathrm{R}^{m_k \times m_k}$, and $Z_L \in \mathcal{U}_{m,m} \cap \mathcal{B}_{L,S}$ is a *collaborator* of $Z(t)$ if $Z_L(t)$ is of the form

$$Z_L(t) = \begin{bmatrix} Z_1 & 0 & \cdots & 0 & 0 \\ 0 & Z_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & Z_{k-1} & 0 \\ 0 & 0 & \cdots & 0 & Z_k \end{bmatrix}, \tag{2.3}$$

where the blocks $Z_1, Z_2, \ldots, Z_{k-1}$ are as in (2.2), $\hat{m} = m - \sum_{j=1}^{k-1} m_j$ and $Z_k \in \mathrm{R}^{\hat{m} \times \hat{m}}$.

The main results of [3] are

- Every ASVD can be transformed into a gregarious ASVD by a sequence of **constant** P– and D–equivalences.
- Any ordering and choice of signs can be obtained.
- A matrix $E(t) \in \mathcal{A}_{m,n}$ has two parallel, gregarious ASVD's if and only if the two paths are Z–equivalent.
- The ordering and the signs of the singular values are not unique, but they are uniquely determined by a constant initial SVD in a generic point.
- Suppose that t_0 and t_1 are generic points of $E(t) \in \mathcal{A}_{m,n}$, $E(t_0) = U_0 \Sigma_0 V_0^T$ and $E(t_1) = U_1 \Sigma_1 V_1^T$ are gregarious, constant SVD's. Then there exists an ASVD and P– and D–equivalences such that

  $$X(t_0) = U_0, \; S(t_0) = \Sigma_0, \; Y(t_0) = V_0$$

  and

  $$X(t_1) = U_1 P_{L_1} D_L, \; S(t_1) = P_{L_1}^T \Sigma_1 D_1 P_1, \; Y(t_1) = V_1 D_1 P_1 D_R.$$

The P– and D–equivalences fix the order and signs of the singular values, and the Z–equivalence describes the freedom of choice in the left and right singular factors for multiple singular values. If $E(t)$ has only simple singular values, an ASVD is uniquely determined by the initial conditions $U(t_0) = U_0, \; S(t_0) = \Sigma_0$ and $V(t_0) = V_0$ for a constant SVD, $E(t_0) = U_0 \Sigma_0 V_0^T$, at a generic point t_0.

2.3. Minimum Variation ASVD. In [3] the total variation

$$\mathrm{Vrn}\,(X(t)) := \int_a^b \|dX/dt\|_F \, dx$$

is used to produce a unique ASVD in the case that $E(t)$ has multiple singular values.

Suppose that $E(t) \in \mathcal{A}_{m,n}([a,b])$ and $t_0 \in [a,b]$ is generic point. If $E(t_0) = U \Sigma V^T$ is a given, constant SVD, then there exists a unique ASVD with $t \in [a,b]$, $E(t) = \hat{X}(t) S(t) \hat{Y}(t)^T$, such that $\hat{X}(t_0) = U, \; \hat{S}(t_0) = \Sigma, \; \hat{Y}(t_0) = V$,

$$\mathrm{Vrn}\left(\hat{X}(t)\right) = \min \left\{ \mathrm{Vrn}\,(X(t)) \;\middle|\; \begin{array}{c} E(t) = X(t) S(t) Y(t)^T \\ \text{is an ASVD} \end{array} \right\} \tag{2.4}$$

and

$$\mathrm{Vrn}\left(\hat{Y}(t)\right) = \min \left\{ \mathrm{Vrn}\,(Y(t)) \;\middle|\; \begin{array}{c} E(t) = \hat{X}(t) S(t) Y(t)^T \\ \text{is an ASVD} \\ \text{subject to (2.4)} \end{array} \right\}. \tag{2.5}$$

ETNA
Kent State University
etna@mcs.kent.edu

If $E(t) = X(t)S(t)Y(t)^T$ is a gregarious ASVD on $[a, b]$ with initial conditions $X(t_0) = U,\ S(t_0) = \Sigma$ and $V(t_0) = V$, then $\hat{X}(t) = X(t)Z_L(t)$ where $Z_L(t) \in B_{L,S}$ solves the initial value problem

$$\dot{Z}_L(t) = \Phi_{L,S}\left(\dot{X}(t)^T X(t)\right) Z_L(t), \quad Z_L(t_0) = I_m, \tag{2.6}$$

where $\Phi_{L,S}$ denotes the projection in the Frobenius inner product $< \cdot, \cdot >_F$ from $\mathcal{A}_{m,m}$ onto $B_{L,S}$.

Expanding $Z_L(t)$ in a Taylor series arround t_0 and using a first order difference approximation for $\dot{X}(t)$ an Euler-like method discretization method for the solution of (2.6) is derived in [3]. For this method one obtains the usual results for ODE's, i.e. $\mathcal{O}(h^2)$ local error and $\mathcal{O}(h)$ global error. It is not known how to produce higher order methods using similar procedures, but one can extrapolate $\hat{X}(t)$, see [3].

3. Algebraic Methods. In this section we introduce a new algebraic method which is based on polar decompositions.

If we have two parallel, gregarious ASVD's of a matrix $E(t) \in \mathcal{A}_{m,n}([a, b])$, i.e. $E(t) = X(t)S(t)Y(t)^T = \hat{X}(t)S(t)\hat{Y}(t)^T$, then the two ASVD's are Z–equivalent and we know that

$$X(t) = \begin{bmatrix} X_{11}(t) & X_{12}(t) & \dots & X_{1k}(t) \\ X_{21}(t) & X_{22}(t) & \dots & X_{2k}(t) \\ \vdots & \vdots & \ddots & \vdots \\ X_{k1}(t) & X_{k2}(t) & \dots & X_{kk}(t) \end{bmatrix} \tag{3.1}$$

$$= \hat{X}(t)Z_L(t), \tag{3.2}$$

and

$$Y(t) = \begin{bmatrix} Y_{11}(t) & Y_{12}(t) & \dots & Y_{1k}(t) \\ Y_{21}(t) & Y_{22}(t) & \dots & Y_{2k}(t) \\ \vdots & \vdots & \ddots & \vdots \\ Y_{k1}(t) & Y_{k2}(t) & \dots & Y_{kk}(t) \end{bmatrix} \tag{3.3}$$

$$= \hat{Y}(t)Z(t), \tag{3.4}$$

where $Z(t) \in \mathcal{B}_S \cap \mathcal{U}_{n,n}$ and $Z_L(t) \in \mathcal{B}_{L,S} \cap \mathcal{U}_{m,m}$ are collaborators of a Z–equivalence. The matrices $X(t)$ and $Y(t)$ are partitioned as $Z_L(t)$ and $Z(t)$ in (2.2) and (2.3) , so that $X_{ii}(t), Y_{ii}(t) \in \mathcal{A}_{m_i,m_i}$ for $i = 1, \dots, k-1$, $X_{kk}(t) \in \mathcal{A}_{m_k,m_k}$, $Y_{kk}(t) \in \mathcal{A}_{\tilde{m}_k,\tilde{m}_k}$ and the off diagonal blocks have corresponding dimensions. If we partition $\hat{X}(t)$ and $\hat{Y}(t)$ in a similar way we get the block equations

$$X_{ii}(t) = \hat{X}_{ii}(t)Z_i(t); \quad i = 1, \dots, k; \tag{3.5}$$

$$X_{ji}(t) = \hat{X}_{ji}(t)Z_i(t); \quad i, j = 1, \dots, k;\ i \neq j; \tag{3.6}$$

and

$$Y_{ii}(t) = \hat{Y}_{ii}(t)Z_i(t); \qquad i = 1, \dots, k-1; \tag{3.7}$$

$$Y_{ji}(t) = \hat{Y}_{ji}(t)Z_i(t); \qquad i, j = 1, \dots, k-1;\ i \neq j; \tag{3.8}$$

$$Y_{kk}(t) = \hat{Y}_{kk}(t)T_k(t); \tag{3.9}$$

$$Y_{jk}(t) = \hat{Y}_{jk}(t)T_k(t); \qquad j = 1, \dots, k-1. \tag{3.10}$$

The idea of the algebraic method is to choose a decomposition of the type

$$A(t) = B(t)Q(t), \tag{3.11}$$

where $A(t), B(t) \in \mathcal{A}_{l,l}([a,b])$ and $Q(t) \in \mathcal{U}_{l,l} \cap \mathcal{A}_{l,l}([a,b])$ for the diagonal blocks so that $\hat{X}_{ii}(t),\ i = 1,\dots,k$ and $\hat{Y}_{kk}(t)$ are independent of the *a priori* ASVD.

In this paper we present a method based on the polar decomposition. Other methods based on QR–decompositions and ASVD's are discussed in [11].

3.1. The Analytic Polar Decomposition. The polar decomposition of a matrix $A \in \mathrm{R}^{m\times n}$ is usually defined as a decomposition of the form $A = QP$, where $Q \in \mathrm{R}^{m\times n}$, $Q^TQ = I_n$, $P \in \mathrm{R}^{n\times n}$ and where P is symmetric and positive semidefinite (see [8, p. 148]). In a similar way we can define a decomposition $A = \hat{P}\hat{Q}$, where $\hat{Q} \in \mathrm{R}^{m\times n}$, $\hat{Q}^T\hat{Q} = I_n$, $\hat{P} \in \mathrm{R}^{m\times m}$ and where $\hat{P}$ is symmetric and positive semidefinite. For our method we require this second type of polar decomposition for $A(t) \in \mathcal{A}_{n,n}([a,b])$.

In an ASVD we must allow the singular values to appear in any order and to change sign. For the analytic polar decomposition we get a similar result when $P(t)$ is symmetric, but not necessary positive semidefinite.

Definition 1. *For a real analytic matrix valued function $E(t) \in \mathcal{A}_{n,n}([a,b])$, an* **analytic polar decomposition** *(APD) is a path of factorizations*

$$A(t) = P(t)Q(t)$$

where

$$Q(t) \in \mathcal{U}_{n,n} \cap \mathcal{A}_{n,n}([a,b])$$

and

$$P(t) \in \mathcal{A}_{n,n}([a,b]) \text{ and } P(t) = P(t)^T.$$

The close relationship between an ASVD and APD is given by the following theorem.

Theorem 2. *If $A(t) \in \mathcal{A}_{n,n}([a,b])$, then there exists an APD on $[a,b]$.*

Proof: If $A(t) \in \mathcal{A}_{n,n}([a,b])$, then there exists an ASVD

$$A(t) = X(t)S(t)Y(t)^T$$

on $[a,b]$. If we set

$$Q(t) := X(t)Y(t)^T$$

and

$$P(t) := X(t)S(t)X(t)^T,$$

then $Q(t) \in \mathcal{U}_{n,n} \cap \mathcal{A}_{n,n}([a,b])$, $P(t) \in \mathcal{A}_{n,n}([a,b])$ and $P(t)$ is symmetric. Hence

$$\begin{aligned} P(t)Q(t) &= \left(X(t)S(t)X(t)^T\right)\left(X(t)Y(t)^T\right) \\ &= X(t)S(t)Y(t)^T \\ &= A(t) \end{aligned}$$

is an APD of $A(t)$. □

As mentioned before, an ASVD is not unique, and we have some freedom in the construction of an ASVD as shown in Section 2.2. The situation for the APD is different. For a **nonsingular** constant matrix $A \in \mathrm{R}^{n\times n}$ there is a unique polar decomposition $A = PQ$, where P is symmetric and positive definite (see [7, p. 614]). The following theorem is the corresponding result for the APD.

THEOREM 3. *If $A(t) \in \mathcal{A}_{n,n}([a,b])$, $t_0 \in [a,b]$ and $A(t_0)$ is nonsingular, then there exists a unique APD,*

$$A(t) = P(t)Q(t),$$

on $[a,b]$, such that $P(t_0)$ is positive definite.

Proof: Let

$$A(t) = \hat{X}(t)\hat{S}(t)\hat{Y}(t)^T \tag{3.12}$$

be a gregarious ASVD with positive singular values in t_0. We know from Section 2.2 that such an ASVD exists, and as in the proof of Theorem 2 we can define the corresponding APD,

$$A(t) = \hat{P}(t)\hat{Q}(t). \tag{3.13}$$

Since $S(t_0)$ has positive diagonal entries, the matrix $\hat{P}(t_0)$ is positive definite.

Assume that there exists a second APD $A(t) = P(t)Q(t)$ such that $P(t_0)$ is positive definite. In [9, p. 120–122] it is shown that there exists an orthogonal matrix $X(t) \in \mathcal{A}_{n,n}([a,b])$ and a diagonal matrix $S(t) \in \mathcal{A}([a,b])_{n,n}$ such that

$$P(t) = X(t)S(t)X(t)^T.$$

Since $P(t_0)$ is positive definite, the diagonal matrix $S(t_0)$ has positive entries. If we set

$$Y(t) = Q(t)^T X(t),$$

then $Y(t) \in \mathcal{A}_{n,n}([a,b])$, and since

$$\begin{aligned} A(t) &= P(t)Q(t) \\ &= \left(X(t)S(t)X(t)^T\right)\left(X(t)Y(t)^T\right) \\ &= X(t)S(t)Y(t)^T, \end{aligned}$$

we get a second ASVD of $A(t)$.

The ASVD (3.12) is gregarious, and from [3, Corollary 6] it follows that

$$\hat{Y}(t) = Y(t)\left(DPZ(t)\right), \tag{3.14}$$

$$\hat{X}(t) = X(t)\left(P_L Z_L(t)\right) \tag{3.15}$$

and

$$\hat{S}(t) = P_L^T S(t) DP, \tag{3.16}$$

where $D \in \mathcal{D}_{n,n}$ is a constant diagonal matrix with ± 1's on the diagonal. $P,\ P_L \in \mathcal{P}_{n,n}$ are permutation collaborators, and $Z,\ Z_L \in \mathcal{A}_{n,n}([a,b]) \cap \mathcal{U}_{n,n}$ are the collaborators of a Z–equivalence.

Since $P(t_0)$ and $\hat{P}(t_0)$ are positive definite, $S(t_0)$ and $\hat{S}(t_0)$ are nonsingular, and from (2.2) and (2.3) it follows that $Z(t) = Z_L(t)$. Using (3.14) and (3.15) for $\hat{Q}$ as in (3.13), we get

$$\begin{aligned} \hat{Q}(t) &= \hat{X}(t)\hat{Y}(t)^T \\ &= X(t)\left(P_L Z_L(t)\right)\left(Z(t)^T P^T D\right) Y(t)^T \\ &= X(t) D Y(t)^T. \end{aligned}$$

Finally, we find for $\hat{P}(t)$ that

$$\hat{P}(t) = X(t)S(t)DX(t)^T. \tag{3.17}$$

Since $P(t_0)$ is positive definite, i.e. $S(t_0)$ has positive diagonal entries, it follows that $D = I_n$, and thus

$$\hat{P}(t) = P(t) \text{ and } \hat{Q}(t) = Q(t). \quad \Box \tag{3.18}$$

Corollary 3.1. *If $A(t) \in \mathcal{A}_{n,n}([a,b])$, $t_0 \in [a,b]$, $A(t_0)$ is nonsingular and $A(t_0) = PQ$ is a given constant polar decomposition, then there exists a unique APD, $A(t) = P(t)Q(t)$, that interpolates the constant polar decomposition at t_0, i.e. $P(t_0) = P$ and $Q(t_0) = Q$.*

The following theorem is the main result of this section. It shows how we can use the unique APD of Theorem 3 to get a unique ASVD.

Theorem 4. *Let $E(t) \in \mathcal{A}_{m,n}([a,b])$ and let $E(t_0) = U\Sigma V^T$ be a gregarious constant SVD at a generic point $t_0 \in [a,b]$, where the blocks $U_{ii},\ i = 1,\ldots,k$ and V_{kk} are symmetric and positive definite. Then there exists a unique ASVD, $E(t) = X(t)S(t)Y(t)^T$, such that $X(t_0) = U$, $S(t_0) = \Sigma$, $Y(t_0) = V$ and the matrices $X_{ii}(t)\ i = 1,\ldots,k$ and $Y_{kk}(t)$ are symmetric.*

Proof: Let $E(t) = X(t)S(t)Y(t)^T$ be an arbitrary ASVD that interpolates the gregarious, constant SVD, $E(t_0) = U\Sigma V^T$. Let $X(t)$ and $Y(t)$ be in the form (3.1) and (3.3). Since the diagonal blocks of $X(t)$ and $Y(t)$ are analytic, there exist APD's $X_{ii}(t) = Q_i(t)\hat{X}_{ii}(t),\ i = 1,\ldots,k$ and $Y_{kk}(t) = \hat{Q}(t)\hat{Y}_{kk}(t)$, where $Q_i(t_0),\ i = 1,\ldots,k$ and $\hat{Q}_k(t_0)$ are identity matrices. If we set $Z_L(t) := \mathrm{diag}(Q_1(t),\ldots,Q_k(t))$ and $Z(t) := \mathrm{diag}(Q_1(t),\ldots,Q_{k-1}(t),\hat{Q}(t))$, then we get the ASVD,

$$E(t) = \hat{X}(t)S(t)\hat{Y}(t), \tag{3.19}$$

where $\hat{X}(t) = X(t)Z_L(t)$ and $\hat{Y}(t) = Y(t)Z(t)$. These two ASVD's are now Z-equivalent, the ASVD (3.19) interpolates the constant SVD, $E(t_0) = U\Sigma V^T$, and the diagonal blocks $\hat{X}_{ii}(t),\ i = 1,\ldots,k$ and $\hat{Y}_{kk}(t)$ are symmetric.

To show that the ASVD (3.19) is unique, assume that there exists a second ASVD,

$$E(t) = \tilde{X}(t)S(t)\tilde{Y}(t), \tag{3.20}$$

which interpolates in t_0 and has symmetric diagonal blocks. Since t_0 is a generic point, the two ASVD's (3.19) and (3.20) are Z–equivalent, and we denote the collaborators of this Z–equivalence by $\tilde{Z}$ and $\tilde{Z}_L$. As in Section 3 we then get the block equations $\tilde{X}_{ii}(t) = \hat{X}_{ii}(t)\tilde{Z}_i(t),\ i = 1,\ldots,k$ and $\tilde{Y}_{kk}(t) = \hat{Y}_{kk}(t)\tilde{T}_k(t)$. All these equations are of the form $\tilde{A}(t) = \hat{A}(t)Q(t)$ where $Q(t)$ is orthogonal, $\tilde{A}(t)$ and $\hat{A}(t)$ are symmetric and $\tilde{A}(t_0) = \hat{A}(t_0)$, since both ASVD's (3.19) and (3.20) interpolate the constant SVD at t_0. Furthermore, since the diagonal blocks of the constant SVD, $E(t_0) = U\Sigma V^T$, are positive definite, $\tilde{A}(t_0)$ and $\hat{A}(t_0)$ are positive definite, and from Theorem 3 we then get that $\tilde{A}(t) = \hat{A}(t)$ on $[a, b]$. This means that $\tilde{Z}$ and $\tilde{Z}_L$ are identity matrices, and thus the ASVD is unique. □

In Theorem 4 we need the nonsingularity of all diagonal blocks U_{ii}, $i = 1, \ldots, k$, and V_{kk}. For gregarious ASVD's it is possible that one of this blocks is singular for all points $t \in [a, b]$ or only at the initial point t_0. In [3] the gregarious ASVD's are used to simplify the proofs of the theorems. In [11] a *weakly gregarious* ASVD is introduced that allows the singular values to appear in any order and multiple singular values have still the same sign. Then one obtains a new type of Z–equivalence, and the results of [3] are still valid for this weakly gregarious ASVD. In many cases it is then possible to find a weakly gregarious, constant SVD that satisfies the conditions of a version of Theorem 4, where gregarious is replaced by weakly gregarious.

3.2. A numerical Method.. Our algebraic method computes an approximation, $A(t) = X(t)S(t)Y(t)^T$, of the unique algebraic ASVD of Theorem 4 at points $t_i,\ i = 0, \ldots, N$. We implemented constant and variable stepsize codes for computing this algebraic ASVD. In the variable stepsize code we choose the stepsize so that the code avoids nongeneric points (except a simple singular value equal to zero) and points t_j where one of the diagonal blocks $X_{ii}(t_j)$ or $Y_{kk}(t_j)$ defined in Section 3 corresponding to a multiple singular values becomes singular.

In the constant stepsize case we have to distinguish two cases. Nongeneric points where only one singular value becomes zero do not cause any difficulties. In this case we can compute a high accuracy approximation of the unique algebraic ASVD. Since we use standard constant SVD's at points t_i in the process of computing the ASVD via the algebraic approach, we have difficulties obtaining high accuracy approximations at points where two singular value paths intersect at a nongeneric point. The standard SVD treats the two singular values at this point as a multiple singular values, and it is difficult to resolve the paths correctly so as to stay on the algebraic ASVD. To accomplish this we solve an orthogonal Procustes problem [8], i.e. we compute an SVD in t_i that is closest to the SVD in $t_{i-1} = t_i - h$ in the Frobenius norm. This SVD is then an $\mathcal{O}(h)$ approximation to the SVD that lies on the algebraic ASVD.

For simplicity we describe our algorithm for a gregarious ASVD and a sequence of nongeneric points $t_i,\ i = 0, \ldots, N$.

In an initial step, using standard methods, we calculate a constant SVD, $E(t_0) = \hat{U}_0\hat{\Sigma}_0\hat{V}_0^T$. Then we compute polar decompositions of the diagonal blocks corresponding to multiple singular values, and in this way obtain an SVD,

$$E(t_0) = U_0\Sigma_0 V_0^T,$$

where the diagonal blocks satisfies the conditions of Theorem 4.

For each nongeneric point $t_{i+1},\ i = 0, \ldots, N-1$, we then repeat the following procedure:

1. We use standard methods to get an SVD $E(t_{i+1}) = \hat{U}_1\hat{\Sigma}_1\hat{V}_1$ at a generic point $t_{i+1} = t_i + h_i$ close to t_i (note that in the case that the original t_i was nongeneric, we have to add extra points).
2. We then use Theorem 7 from [3] to adjust the SVD in t_{i+1} so that the new SVD $E(t_{i+1}) = U_1\Sigma_1 V_1^T$ lies on an ASVD that interpolates $E(t_i) = U_0\Sigma_0 V_0^T$. This procedure is performed by the following matlab routine:

ALGORITHM 1.

Input: *A constant SVD* $E(t_i) = U_0\Sigma_0V_0^T$ *and a constant matrix* $E(t_{i+1}) \in \mathrm{R}^{m\times n}$ *at generic points* t_i *and* t_{i+1}.

Output: *A constant SVD* $E(t_{i+1}) = U_1\Sigma_1V_1^T$*, that lies on an interpolating ASVD.*

```
%%%%% 1. step: Calculate SVD E(t_i+1) = U_h S_h V^T_h
[uh,sh,vh] = svd(e1)
%%%%% 2. step: Calculate permutation matrices P_Lh and P_h
u = u0'*uh;
for i = 1:n
   [dummy,jbig] = max(abs(u(i,1:n)));
   if i ~= jbig,
      u(:,[i jbig]) = u(:,[jbig i]);

      uh(:,[i jbig]) = uh(:,[jbig i]);
      sh(:,[i jbig]) = sh(:,[jbig i]);
      sh([i jbig],:) = sh([jbig i],:);
      vh(:,[i jbig]) = vh(:,[jbig i]);
   end
end;
%%%%% 3. step: Calculate the diagonal matrices
for i = 1:n
    v(i) = v0(:,i)'*vh(:,i);
end;
s1 = sh * diag(sign(v));
v1 = vh * diag(sign(v));
s1 = diag(sign(diag(u))) * s1;
u1 = uh * diag(sign(diag(u)));
```

3. We compute polar decompositions of the diagonal blocks of U_1 and possibly the last diagonal block of V_1. If one of these diagonal blocks is singular and the dimension of this block is larger then one, we cannot determine a unique polar decomposition for this matrix, and instead we solve again an orthogonal Procustes problem. In this case we get only an $\mathcal{O}(h)$ approximation to the ASVD in this part of the singular factors.

For one time step of the computation of the algebraic ASVD we then have the following algorithm:

ALGORITHM 2.

Input: *A constant gregarious SVD,* $E(t_i) = U_0\Sigma_0V_0^T$*, at a generic point* t_i*, where the diagonal blocks of* U_0 *and the* k*–th diagonal block of* V_0 *are symmetric and nonsingular and a constant matrix* $E(t_{i+1}) \in \mathrm{R}^{m\times n}$ *at a generic point* t_{i+1}.

Output: *An approximation,* $E(t_{i+1}) = U_1\Sigma_1V_1^T$*, of the constant SVD that lies on the algebraic ASVD of Theorem 4 and interpolates* $E(t_i) = U_0\Sigma_0V_0^T$.

1. *Use Algorithm 1 with input,* $E(t_i) = U_0\Sigma_0V_0^T$ *and* $E(t_{i+1})$*. This calculates the constant SVD,* $E(t_{i+1}) = U\Sigma V^T$.
2. *Determine the multiplicities of the singular values of* Σ.
3. *Partition* U *and* V *into blocks as* $X(t)$ *and* $Y(t)$ *in (3.1) and (3.3).*
4. *For* $j = 1,\ldots,k$ *compute* $U_{jj} = P_jQ_j^T$*. If* P_j *is singular, then choose* Q_j *so that* Q_j *solves a special orthogonal Procustes problem.*

ETNA
Kent State University
etna@mcs.kent.edu

5. *If there are zero singular values, then compute* $V_{kk} = \tilde{P}_k \tilde{Q}_k^T$. *If* $\tilde{P}_k$ *is singular, then choose* $\tilde{Q}_k$ *so that* $\tilde{Q}_k$ *solves a special orthogonal Procustes problem.*
6. *Set* $Q = diag(Q_1, Q_2, \ldots, Q_k)$, $\tilde{Q} = diag(Q_1, Q_2, \ldots, Q_{k-1}, \tilde{Q}_k)$.
7. *Set* $U_1 \leftarrow UQ$, $\Sigma_1 \leftarrow \Sigma$ *and* $V_1 \leftarrow V\tilde{Q}$.

Remarks on Steps 4 and 5: We compute the polar decompositions with help of another SVD. This allows us to estimate the numerical rank of the blocks and to choose the correct polar decomposition.

If the dimension of a block is one, then the matrix Q_j is ± 1, and we choose the correct sign. Note that this is always possible, even if $P_j = 0$.

If the dimension of a block is larger than one and P_j (or $\tilde{P}_k$) is singular we solve the orthogonal Procustes problem, see [8],

$$\text{minimize } ||\bar{U}_{jj} - U_{jj}Qj||_F \text{ subject to } Q_j^T Q = I_{m_j},$$

where $\bar{U}_{jj}$ is the diagonal block of U_0 which corresponds to U_{jj}.

4. Differential Equations for the ASVD. If $E(t) \in \mathcal{A}_{m,n}([a,b])$ and $E(t) = X(t)S(t)Y(t)$ is an ASVD, then the factors satisfy the following set of differential equations:

$$\text{diag}\left(\dot{S}(t)\right) = \text{diag}\left(Q(t)\right), \tag{4.1}$$

$$\dot{X}(t) = X(t)Z(t), \tag{4.2}$$

$$\dot{Y}(t) = Y(t)W(t), \tag{4.3}$$

where $Q(t) = X(t)^T \dot{E}(t) Y(t)$ and $Z(t) = X(t)^T \dot{X}(t)$, $W(t) = Y(t)^T \dot{Y}(t)$ are skew symmetric. We get the initial values from the constant SVD, $E(t_0) = X(t_0)S(t_0)Y(t_0)^T$. This set of equations was derived by Wright in [14]. Let $S(t) := \text{diag}(\sigma_1, \ldots, \sigma_n)$, $Z(t) := [z_{ij}(t)]$, $W(t) := [w_{ij}(t)]$ and $Q(t) := [q_{ij}(t)]$.

If $n = m$ and $j > k$ we have the equations

$$\sigma_k(t) z_{j,k}(t) + \sigma_j(t) w_{k,j}(t) = q_{j,k}(t) \tag{4.4}$$

and

$$\sigma_j(t) z_{j,k}(t) + \sigma_k(t) w_{k,j}(t) = -q_{k,j}(t). \tag{4.5}$$

corresponding to the j, k position. Unless $\sigma_k(t)^2 \neq \sigma_j(t)^2$ we can solve these equations and get

$$z_{j,k}(t) = \frac{\sigma_k(t) q_{j,k}(t) + \sigma_j(t) q_{k,j}(t)}{\sigma_k(t)^2 - \sigma_j(t)^2} \tag{4.6}$$

and

$$w_{k,j}(t) = \frac{\sigma_j(t) q_{j,k}(t) + \sigma_k(t) q_{k,j}(t)}{\sigma_j(t)^2 - \sigma_k(t)^2}. \tag{4.7}$$

If $m > n$ we have the additional equations

$$\sigma_k(t) z_{j,k}(t) = q_{j,k}(t),\ j = n+1, \ldots, m;\ k = 1, \ldots, n, \tag{4.8}$$

and we can solve these equations for $z_{j,k}$ if $\sigma_k(t) \neq 0$. This leaves the components $z_{j,k}$, $j = n+1, \ldots, m$; $k = n+1, \ldots, j-1$ undetermined. Note that $\sigma_k(t) = 0$ does not cause any problems for $n = m$.

Equations (4.2) and (4.3) have a special structure, since

$$\dot{U}(t) = U(t)H(t),\ U(t_0)^T U(t_0) = I_n, \tag{4.9}$$

where $H(t), U(t) \in \mathcal{A}_{l,l}([a,b]),\ l \in \{n, m\}$ and where $H(t)$ is skew symmetric. The solution $U(t)$ of the initial value problem (4.9) is orthogonal, i.e. $U(t)^T U(t) = I_l$ for all $t \in [a,b]$.

Differential equations of this type are studied in [5]. It is shown there that two types of methods preserve the orthogonality of $U(t)$ during the integration. These are the so called automatic and projective–orthogonal integrators. Another integrator, which also preserves the orthogonality was recently introduced in [2].

Here we use projective–orthogonal integrators for solving the differential equations (4.1)–(4.3). If U_i is a given orthogonal approximation of $U(t_i)$ we use an explicit Runge–Kutta method to compute a nonorthogonal approximation $\hat{U}_{i+1}$ of $U(t_{i+1})$ for $t_{i+1} = t_i + h$. Then we compute a QR decomposition $\hat{U}_{i+1} = U_{i+1}R_{i+1}$, where R_{i+1} has positive diagonal elements, and we use U_{i+1} as orthogonal approximation to $U(t_{i+1})$. If the stepsize h is small enough, $\hat{U}_{i+1}$ is nonsingular, since it is an $\mathcal{O}(h^p)$ approximation to the orthogonal matrix $U(t_{i+1})$ (p is the order of the Runge–Kutta method). Therefore $\hat{U}_{i+1}$ has a small condition number if h is small, and we can use the modified Gram–Schmidt algorithm, see [8], to compute U_{i+1}.

If $m > n$ or if $E(t)$ has multiple singular values, then not all entries of $Z(t)$ and $W(t)$ are determined by (4.1)–(4.3). A simple strategy is to set all these values in $Z(t)$ to zero and to compute the corresponding values of $W(t)$ from equation (4.4). Another problem appears if the squares of two singular values intersect at a point t_i, since we cannot determine the corresponding values of $Z(t)$ and $W(t)$ from equation (4.4) and (4.5). If two singular values are close in modulus we use the old values for the corresponding values of $Z(t)$, and we compute the values of $W(t)$ again from equation (4.4). Finally if $m > n$ and a singular value is close to zero we use the old values for the undetermined entries in $Z(t)$.

Combining all these cases, we initialize the matrices $Z(t)$ and $W(t)$ at t_0 by setting them to zero and we compute $Q(t),\ Z(t)$ and $W(t)$ with the following algorithm. In this algorithm we use a cut–off tolerance `ctol` to test whether two singular values are approximately equal in modulus or one singular value is close to zero.

ALGORITHM 3.

Input: *A constant SVD, $E(t_i) = U_0 S_0 V_0^T$, of $E(t)$ at a point t_i, the matrices Z_0 and W_0 at t_{i-1}, the first derivative* `edash` *of $E(t)$ at t_i, and a cut–off tolerance* `ctol`.

Output: *Approximations for the values of Q_0, Z_0 and W_0 at t_i.*

```
%%%%% Compute q0
q0 = u0' * edash * v0;
%%%%% Compute w0 and z0:
for j=1:n
  for k=j+1:n
   if abs(abs(s0(k,k))-abs(s0(j,j))) <= ctol
     if abs(s0(k,k)) > ctol
       z0(j,k) =  q0(j,k)/s0(k,k) - w0(k,j);
       z0(k,j) = -z0(j,k);
     end;
   else
      sqq     =  (s0(j,j)+s0(k,k))*(s0(j,j)-s0(k,k));
```

```
      z0(j,k) = -(s0(k,k)*q0(j,k)+s0(j,j)*q0(k,j))/sqq;
      z0(k,j) = -z0(j,k);
      w0(k,j) =  (s0(j,j)*q0(j,k)+s0(k,k)*q0(k,j))/sqq;
      w0(j,k) = -w0(k,j);
   end;
  end;
end;
%%%%% If m > n, then compute the rest of z0
for k=1:n,
  if abs(s0(k,k)) > ctol,
    for j=n+1:m,
      z0(j,k) = q0(j,k)/s0(k,k);
      z0(k,j) = -z0(j,k);
    end;
  end;
end;
```

We implemented fix and variable stepsize codes using Algorithm 3. These codes compute approximations in nongeneric points with higher accuracy than the Euler–like method of [3] or the algebraic method described in Section 3. Like other similar methods they also can start only at a nongeneric point, provided the initial value SVD lies on an ASVD.

These codes tend to get unstable if there are many nongeneric points in the integration interval.

5. Numerical Results. We tested and compared different methods for computing an ASVD using variable stepsize codes.

For the minimal variation ASVD code we use stepsize control for ODE's and keep the difference between the output orthogonal factors $X(t_{i+1})$ and $Y(t_{i+1})$ and the input orthogonal factors $X(t_i)$ and $Y(t_i)$ in the Frobenius norm smaller then the *ad–hoc* constant 1/2, i.e. $||M(t_i)-M(t_{i+1})||_F < 0.5$ where $M = X, Y$. The ODE stepsize control is the usual strategy (see for example [13]). We compute an approximation for $X(t_{i+1})$ and $Y(t_{i+1})$ with stepsizes h and $h/2$ and estimate the error. From this estimation we obtain a new stepsize and if the ratio of the old and new stepsize is smaller than 3, we accept the step. If not, we compute new approximations with the new stepsize.

For the algebraic method we adjust the stepsize so that the difference between the output and input orthogonal factors is again smaller then 1/2, i.e. $||M(t_i) - M(t_{i+1})||_F < 0.5$, and we increase the stepsize in the next step if $0.125 < ||M(t_i) - M(t_{i+1})||_F$ where $M = X, Y$.

The variable stepsize codes for the method that solves ODE's (4.1) – (4.3) uses only the stepsize control for ODE's from [13].

All codes are implemented in MATLAB [10] and the examples were run on a SPARCstation 2. The unit roundoff of this machine is approximately 10^{-16}. In the following tables *fnval* indicates the number of function evaluations and *kflops* the number of floating point operations (times 1024). Furthermore S, X and E denote the Frobenius norm error of the singular values, the left singular factor $X(t)$ and the recomputed matrix $E(t)$, respectively.

5.1. Example 1. The first example is Example 2 from [14]:

$$E(t) = U(t)S(t)U(t),$$

where

$$U(t) = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & c_2 & s_2 & 0 \\ 0 & -s_2 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & c_3 & s_3 \\ 0 & 0 & -s_3 & c_3 \end{bmatrix}$$

with $c_1 = \cos(t)$, $s_1 = \sin(t)$, $c_2 = \cos(t+1)$, $s_2 = \sin(t+1)$, $c_3 = \cos(t+2)$, $s_3 = \sin(t+2)$ and

$$S(t) = \mathrm{diag}(0.5 + t, 2 - t, 1 - t, t).$$

The integration interval is $[0, 2]$ with nongeneric points at $t = 0,\ 0.25,\ 0.5,\ 0.75,\ 1.5$ and 2. At the nongeneric points $t = 0$ and $t = 2$ simple singular values get zero, and as we described in Section 3.2 and 4 the numerical methods have no problems with this type of nongeneric points. Since $t_0 = 0$ is a nongeneric starting point, we use $E(t_0) = U(t_0)S(t_0)U(t_0)$ as initial SVD.

Frobenius norm errors: Example 1 Ctol = 1.0e-3, RKtol = 1.0e-6					
Method	fnval	kflops	S	X	E
Polar decomposition	31	80	9.95e-16	4.24e-14	2.44e-15
Euler-like	78	281	7.61e-15	2.66e-14	1.69e-14
Orthogonal–projector	348	949	8.80e-06	1.24e-04	1.87e-05
Classical 4-step RK	480	1238	1.86e-04	1.46e-03	3.99e-04

Table 5.1

Table 5.1 shows the results of the algebraic method using a polar decomposition, the Euler–like method from [3], the orthogonal–projector based on the classical 4–step Runge–Kutta method. The algebraic method and the Euler-like method compute high accuracy approximations for $S(t)$, $X(t)$ and the recomputed $E(t)$. Note that these errors are almost as small as the errors that would occur by rounding the exact solution to finite precision. The Euler-like method uses a larger stepsize, but since the stepsize control is more expensive it needs more function evaluations and flops than the algebraic method. The methods which solve the ODE's (4.1)–(4.3) compute approximations with lower accuracy and are more expensive than the other two methods. For the combination of Runge-Kutta and cut-off tolerance we used in this example, the projective method is more efficient. It uses fewer function evaluations and flops than the classical Runge–Kutta method. Furthermore, the solution of the projective integrator has a higher accuracy.

The methods based on higher order ODE methods for (4.1)–(4.3) are sensitive to the choice of the tolerance parameter. To demonstrate this we have included the following two tables.

Table 5.2 shows the results of the classical 4–step Runge–Kutta method for different values of the cut–off tolerance, Ctol, and the Runge–Kutta tolerance, RKtol. Here and in the following X^TX denotes the product $X(t)^TX(t)$.

Table 5.3 shows the results for the orthogonal–projector method based on the 4–step Runge–Kutta method as nonorthogonal projector. In this case the error for $X(t)^TX(t)$ is always about 10^{-15}.

Frobenius norm errors: Example 1 4 step RK method							
Ctol	RKtol	fnval	kflops	S	X	E	X^TX
1e-1	1e-4	252	649	1.12e-02	3.33e-02	3.03e-02	6.30e-05
1e-1	1e-6	1692	4357	3.79e-03	2.11e-02	9.56e-03	1.04e-06
1e-1	1e-8	4092	10536	3.32e-03	2.02e-02	8.08e-03	4.22e-08
1e-3	1e-4	264	681	4.78e-03	3.63e-03	1.14e-02	5.41e-03
1e-3	1e-6	480	1238	1.86e-04	1.46e-03	3.99e-04	1.46e-04
1e-3	1e-8	936	2414	3.52e-07	8.30e-06	8.64e-07	2.76e-07
1e-5	1e-4	828	2135	1.43e+00	2.00e+00	4.19e-02	1.52e-02
1e-5	1e-6	1200	3095	1.41e+00	2.00e+00	7.37e-04	2.83e-04
1e-5	1e-8	936	2414	3.52e-07	8.31e-06	8.64e-07	2.76e-07
1e-5	1e-10	2448	6313	2.37e-09	1.12e-05	7.84e-09	3.08e-09

Table 5.2

Frobenius norm errors: Example 1 Orthogonal–projector based on 4 step RK method							
Ctol	RKtol	Fnval	kflops	S	X	E	X^TX
1e-1	1e-4	252	687	1.12e-03	3.33e-02	3.03e-02	9.81e-16
1e-1	1e-6	1404	3823	2.98e-03	2.00e-02	7.00e-02	1.33e-15
1e-1	1e-8	4200	11436	3.32e-03	2.02e-02	8.08e-03	1.63e-15
1e-3	1e-4	252	687	4.45e-05	8.77e-04	6.89e-05	1.29e-15
1e-3	1e-6	348	949	8.80e-06	1.24e-05	1.87e-05	1.28e-15
1e-3	1e-8	1020	2781	1.32e-07	8.15e-06	2.02e-07	1.68e-15
1e-5	1e-4	264	720	2.49e-05	2.04e-04	3.40e-05	1.34e-15
1e-5	1e-6	348	949	8.80e-06	1.24e-05	1.87e-05	1.28e-15
1e-5	1e-8	996	2716	1.36e-07	2.20e-05	2.07e-07	1.53e-15
1e-5	1e-10	2304	6283	3.63e-09	7.21e-07	5.20e-09	1.63 -15

Table 5.3

These results indicate that the orthogonal–projector method may be more robust than the usual explicit Runge–Kutta method. For Ctol $= 10^{-5}$ and RKtol $= 10^{-4}$ and 10^{-6} the classical 4–step Runge–Kutta method fails, whereas the orthogonal–projective method computes a good approximation within the given tolerances. The projective method allows in most cases larger stepsizes and the number of floating point operations is smaller or of the same order as for the nonprojective method.

5.2. Example 2. This is Example 5, Section 3.5 from [3]:

$$E(t) = X(t)S(t)Y(t)^T$$

where $Y(t) = I_4,\ S(t) = \mathrm{diag}(-t, -t, t^2, t^2), X(t) = exp(tK)$ and $K \in \mathrm{R}^{4\times 4}$ is the skew symmetric matrix

$$K = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 2 & 0 \\ 0 & -2 & 0 & 3 \\ 0 & 0 & -3 & 0 \end{bmatrix}.$$

ETNA
Kent State University
etna@mcs.kent.edu

The integration interval is $[-2, 2]$ with nongeneric points at $t = -1,\ 0$ and 1.

Frobenius norm errors: Example 2 Ctol = 1.0e-5, RKtol = 1.0e-6				
Method	Fnval	kflops	S	E
Polar decomposition	93	336	2.00e-14	6.29e-15
Euler-like	1119	4399	3.30e-13	6.30e-13
Orthogonal–projector	18804	88949	3.12e-07	4.12e-07
Classical 4-step RK	26472	123939	4.00e+00	1.55e-03

Table 5.4

Table 5.4 shows the results for the algebraic method using a polar decomposition, the Euler–like method from [3], the orthogonal–projector based on the classical 4–step Runge–Kutta method.

The matrix in this example has multiple singular values and the methods track different ASVD's. Again, the algebraic and the Euler–like method calculate high accuracy approximations for $S(t)$ and the recomputed $E(t)$, but this time the Euler-like method uses more function evaluations, due to a smaller stepsize, and it requires 13 times the work of the algebraic method. For the methods which solves (4.1) – (4.3) the situation is worse. Even with the given small tolerances the classical 4–step Runge–Kutta method fails to compute an accurate approximation. The orthogonal–projector calculates an approximation of the order 10^{-7}, but uses many more function evaluations and flops than the algebraic or Euler–like method.

6. Conclusion. In this paper we presented two new methods for computing an ASVD. The algebraic method uses the structure of an ASVD and adjusts in each step a constant SVD using polar decompositions. The projective orthogonal integrators solve the ODE's proposed by Wright. They are based on an explicit Runge–Kutta method and project the Runge–Kutta approximation onto the set of the orthogonal matrices. We compared these methods with the Euler–like method from Bunse–Gerstner et al. and explicit ODE methods.

Our numerical experiments indicate that the algebraic method uses fewer floating point operations than the other methods, especially those methods which solve ODE's. The algebraic method allows us to choose bigger stepsizes. Both, the algebraic and Euler–like method calculate high accuracy approximations at generic points, but at nongeneric points they calculate only $O(h)$ approximations. Therefore our variable stepsize codes avoid nongeneric points.

The explicit Runge–Kutta and orthogonal projective methods can in principle also compute approximations with higher accuracy at nongeneric points, but we have to perturb the differential equations near nongeneric points. The consequence is that these methods sometimes fail to compute accurate approximations. Our results show that the orthogonal–projector methods may be more robust and allow bigger stepsizes, which compensates for the additional work. In [15] recently implicit methods were used which seem to give better accuracy even at nongeneric points.

REFERENCES

[1] E. Anderson, Z. Bai, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK User' Guide*, SIAM, Philadelphia, PA, USA, 1992.

[2] S. Bell and N. Nichols, *Numerical computation of the anlytic singular value decomposition*, to appear in Proceedings of the MTNS-93, Regensburg.

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

[4] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Stat. Comp. 11 (1990), pp. 873–912.

[5] L. Dieci, R. D. Russel, and E. S. Van Vleck, *Unitary integrators and applications to continuous orthonormalisation techniques*, SIAM J. Numer, Anal., to appear.

[6] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *LINPACK User's Guide*, SIAM, Philadelphia, PA, 1979.

[7] D. K. Faddejew and W. N. Faddejewa, *Numerische Methoden der linearen Algebra*, Fifth ed., R. Oldenbourg Verlag, München, 1979.

[8] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Second ed., The John Hopkins University, Baltimore, Maryland, 1989.

[9] T. Kato, *Pertubation Theory for Linear Operators*, Second ed. Springer-Verlag, Berlin, 1976.

[10] *386-Matlab*, The MathWorks, Inc., Cochituate Place, 24 Prime Park Way, Natick, MA 01760, USA., 1990.

[11] W. Rath, *Numerische Methoden zur Berechnung von glatten Singulärwertzerlegungen*, Diplomarbeit, RWTH Aachen, Institut für Geometrie und Praktische Mathematik, März 1993.

[12] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler, *Matrix Eigensystem Routines – EISPACK Guide*, Lecture Notes in Computer Science. Springer Verlag, New York, N.Y., 1979.

[13] J. Stoer and R. Bulirsch, *Numerische Mathematik*, volume 2. Springer-Verlag, Berlin, FRG, 1973.

[14] K. Wright, *Differential equations for the analytic singular value decomposition of a matrix*, Numer. Math., 63 (1992), pp. 283–295.

[15] K. Wright, *Numerical solution of differential equations for the analytic singular value decomposition*, Technical Report Series 405, University of Newcastle upon Tyne, Computing Laboratory, Oct., 1992.