Konrad-Zuse-Zentrum für Informationstechnik Berlin

Hongyuan Zha*

A Numerical Algorithm for Computing the Restricted Singular Value Decomposition of Matrix Triplets

* Konrad-Zuse-Zentrum
für Informationstechnik Berlin
Heilbronner Straße 10
1000 Berlin 31

Institute of Mathematics,
Fudan University,
Shanghai, P.R. of China

Preprint SC-89-1 (März 1989)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Strasse 10
1000 Berlin 31
Verantwortlich: Dr. Klaus André
Umschlagsatz und Druck: Verwaltungsdruckerei Berlin

ISSN 0933-7911

Hongyuan Zha

A Numerical Algorithm for Computing the Restricted Singular Value Decomposition of Matrix Triplets[1]

Dedicated to Prof. Richard S. Varga on the occasion of his 60th Birthday.

Abstract

This paper presents a numerical algorithm for computing the restricted singular value decomposition of matrix triplets (RSVD). It is shown that one can use unitary transformations to separate the regular part from a general matrix triplet. After preprocessing on the regular part, one obtains a matrix triplet consisting of three upper triangular matrices of the same dimensions. The RSVD of this special matrix triplet is computed using the implicit Kogbetliantz technique. The algorithm is well suited for parallel computation.

Keywords: Restricted singular values, matrix triplets, unitary transformations, implicit Kogbetliantz technique.

Subject Classification: AMS(MOS): 65F15, 65F30, 65H15.

[1]Contributed paper accepted for Conference on Approximation Theory and Numerical Linear Algebra (in honor of Richard S. Varga), March 30-31, April 1, 1989

Inhalt

1. Introduction

In [9] we introduce the concept of restricted singular values of matrix triplets (RSV) as follows.

Definition 1.1. Let $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{m\times p}$ and $C \in \mathbb{C}^{q\times n}$, the restricted singular values (RSV) of the matrix triplet (A, B, C) are defined as follows

$$\sigma_k(A,B,C) = \min_{D\in\mathbb{C}^{p\times q}} \{\|D\|_2 |\ \text{rank}\ (A+BDC) \leq k-1\} \tag{1.1}$$

$k = 1,\ldots,n$ where $\|\cdot\|_2$ denotes the spectral norm of a matrix. A main theorem concerning the RSV is proved in [9], which is termed as RSVD theorem.

Theorem 1.1. [9], (RSVD).
1. Let $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{m\times p}$ and $C \in \mathbb{C}^{q\times n}$, then there exist nonsingular matrices P, Q and unitary matrices U and V such that

$$PAQ = \begin{array}{c} \\ s_2 \\ t_2 \\ \\ \end{array} \begin{array}{c} \left(\begin{array}{ccc} \Sigma_A & & \\ & O_A^{(1)} & \\ & & O_A^{(2)} \end{array}\right) \\ \begin{array}{ccc} & s_1 & t_1 \end{array} \end{array} \tag{1.2.a}$$

$$PBU = \begin{array}{c} \\ t_2 \end{array} \begin{pmatrix} \Sigma_B \\ O_B^{(2)} \end{pmatrix} \tag{1.2.b}$$

$$VCQ = \begin{array}{cc} (\Sigma_c, & O_c^{(2)}) \\ & t_1 \end{array} \tag{1.2.c}$$

$$\Sigma_A = \begin{array}{c} \\ \\ \\ s \\ \\ \end{array} \begin{array}{c} \left(\begin{array}{cccc} I_j & & & \\ & I_k & & \\ & & I_l & \\ & & & \Sigma \end{array}\right) \\ \begin{array}{cccc} & & & s \end{array} \end{array} \tag{1.3.a}$$

$$
\Sigma_B = \begin{array}{c c}
 & \begin{pmatrix} I_j & & & \\ & O_B^{(1)} & & \\ & & I_s & \\ & & & I_{s_2} \end{pmatrix} \\
k+l & \\
 & p-j-r-s_2
\end{array}
\tag{1.3.b}
$$

$$
\Sigma_C = \begin{array}{c c}
q-l-r-s_1 & \begin{pmatrix} O_C^{(1)} & & & \\ & I_l & & \\ & & I_s & \\ & & & I_{s_1} \end{pmatrix} \\
 & j+k
\end{array}
\tag{1.3.c}
$$

where $\Sigma = \text{diag}\ (\sigma_i)$ and $\sigma_1 \geq \ldots \geq \sigma_s > 0$.

2. Let $r = i + j + k$, denote

$$
\begin{array}{llll}
\alpha_i = 1, & \beta_i = 1, & \gamma_i = 0 & i = 1, \ldots, j \\
\alpha_i = 1, & \beta_i = 0, & \gamma_i = 0 & i = j+1, \ldots, j+k \\
\alpha_i = 0, & \beta_i = 1, & \gamma_i = 0 & i = j+k+1, \ldots, r \\
\alpha_i = \sigma_{i-j-k}, & \beta_i = 1, & \gamma_i = 1 & i = r+1, \ldots, r+s \\
\alpha_i = 0, & \beta_i = 1, & \gamma_i = 1 & i = r+s+1, \ldots, r+s+\min(s_1, s_2)
\end{array}
\begin{array}{r}
(1.4.a) \\ (1.4.b) \\ (1.4.c) \\ (1.4.d) \\ (1.4.e)
\end{array}
$$

then

$$
\begin{array}{ll}
\text{(i)} \quad \sigma_i(A, B, C) = \dfrac{\alpha_i}{\beta_i \gamma_i} & i = 1, \ldots, r+s+\min(s_1, s_2) \\
\text{(ii)} \quad \sigma_i(A, B, C) = 0 & i = r+s+\min(s_1, s_2)+1, \ldots, n.
\end{array}
$$

For the applications of RSVD, the reader is referred to [7] and [10]. The purpose of this paper is to provide a numerical algorithm for computing the RSVD of general matrix triplets. It is organized as follows. In the second section we introduce the concept of regular RSV of a general matrix triplet, then we show how to use unitary transformations to separate a regular sub-triplet which only contains the regular RSV from a general matrix triplet. In the third section we provide an efficient algorithm for preprocessing the regular sub-triplet resulted from the previous section. In the last section we discuss implicit Kogbetliantz technique applied to the product of three matrices. In this paper only complex matrices are discussed, however, the case of real matrices can be considered similarly. We use $\mathbb{C}^{m \times n}$ to denote the set of all m by n complex matrices; I_s denotes the identity matrix of order s; O with different sub- and super-scripts (e.g. $O_A^{(1)}$) denotes zero matrices with different dimensions.

2. Separating the Regular Sub-Triplet from a General Matrix Triplet

Definition 2.1. The triplets

$$\alpha_i = \sigma_{i-j-k}, \quad \beta_i = 1, \quad \gamma_i = 1 \quad i = r+1, \ldots, r+s \tag{2.1.a}$$

$$\alpha_i = 0, \quad \beta_i = 1, \quad \gamma_i = 1 \quad i = r+s+1, \ldots, r+s+\min(s_1, s_2) \tag{2.1.b}$$

in (1.4) are called the regular RSV of (A, B, C). In other words, they are the nontrivial finite RSV of (A, B, C). From the above definition and Theorem 1.1, it is easy to obtain the following

Proposition 2.2. If a matrix triplet (A, B, C) has the property that B and C are nonsingular, then (A, B, C) only has regular RSV, and such a matrix triplet is called a *regular* matrix triplet.

The purpose of this section is to show that one can use unitary transformations to separate the regular sub-triplet from a general matrix triplet. The whole process consists of five steps, the transformation from one step to the next step is of the following form:

Step k to step $k+1$

$$\begin{pmatrix} A^{(k)} & B^{(k)} \\ C^{(k)} & O \end{pmatrix} \longrightarrow \begin{pmatrix} A^{(k+1)} & B^{(k+1)} \\ C^{(k+1)} & O \end{pmatrix} := \begin{pmatrix} U_1^{(k)} A^{(k)} U_2^{(k)} & U_1^{(k)} B^{(k)} V_1^{(k)} \\ V_2^{(k)} C^{(k)} U_2^{(k)} & O \end{pmatrix} \tag{2.2}$$

where $U_i^{(k)}$ and $V_i^{(k)} (i = 1, 2)$ are unitary matrices. $A^{(k)}$, $B^{(k)}$ and $C^{(k)}$ are the transformed A, B and C at step k. All the submatrices are conformally partitioned. Let

$$A^{(0)} = A\,, \quad B^{(0)} = B\,, \quad C^{(0)} = C \tag{2.3}$$

Step 1. Transform $C^{(0)}$ to $(O|C_2^{(1)})$ such that $C_2^{(1)}$ has full column rank

$$\begin{pmatrix} (A_1^{(1)}| & A_2^{(1)}) & B^{(1)} \\ (O| & C_2^{(1)}) & O \end{pmatrix} \tag{2.4.a}$$

Step 2. Transform $A_1^{(1)}$ to $\left(\frac{A_{11}^{(2)}}{O}\right)$ such that $A_{11}^{(2)}$ has full row rank

$$\begin{pmatrix} \left(\begin{array}{c|c} A_{11}^{(2)} & A_{12}^{(2)} \\ \hline O & A_{22}^{(2)} \end{array}\right) & \left(\begin{array}{c} B_1^{(2)} \\ \hline B_2^{(2)} \end{array}\right) \\ (\,O \mid C_2^{(2)}\,) & O \end{pmatrix} \tag{2.4.b}$$

Step 3. Transform $B_2^{(2)}$ to $\left(\frac{O}{B_3^{(3)}}\right)$ such that $B_3^{(3)}$ has full row rank

$$\begin{pmatrix} \left(\begin{array}{c|c} A_{11}^{(3)} & A_{12}^{(3)} \\ \hline O & A_{22}^{(3)} \\ \hline O & A_{32}^{(3)} \end{array}\right) & \left(\begin{array}{c} B_1^{(3)} \\ \hline O \\ \hline B_3^{(3)} \end{array}\right) \\ (O|C_2^{(3)}) & O \end{pmatrix} \tag{2.4.c}$$

Step 4. Transform $A_2^{(3)}$ to $\left(A_{22}^{(4)}|O\right)$ such that $A_{22}^{(4)}$ has full column rank

$$\begin{pmatrix} \left(\begin{array}{c|c|c} A_{11}^{(4)} & A_{12}^{(4)} & A_{13}^{(4)} \\ \hline O & A_{22}^{(4)} & O \\ \hline O & A_{32}^{(4)} & A_{33}^{(4)} \end{array}\right) & \left(\begin{array}{c} B_1^{(4)} \\ \hline O \\ \hline B_3^{(4)} \end{array}\right) \\ (O|C_2^{(4)}|C_3^{(4)}) & O \end{pmatrix} \tag{2.4.d}$$

Step 5. Transform $B_3^{(4)}$ to $\left(B_{31}^{(5)}|O\right)$ and $C_3^{(4)}$ to $\left(\frac{C_{13}^{(5)}}{O}\right)$ such that $B_{31}^{(5)}$ and $C_{13}^{(5)}$ are nonsingular matrices (see Proposition 2.4.).

$$\begin{pmatrix} \left(\begin{array}{c|c|c} A_{11}^{(5)} & A_{12}^{(5)} & A_{13}^{(5)} \\ \hline O & A_{22}^{(5)} & O \\ \hline O & A_{32}^{(5)} & A_{33}^{(5)} \end{array}\right) & \left(\begin{array}{c|c} B_{11}^{(5)} & B_{12}^{(5)} \\ \hline O & O \\ \hline B_{31}^{(5)} & O \end{array}\right) \\ \left(\begin{array}{c|c|c} O & C_{12}^{(5)} & C_{13}^{(5)} \\ \hline & C_{22}^{(5)} & O \end{array}\right) & O \end{pmatrix} \tag{2.4.e}$$

Remark 2.3. Both the QR decomposition (with column pivoting) and SVD can be used in the above transformations, for details of QRD and SVD see [3] and [8].

Proposition 2.4. After the above five steps, the resulted matrix triplet $(A_{33}^{(5)}, B_{31}^{(5)}, C_{13}^{(5)})$ only consists the regular RSV of (A, B, C) and thus is a regular matrix triplet.

Proof. Since $B_3^{(4)}$ has full row rank and $C_3^{(4)}$ has full column rank, $B_{31}^{(5)}$ and $C_{13}^{(5)}$ are nonsingular. Let us now consider the structure of $A^{(5)}+B^{(5)}D^{(5)}C^{(5)}$, while $D^{(5)}$ are partitioned conformally with $B^{(5)}$ and $C^{(5)}$ as in (2.4.e), then

$$A^{(5)} + B^{(5)}D^{(5)}C^{(5)} = \left(\begin{array}{c|c|c} A_{11}^{(5)} & X_{12} & X_{13} \\ \hline O & A_{22}^{(5)} & O \\ \hline O & X_{32} & A_{33}^{(5)} + B_{31}^{(5)}D_{11}^{(5)}C_{13}^{(5)} \end{array}\right)$$

where X_{12}, X_{13} and X_{13} are possible nonzero submatrices of $A^{(5)}+B^{(5)}D^{(5)}C^{(5)}$. Since in Step 1 to step 5, all the transformations used are unitary (see Remark 2.3) and hence nonsingular, $A_{11}^{(5)}$ and $A_{22}^{(5)}$ have full row rank and full column rank respectively, then

$$\begin{aligned} \text{rank}\,(A+BDC) &= \text{rank}\,(A^{(5)} + B^{(5)}D^{(5)}C^{(5)}) \\ &= \text{rank}\left(\begin{array}{c|c|c} A_{11}^{(5)} & O & O \\ \hline O & A_{22}^{(5)} & O \\ \hline O & O & A_{33}^{(5)} + B_{31}^{(5)}D_{11}^{(5)}C_{13}^{(5)} \end{array}\right) \\ &= \text{rank}\,(A_{11}^{(5)}) + \text{rank}\,(A_{22}^{(5)}) + \text{rank}\,(A_{33}^{(5)} + B_{31}^{(5)}D_{11}^{(5)}C_{13}^{(5)}) \end{aligned}$$

Combining the above with Definition 1.1 and Proposition 2.2, the proposition is proved. ∎

3. Preprocessing of a Regular Matrix Triplet

Let (A,B,C) be a regular matrix triplet $A \in \mathbb{C}^{m\times n}$ arbitrary, $B \in \mathbb{C}^{m\times m}$ and $C \in \mathbb{C}^{n\times n}$ are nonsingular matrices. We first give the following result which can be obtained from definition 1.1.

Proposition 3.1. Let the singular values of $B^{-1}AC^{-1}$ be

$$\sigma_1 \geq \ldots \geq \sigma_n \geq 0$$

then

$$\sigma_i(A, B, C) = \sigma_i \quad i = l, \ldots, n \,. \tag{3.1}$$

For the preprocessing procedure, we distinguish two cases

1. $m \geq n$

   - Transform C to upper triangular form using QR decomposition $Q_C C = R_C$ where R_C is upper triangular and Q_C is unitary.

   - Transform A to upper triangular form

   $$Q_A A = \tilde{R}_A = \begin{array}{cc} \begin{pmatrix} R_A \\ O \end{pmatrix} & \begin{array}{c} n \\ m-n \end{array} \end{array}$$

   where R_A is $n \times n$ upper triangular and Q_A is unitary.

   - Let $\tilde{B} = Q_A B$, transform $\tilde{B}$ to upper triangular form

   $$\tilde{B} Q_B = \begin{array}{cc} \begin{pmatrix} R_B & R_{12} \\ O & R_{22} \end{pmatrix} & \begin{array}{c} n \\ m-n \end{array} \\ \begin{array}{cc} n & m-n \end{array} & \end{array}$$

   where R_B is $n \times n$ upper triangular matrix and Q_B is unitary. It is easy to see that

   $$Q_B^T(B^{-1}AC^{-1})Q_C^T = \begin{array}{cc} \begin{pmatrix} R_B^{-1} R_A R_C^{-1} \\ O \end{pmatrix} & \begin{array}{c} n \\ m-n \end{array} \end{array} \tag{3.2}$$

2. $m < n$

- Transform B to upper triangular form using QR decomposition

$$BW_B = R_B$$

where R_B is upper triangular and Q_B is unitary.

- Transform A to upper triangular form

$$\begin{array}{ccc} AQ_A & = & (O, \quad R_A) \\ & & \;\; n-m \quad m \end{array}$$

where R_A is upper triangular and Q_A is unitary.

- Let $\tilde{C} = CQ_A$, transform $\tilde{C}$ to upper triangular form

$$Q_C\tilde{C} = \begin{array}{cc} \begin{pmatrix} R_{11} & R_{12} \\ O & R_C \end{pmatrix} & \begin{array}{c} n-m \\ m \end{array} \\ \quad n-m \;\; m & \end{array}$$

where R_C is upper triangular and Q_C is unitary.

It is easy to see that

$$Q_B^T(B^{-1}AC^{-1})Q_C^T = (O, R_B^{-1}T_AR_C^{-1}) \,. \tag{3.3}$$

Summarizing (3.2) and (3.3), we can transform the regular triplet using unitary matrices to a special matrix triplet consisting of three upper triangular matrices of order $\min(m,n)$. For the detailed description of the QR decomposition and its variants see [3].

It is interesting to observe that if in step 5 of section 2 (2.4.e), QR decompositions are used to transform $B_3^{(4)}$ to $B_{31}^{(5)}$ and $C_3^{(4)}$ to $C_{13}^{(5)}$ respectively. $B_{31}^{(5)}$ and $C_{13}^{(5)}$ are already in upper triangular form. Computational work can be reduced if the following more efficient algorithm for preprocessing is applied. The idea is to combine step two and three in a single step, so that the aim now is to find unitary matrices Q_A and Q_B such that Q_AA and Q_ABQ_B are in upper triangular form.

We illustrate the procedure of the algorithm using a low dimension example, where we assume $m = 3$ and $n = 2$. As used conventionally, $\times$ represents possible nonzero entries of a matrix; $\otimes$ represents the entry to be zeroed out at the current step. The transformations used are complex Givens transformations [3].

$$A = \begin{pmatrix} \times & \times & \times \\ & \times & \times \\ & & \times \end{pmatrix} B = \begin{pmatrix} \times & \times \\ \times & \times \\ \times & \times \end{pmatrix}$$

$$\text{i)} \qquad \begin{array}{c} \\ \rightarrow \\ \rightarrow \end{array} \begin{pmatrix} \times & \times & \times \\ & \times & \times \\ & & \times \end{pmatrix} \qquad \begin{pmatrix} \times & \times \\ \times & \times \\ \otimes & \times \end{pmatrix}$$

(This notation, in words, means complex Givens transformation is applied to row 2 and 3 of A and B to zero out the (3.1) entry of B. We will not repeat this explanation in the following steps).

$$\text{ii)} \qquad \begin{array}{c} \begin{pmatrix} \times & \times & \times \\ & \times & \times \\ & \otimes & \times \end{pmatrix} \\ \begin{array}{ccc} & \uparrow & \uparrow \end{array} \end{array} \qquad \begin{pmatrix} \times & \times \\ \times & \times \\ \circ & \times \end{pmatrix}$$

$$\text{iii)} \qquad \begin{array}{c} \rightarrow \\ \rightarrow \\ \\ \end{array} \begin{pmatrix} \times & \times & \times \\ & \times & \times \\ & & \times \end{pmatrix} \qquad \begin{pmatrix} \times & \times \\ \otimes & \times \\ & \times \end{pmatrix}$$

$$\text{iv)} \qquad \begin{array}{c} \begin{pmatrix} \times & \times & \times \\ \otimes & \times & \times \\ & & \times \end{pmatrix} \\ \begin{array}{ccc} \uparrow & \uparrow & \end{array} \end{array} \qquad \begin{pmatrix} \times & \times \\ \circ & \times \\ \circ & \times \end{pmatrix}$$

$$\text{v)} \qquad \begin{array}{c} \\ \rightarrow \\ \rightarrow \end{array} \begin{pmatrix} \times & \times & \times \\ & \times & \times \\ & & \times \end{pmatrix} \qquad \begin{pmatrix} \times & \times \\ \circ & \times \\ \circ & \otimes \end{pmatrix}$$

$$\text{vi)} \qquad \begin{array}{c} \begin{pmatrix} \times & \times & \times \\ & \times & \times \\ & \otimes & \times \end{pmatrix} \\ \begin{array}{ccc} & \uparrow & \uparrow \end{array} \end{array} \qquad \begin{pmatrix} \times & \times \\ \circ & \times \\ \circ & \circ \end{pmatrix}$$

$$\text{vii)} \qquad \begin{pmatrix} \times & \times & \times \\ & \times & \times \\ & & \times \end{pmatrix} \qquad \begin{pmatrix} \times & \times \\ \circ & \times \\ \circ & \circ \end{pmatrix}$$

To make the paper concise, here we omit the detailed analysis of the operation counts of the above procedure.

4. Implicit Kogbetliantz Technique Applied to the Product of Three Matrices

The idea of implicitly using Kogbetliantz algorithm is not new, it is firstly used in [6] for computing the generalized singular value decomposition (GSVD) in [4] for computing the singular decomposition of the product of two matrices. Recently algorithms for computing the singular value decomposition of the product of three matrices based on GSVD of [2] is proposed. In this section we derive a different algorithm which does not base on GSVD. A previous version of this algorithm which is designed for implicitly computing the SVD of $CA^{-1}B$ appeared in [5]. We assume the reader is familiar with Kogbetliantz algorithm and only present the core algorithm here i.e. algorithm for implicitly computing the 2×2 subproblem. For other issues concerning the implementation of Kogbetliantz algorithm, for example, ordering schemes, convergence analysis and systolic (parallel) implementation, the reader is referred to the above mentioned papers and the references therein.

From (3.2) and (3.3), we can assume now that A, B and C are upper triangular matrices of the same order n, furthermore let

$$E = B^{-1}AC^{-1}$$

and $E = (e_{ij})$, $B = (b_{ij})$, $A = (a_{ij})$ and $C = (c_{ij})$. It is easy to verify that

$$\begin{pmatrix} e_{ii} & e_{i\,i+1} \\ O & e_{i+1\,i+1} \end{pmatrix} = \begin{pmatrix} b_{ii} & b_{i\,i+1} \\ O & b_{i+1\,i+1} \end{pmatrix}^{-1} \begin{pmatrix} a_{ii} & a_{i\,i+1} \\ O & a_{i+1\,i+1} \end{pmatrix} \begin{pmatrix} c_{ii} & c_{i\,i+1} \\ O & c_{i+1\,i+1} \end{pmatrix} \tag{4.1}$$

for $i = 1, \ldots, n-1$. In the following we also consider the case that

$$\begin{pmatrix} b_{ii} & b_{i\,i+1} \\ O & b_{i+1\,i+1} \end{pmatrix} \text{ and } \begin{pmatrix} c_{ii} & c_{i\,i+1} \\ O & c_{i+1\,i+1} \end{pmatrix}$$

are possibly singular or nearly singular, so that instead of (4.1) we consider

$$\begin{pmatrix} f_{ii} & f_{i\,i+1} \\ O & f_{i+1\,i+1} \end{pmatrix} = \operatorname{adj} \begin{pmatrix} b_{ii} & b_{i\,i+1} \\ O & b_{i+1\,i+1} \end{pmatrix} \begin{pmatrix} a_{ii} & a_{i\,i+1} \\ O & a_{i+1\,i+1} \end{pmatrix} \operatorname{adj} \begin{pmatrix} c_{ii} & c_{i\,i+1} \\ O & c_{i+1\,i+1} \end{pmatrix} \tag{4.2}$$

where $\operatorname{adj}(T)$ means the adjoint of T. The idea of using adj is suggested in [6].

Proposition 4.1. Let J_1 and J_2 be the combination of rotations and permutations such that

$$J_1^H \begin{pmatrix} f_{ii} & f_{i\,i+1} \\ O & f_{i+1\,i+1} \end{pmatrix} J_2^H = \begin{pmatrix} \hat{f}_{ii} & O \\ O & \hat{f}_{i+1\,i+1} \end{pmatrix} \tag{4.3}$$

then we can choose 2×2 unitary matrices J_3 and J_4 such that

$$J_3 \begin{pmatrix} b_{ii} & b_{i\,i+1} \\ O & b_{i+1\,i+1} \end{pmatrix} J_1$$

$$J_3 \begin{pmatrix} a_{ii} & a_{i\,i+1} \\ O & a_{i+1\,i+1} \end{pmatrix} J_4$$

$$J_2 \begin{pmatrix} c_{ii} & c_{i\,i+1} \\ O & c_{i+1\,i+1} \end{pmatrix} J_4$$

are 2×2 upper triangular matrices.

Proof. Let

$$\begin{aligned} \tilde{B} &= \begin{pmatrix} b_{ii} & b_{i\,i+1} \\ O & b_{i+1\,i+1} \end{pmatrix} J_1 &= \begin{pmatrix} \tilde{b}_{ii} & \tilde{b}_{i\,i+1} \\ \tilde{b}_{i\,i+1} & \tilde{b}_{i+1\,i+1} \end{pmatrix} \\ \tilde{C} &= J_2 \begin{pmatrix} c_{ii} & c_{i\,i+1} \\ O & c_{i+1\,i+1} \end{pmatrix} &= \begin{pmatrix} \tilde{c}_{ii} & \tilde{c}_{i\,i+1} \\ \tilde{c}_{i\,i+1} & \tilde{c}_{i+1\,i+1} \end{pmatrix} \end{aligned}$$

In the following we distinguish three cases

1. If $|\tilde{b}_{ii}|^2 + |\tilde{b}_{i+1\,i}|^2 \neq 0$ and $|\tilde{c}_{i+1\,i}|^2 + |\tilde{b}_{i+1\,i+1}|^2 \neq 0$ chose J_3 and J_4 such that

$$\begin{aligned} J_3\tilde{B} &= \begin{pmatrix} \hat{b}_{ii} & \hat{b}_{i\,i+1} \\ O & \hat{b}_{i+1\,i+1} \end{pmatrix} \\ \tilde{C}J_4 &= \begin{pmatrix} \hat{c}_{ii} & \hat{c}_{i\,i+1} \\ O & \hat{c}_{i+1\,i+1} \end{pmatrix} \end{aligned}$$

It is easy to see that

$$\hat{b}_{ii} \neq 0 \text{ and } \hat{c}_{i+1\,i+1} \neq 0 \,.$$

Let

$$J_3 \begin{pmatrix} a_{ii} & a_{i\,i+1} \\ O & a_{i+1\,i+1} \end{pmatrix} J_4 = \begin{pmatrix} \hat{a}_{ii} & \hat{a}_{i\,i+1} \\ \hat{a}_{i+1\,i} & \hat{a}_{i+1\,i+1} \end{pmatrix}$$

One can derive from

$$\begin{pmatrix} \hat{f}_{ii} & O \\ O & \hat{f}_{i+1\,i+1} \end{pmatrix} = \operatorname{adj} \begin{pmatrix} \hat{b}_{ii} & \hat{b}_{i\,i+1} \\ O & \hat{b}_{i+1\,i+1} \end{pmatrix} \begin{pmatrix} \hat{a}_{ii} & \hat{a}_{i\,i+1} \\ \hat{a}_{i+1\,i} & \hat{a}_{i+1\,i+1} \end{pmatrix} \operatorname{adj} \begin{pmatrix} \hat{c}_{ii} & \hat{c}_{i\,i+1} \\ O & \hat{c}_{i+1\,i+1} \end{pmatrix}$$

that

$$\hat{a}_{i\,i+1} = O\,.$$

2. If $|\hat{b}_{ii}|^2 + |\hat{b}_{i\,i+1}|^2 = 0$ choose J_4 unitary such that

$$\hat{C}J_4 = \begin{pmatrix} \hat{c}_{ii} & \hat{c}_{i\,i+1} \\ O & \hat{c}_{i+1\,i+1} \end{pmatrix}$$

choose J_3 unitary such that

$$J_3\left[\begin{pmatrix} \hat{a}_{ii} & \hat{a}_{i\,i+1} \\ O & \hat{a}_{i+1\,i+1} \end{pmatrix} J_4\right] = \begin{pmatrix} \hat{a}_{ii} & \hat{a}_{i\,i+1} \\ O & \hat{a}_{i+1\,i+1} \end{pmatrix}$$

then

$$J_3\hat{B} = \begin{pmatrix} O & \hat{b}_{i\,i+1} \\ O & \hat{b}_{i+1\,i+1} \end{pmatrix}$$

3. If $|\hat{c}_{i+1\,i}|^2 + |\hat{c}_{i+1\,i+1}|^2 = 0$ choose J_3 unitary such that

$$J_3\hat{B} = \begin{pmatrix} O & \hat{b}_{i\,i+1} \\ O & \hat{b}_{i+1\,i+1} \end{pmatrix}$$

choose J_4 unitary such that

$$\left[J_3\begin{pmatrix} a_{ii} & a_{i\,i+1} \\ O & a_{i+1\,i+1} \end{pmatrix}\right] J_4 = \begin{pmatrix} \hat{a}_{ii} & \hat{a}_{i\,i+1} \\ O & \hat{a}_{i+1\,i+1} \end{pmatrix}$$

then

$$\hat{C}J_4 = \begin{pmatrix} \hat{c}_{ii} & \hat{c}_{i\,i+1} \\ O & O \end{pmatrix}$$

$\blacksquare$

Remark 4.2 The above algorithm can be easily modified when (4.3) is computed using the approximate variant proposed in [1].

Conclusion

This paper presents a numerical algorithm for computing the RSVD of a general matrix triplets. The algorithm is based on separating the regular sub-triplet from a general matrix triplet and application of implicit Kogbetliantz technique. Through out the algorithm only unitary transformations are used which may guarantee the numerical reliability of the algorithm. Detailed implementation will be developed in connection with [7] and numerical experiments will be reported in a separate paper.

Acknowledgement. The author wishes sincerely to thank Prof. Dr. P. Deuflhard for his support and encouragement and Mrs. S. Wacker for her careful and excellent typing of the manuscript.

References

[1] J.P. Charlier, M. Vanbegin, P. Van Dooren: *On efficient implementations of Kogbetliantz's algorithm for computing singular value decomposition.* Numer. Math. Vol. 52, p. 279-300 (1988).

[2] L.M. Ewerbring, F.T. Luk: *Canonical correlations and generalized SVD, applications and new algorithms.* To appear in J. Comput. Appl. Math. (1989).

[3] G. Golub, C. Van Loan: *Matrix Computations.* The Johns Hopkins Univ. Press, USA (1983).

[4] M.T. Heath, A.J. Laub, C.C. Paige, R.C. Ward: *Computing the singular value decomposition of a product of two matrices.* SIAM J. Sci. Stat. Comput. Vol. 7, No. 4, p. 1147-1159 (1986).

[5] Erxiong Jiang, Hongyuan Zha: *Eigenstructure Computation of Matrices* (in Chinese) Scientific Publishing House, Beijing (in preparation).

[6] C.C. Paige: *Computing the generalized singular value decomposition.* SIAM J. Sci. Stat. Comput. Vol. 7, No. 4, p. 1126-1146 (1986).

[7] S. Van Huffel, Hongyuan Zha: *Structured total least squares (STLS): a unified approach for solving structured generalized LS and total LS problems.* Contributed paper for SIAM annual meeting 1989, San Diego, CA, USA.

[8] P. Van Dooren: *The computation of Kronecker's canonical form of a singular pencil.* Linear Algebra Appl., Vol. 27, p. 103-140 (1979).

[9] Hongyuan Zha: *Restricted singular value decomposition of matrix triplets and rank determination of matrices.* Submitted to SIAM J. Matrix Anal. and Appl. (1988).

[10] Hongyuang Zha, P.C. Hansen: *Regularization and the general Gauss-Markov Linear Model (*in preparation).

Veröffentlichungen des Konrad-Zuse-Zentrums für Informationstechnik Berlin
Preprints März 1989

SC 86-1. P. Deuflhard; U. Nowak. *Efficient Numerical Simulation and Identification of Large Chemical Reaction Systems.* (vergriffen)

SC 86-2. H. Melenk; W. Neun. *Portable Standard LISP for CRAY X-MP Computers.*

SC 87-1. J. Anderson; W. Galway; R. Kessler; H. Melenk; W. Neun. *The Implementation and Optimization of Portable Standard LISP for the CRAY.*

SC 87-2. Randolph E. Bank; Todd F. Dupont; Harry Yserentant. *The Hierarchical Basis Multigrid Method.* (vergriffen)

SC 87-3. Peter Deuflhard. *Uniqueness Theorems for Stiff ODE Initial Value Problems.*

SC 87-4. Rainer Buhtz. *CGM-Concepts and their Realization.*

SC 87-5. P. Deuflhard. *A Note on Extrapolation Methods for Second Order ODE Systems.*

SC 87-6. Harry Yserentant. *Preconditioning Indefinite Discretization Matrices.*

SC 88-1. Winfried Neun; Herbert Melenk. *Implementation of the LISP-Arbitrary Precision Arithmetic for a Vector Processor.*

SC 88-2. H. Melenk; H.M. Möller; W. Neun. *On Gröbner Bases Computation on a Supercomputer Using REDUCE.* (vergriffen)

SC 88-3. J. C. Alexander; B. Fiedler. *Global Decoupling of Coupled Symmetric Oscillators.*

SC 88-4. Herbert Melenk; Winfried Neun. *Parallel Polynomial Operations in the Buchberger Algorithm.*

SC 88-5. P. Deuflhard; P. Leinen; H. Yserentant. *Concepts of an Adaptive Hierarchical Finite Element Code.*

SC 88-6. P. Deuflhard; M. Wulkow. *Computational Treatment of Polyreaction Kinetics by Orthogonal Polynomials of a Discrete Variable.* (vergriffen)

SC 88-7. H. Melenk; H. M. Möller; W. Neun. *Symbolic Solution of Large Stationary Chemical Kinetics Problems.*

SC 88-8. Ronald H. W. Hoppe; Ralf Kornhuber. *Multi-Grid Solution of Two Coupled Stefan Equations Arising in Induction Heating of Large Steel Slabs.*

SC 88-9. Ralf Kornhuber; Rainer Roitzsch. *Adaptive Finite-Element-Methoden für konvektions-dominierte Randwertprobleme bei partiellen Differentialgleichungen.*

SC 88-10. C. -N. Chow; B. Deng; B. Fiedler. *Homoclinic Bifurcation at Resonant Eigenvalues.*

SC 89-1. Hongyuan Zha. *A Numerical Algorithm for Computing the Restricted Singular Value Decomposition of Matrix Triplets.*