Updating Low-Rank Approximations

Masterarbeit

Master of Science Mathematik

Mathematisches Institut
Universität Tübingen

vorgelegt von
Benjamin Kiefer
3870596
ben93_kiefer@yahoo.de

betreut durch
Prof. Dr. Christian Lubich

06/2019

Acknowledgment

I would first like to thank my thesis advisor Prof. Dr. Lubich for giving me the opportunity to write this thesis in the Numerical Analysis Group and the continuous support he provided. In particular, I'm grateful for the office he provided me with during the writing period of this work.
Furthermore, I would like to thank the Numerical Analysis Group for the welcoming atmosphere and the diverse and helpful discussions and conversations throughout the whole time.
Finally, I want to express my gratitude to Dominik Edelmann and Jörg Nick for proofreading this thesis.

Contents

1 Introduction

The low-rank matrix approximation problem is approximating a matrix by one whose rank is less than that of the original matrix. Low-rank approximation (LRA) is an essential tool in various applications, e.g. in image compression [1], data mining [2],[3], Latent Semantic Indexing [4], Principal Component Analysis [5],[6], Statistics and Machine Learning [7], [8], [9] and many financial transaction or network activity streams [10].

With the ever-growing data sizes and the need for fast processing such data bases, fast and memory-efficient algorithms for updating LRAs are needed. Updating a LRA means, given an LRA $Z_0 \in \mathbb{R}^{m\times n}$ of $A_0 \in \mathbb{R}^{m\times n}$ and an update ΔA, compute a LRA Z_1 of $A_1 \coloneqq A_0 + \Delta A$ or $A_1 \coloneqq [A_0, \Delta A]$ based on the old LRA Z_0. The aim of updating a LRA is to be more efficient and yet not much less inaccurate than just recomputing a LRA from scratch.

Updating LRAs is closely connected to updating the Singular Value Decomposition (SVD). This is due to the well-known Eckart–Young–Mirsky theorem which states that the minimization problem of finding the best rank-r approximation to a given matrix A_0,

$$\min_{\mathrm{rank}(B)=r} \|A_0 - B\|,$$

is given by the SVD that is truncated after the r-th largest singular value. Hence, updating SVDs gives rise to updating LRAs as will be seen later.

The field of updating SVD has extensively been studied in the past, see e.g. [11], [12], [13]. But also other methods for updating an LRA exist; such as one based on the similar ULV decomposition [14] or an entirely different approach, called the Dynamical LRA (DLRA) [15] which is based on systems of ordinary differential equations.

This work intends to provide an overview over some algorithms that exist in the field of updating LRA while comparing them theoretically and numerically with respect to complexity and their approximation properties. More specifically, we review a general SVD-based updating method and its advantageous usage in LRA and compare the resulting algorithms with that of the DLRA. This work is structured as follows.

In section 2 we formulate the nomenclature and the conventions used in this work while in section 3 we formulate the LRA updating problem mathematically.

In section 4 we review the process of updating LRAs via updating the SVD. We review certain kinds of updates and justify the usage of updating an LRA based on an old LRA. Furthermore, we extend an existing SVD-based decomposition update routine from column appending updates to the case of additive modification updates and discuss the truncation strategies and its advantegous usage in LRAs in detail.

In section 5 we review the Dynamical Low-Rank Approximation. We review the theoretical foundation of the ordinary differential equations and the resulting approximation with its error estimates and a numerically efficient integrator.

Finally, we conclude the analysis with numerical experiments that compare the SVD-based LRAs and the DLRA on model problems and on an application example from Latent Semantic Indexing.

2 Prerequisites

2.1 Notation

Unless otherwise stated, in the whole work $\|\cdot\|$ denotes the Frobenius norm: for $A \in \mathbb{R}^{m\times n}$, we write

$$\|A\| = \left(\sum_{i,j} a_{i,j}^2\right)^{1/2}.$$

Correspondingly, $\langle \cdot, \cdot \rangle$ denotes the inner product: for $A, B \in \mathbb{R}^{m\times n}$, we write

$$\langle A, B\rangle = \operatorname{tr}\left(A^T B\right) = \sum_{i,j} a_{i,j} b_{i,j}.$$

By convention, $\|\cdot\|_2$ denotes the spectral norm. I_n denotes the quadratic identity matrix of size $n \times n$. We write $(A)_j$ (important: with brackets) if we mean the j-th column of a matrix A. Sometimes, we use `Matlab`-notation if we consider submatrices: $A_{r:t,d:f}$ denotes the submatrix obtained by taking the rows r to t and the columns d to f (inclusively). If we mean all columns or all rows of A, we just write ":". For example, the submatrix obtained by taking the first r columns is denoted as $A_{:,1:r}$. In other cases, a subscript means the index of a sequence or family of matrices/vectors. However, this either becomes clear from context or will be made clear specifically.

We say $U \in \mathbb{R}^{m\times n}$, $m \geq n$, is orthonormal if the columns of U are mutually orthogonal and normalized. Consequentially, iff $m = n$, then the columns of U represent an orthonormal basis of $\mathbb{R}^n$ and the definition coincides with that of an conventional orthogonal matrix.

Many of the algorithms rely on the singular value decomposition for which we also introduce some notation.

2.2 Singular Value Decomposition

Theorem 2.2.1 (Singular Value Decomposition)**.** *Any matrix $A \in \mathbb{R}^{m\times n}$ can be decomposed as a product of three matrices,*

$$A = USV^T,$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal, i.e.,

$$U^T U = I_m, \ \ V^T V = I_n,$$

where I_m is the identity matrix of dimension m, and $S \in \mathbb{R}^{m\times n}$ is diagonal with its diagonal entries (singular values) being non-negative and in non-increasing order; that is, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, where $p = \min(m, n)$. Furthermore, $\sigma_k = 0$ for $k > \mathrm{rank}(A)$. Such a decomposition is called (full) singular value decomposition.

Proof. See for example [16]. □

As the $m - n$ last rows (for $m > n$) or the $n - m$ last columns (for $n > m$) of S are zero, we can omit them by truncating the corresponding U- or V-matrix, respectively, to yield the so-called reduced or thin SVD;

$$A = USV^T \in \mathbb{R}^{m\times n}$$

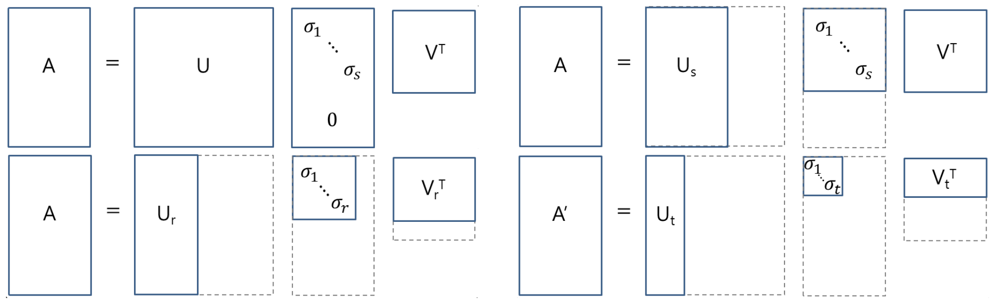

Figure 1: Different forms of SVDs for $A \in \mathbb{R}^{m\times s}, m > s$, $\text{rank}(A) = r > t$. From top left to bottom right: Full SVD, Thin SVD, Compact SVD, Truncated SVD or best rank-t approximation.

where $U \in \mathbb{R}^{m\times n}$, $S \in \mathbb{R}^{n\times n}$ diagonal and $V \in \mathbb{R}^{n\times n}$ for the case $m > n$ and $U \in \mathbb{R}^{m\times m}$, $S \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times m}$ for the case $n > m$.

We can even extend the thin SVD by considering the rank of the matrix. From last theorem, we know that for a rank-r matrix $A \in \mathbb{R}^{m\times n}$, $r < m, n$, the singular values $\sigma_{r+1}, \dots, \sigma_m$ (for $m \geq n$) are zero and we can further truncate them and the columns of U and V corresponding to these zero singular values and still keeping an exact decomposition. We call

$$A = USV^T \in \mathbb{R}^{m\times n}$$

with $U \in \mathbb{R}^{m\times r}$, $V \in \mathbb{R}^{n\times r}$ and $S \in \mathbb{R}^{r\times r}$ the compact SVD.

Finally, we introduce the notation that whenever we're given a decomposition $A = USV^T$, U_r, V_r denote the matrices U and V, respectively, where all the columns after the r-th are truncated and S_r denotes the r-th leading principal submatrix of S such that $Z = U_r S_r V_r^T$ is a rank-r matrix approximation of A.

3 Problem Formulation

Low-rank approximation (LRA) is a minimization problem which seeks for a given matrix $A \in \mathbb{R}^{m\times n}$ another matrix X with $r = \text{rank}(X) < \text{rank}(A)$ such that

$$\|A - X\| = \min!$$

The solution, called the *best rank-r-approximation*, is obtained by computing the singular value decomposition of A and truncating all the singular values after the r largest ones:

Theorem 3.0.1. *Let the SVD of A be given as $A = USV^T$ and define*

$$X := \sum_{i=1}^{r} u_i \cdot \sigma_i \cdot v_i^T = U_r S_r V_r^T,$$

then

$$\min_{rank(B)=r} \|A - B\| = \|A - X\|$$

Proof. See the proof of Eckart and Young [17]. □

From now on, unless otherwise stated, X denotes the best rank-r approximation to a given matrix A. What we are interested in here is updating an LRA. Suppose we already have a rank-r approximation (r-LRA) (either best rank-r approximation X_0 or another r-LRA Z_0) of a matrix A_0 and now we want to update A_0 by (1) adding columns or (2) rows or simply by (3) changing values of A_0 ($\hat{=}$ additive modification) and obtain a corresponding updated low-rank approximation X_1 or Z_1 of the new matrix A_1. In many applications, the updated LRA should retain a rank $\leq r$ mainly because computational limits are set. Note that we will focus on (1) and (3) only, since (2) directly follows from (1) by transposing A_0: Instead of adding a row to A_0, we are adding a column to $(A_0)^T$. In the case of additive modifications, we're going to denote the update either by a matrix ΔA or by a matrix product CD^T. The latter approach is motivated from the field of Latent Semantic Indexing [13]. There, in each time step we modify our term-document-matrix, A_0, by changing c many terms where $D^T \in \mathbb{R}^{c\times n}$ specifies the difference between old weights and new ones (in general just matrix entries) and $C \in \mathbb{R}^{m\times c}$ is a selection matrix indicating the c terms that need adjusting. We can use both notions of updates interchangeably because we can always transform one expression to the other. This is easily seen by computing the compact SVD of the update matrix as $\Delta A = USV^T$ where $U \in \mathbb{R}^{m\times r}$, $V \in \mathbb{R}^{n\times r}$ and $S \in \mathbb{R}^{r\times r}$ where r denotes the rank of ΔA. Setting $C = U$ and $D = SV$ yields a corresponding (non-unique) rank factorization. The first few algorithms strictly rely on the representation of the update in the factorized form. Thus, we have to ask ourselves how we obtain our data. If we cannot fall back on information such as which terms change in which documents but rather just are given the update in its matrix form, we must decompose the update matrix first in order to be able to apply the corresponding algorithms. Depending on the density of the update matrix this may itself be a computational complex question. In this work, we assume that the update is given in matrix and its rank factorization form.

The naive way to approach such an update problem would be to simply compute the SVD of the new A_1 and truncating it, ignoring any of the information of the previously computed old low-rank approximation X_0 or Z_0 of A_0 as described in Algorithm 1. This could be done by means of a full SVD algorithm or partial SVD via a bidiagonalization procedure [18], [19].

Algorithm 1 Naive algorithm

```
1: procedure COMPUTE BEST RANK-r-APPROXIMATION OF A_1
2:     input: A_1
3:     U, S, V ← svd(A_1) or partial SVD using e.g. Matlab's svds
4:     output: U_r, S_r, V_r
```

Computing a full SVD would be very expensive and is not feasible for large matrices that might even demand for real-time changes. For the bidiagonalized method, computing the SVD of a matrix $A \in \mathbb{R}^{m\times n}$ costs $4mn^2 - \frac{4}{3}n^3$ for $m > n$ for the bidiagonalization part. It is said that, in practise, the iterative procedure to obtain a diagonal matrix, is insignificant compared to the bidiagonalization part [20]. Essentially, computing a full SVD then costs $\mathcal{O}(mn \min(m, n))$ operations. If a complete SVD is not feasible, an approximate solution can be obtained by means of the Lanczos bidiagonalization process. However, in this work, we are concerned with non-iterative methods.

4 Updating LRAs Using SVD

4.1 General Additive Identity

In this section, we are going to establish a general identity for additive modifications of an existing LRA. These derivations exist in various different forms, e.g. in [12], [13].
Suppose we are given a r-LRA $Z_0 = U_0 S_0 V_0^T$ of our data matrix $A_0 \in \mathbb{R}^{m\times n}$ where $S_0 \in \mathbb{R}^{r\times r}$, $U_0 \in \mathbb{R}^{m\times r}$ and $V_0 \in \mathbb{R}^{n\times r}$. We're interested in a LRA of

$$Z_0 + CD^T = [U_0, C] \begin{bmatrix} S_0 & 0 \\ 0 & I_c \end{bmatrix} [V_0, D]^T, \tag{1}$$

where $C \in \mathbb{R}^{m\times c}$ and $D \in \mathbb{R}^{n\times c}$ are given matrices of rank $c \leq n$. Let P be an orthonormal basis (i.e. $P \in \mathbb{R}^{m\times r_1}$ denotes a matrix consisting of columns that are orthonormal basis vectors) of the column space of $(I_m - U_0U_0^T)C \in \mathbb{R}^{m\times c}$ with rank $0 \leq r_1 \leq c$ and likewise let $Q \in \mathbb{R}^{n\times r_2}$ be an orthonormal basis of the column space of $(I_n - V_0V_0^T)D \in \mathbb{R}^{n\times c}$ with rank $0 \leq r_2 \leq c$. Subsequently, set $R_C \coloneqq P^T(I_m - U_0U_0^T)C \in \mathbb{R}^{r_1\times c}$ and $R_D \coloneqq Q^T(I_n - V_0V_0^T)D \in \mathbb{R}^{r_2\times c}$. If r_1 or r_2 are zero, then $P = 0$ and $R_C = 0$ or $Q = 0$ and $R_D = 0$, respectively, and in that case, we suppress the dimensions occupied by these matrices. Note that this looks very similar to a QR decomposition but is not since R_C and R_D aren't necessarily upper triangular or square. In our first algorithms however, we use the QR decomposition to stably obtain a basis. Furthermore, note that PP^T and QQ^T are the orthogonal projections onto the column space of P and Q, respectively. Since they themselves are bases of $(I_m - U_0U_0^T)C$ and $(I_n - V_0V_0^T)D$, respectively, it follows that

$$PR_C = PP^T(I_m - U_0U_0^T)C = (I_m - U_0U_0^T)C \quad \text{and} \quad QR_D = QQ^T(I_n - V_0V_0^T)D = (I_n - V_0V_0^T)D.$$

With these identities we can rewrite the outer matrices from (1) to

$$[U_0, C] = [U_0, P] \begin{bmatrix} I_r & U_0^T C \\ 0 & R_C \end{bmatrix} \quad \text{and} \quad [V_0, D] = [V_0, Q] \begin{bmatrix} I_r & V_0^T D \\ 0 & R_D \end{bmatrix}. \tag{2}$$

Note that both $[U_0, P] \in \mathbb{R}^{m\times r+r_1}$ and $[V_0, Q] \in \mathbb{R}^{n\times r+r_2}$ are orthonormal since we chose P and Q to be an orthonormal basis of the column space of $(I_m - U_0U_0^T)C$ and $(I_n - V_0V_0^T)D$, respectively, which by themselves are orthogonal to U_0 and V_0, respectively. Plugging (2) into (1) yields

$$\begin{aligned} Z_0 + CD^T &= [U_0, P] \begin{bmatrix} I_r & U_0^T C \\ 0 & R_C \end{bmatrix} \begin{bmatrix} S_0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I_r & V_0^T D \\ 0 & R_D \end{bmatrix}^T [V_0, Q]^T \\ &= [U_0, P] \begin{bmatrix} S & U_0^T C \\ 0 & R_C \end{bmatrix} \begin{bmatrix} I_r & V_0^T D \\ 0 & R_D \end{bmatrix}^T [V_0, Q]^T \\ &= [U_0, P] \begin{bmatrix} S_0 + U_0^T C D^T V_0 & U_0^T C R_D^T \\ R_C D^T V_0 & R_X R_Y^T \end{bmatrix} [V_0, Q]^T \\ &= [U_0, P] \underbrace{\left(\begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} U_0^T C \\ R_C \end{bmatrix} \begin{bmatrix} V_0^T D \\ R_D \end{bmatrix}^T \right)}_{=:K} [V_0, Q]^T \end{aligned} \tag{3}$$

We compute the full SVD of $K \in \mathbb{R}^{r+r_1 \times r+r_2}$ as $K = U'S'V'^T$ where $U' \in \mathbb{R}^{r+r_1 \times r+r_1}$, $S' \in \mathbb{R}^{r+r_1 \times r+r_2}$ and $V' \in \mathbb{R}^{r+r_2 \times r+r_2}$. Finally, equations (3) together with this SVD yield the exact compact SVD of our altered matrix

$$Z_0 + CD^T = ([U_0, P]U')S'([V_0, Q]V')^T. \tag{4}$$

In general, it is of higher rank than r. To control the rank we could simply truncate the SVD to obtain the best rank-r-approximation of $Z_0 + CD^T$ by truncating the smaller matrices U', S' and V':

$$Z_1 \coloneqq U_1 S_1 V_1^T \coloneqq ([U_0, P]U_r')S_r'([V_0, Q]V_r')^T). \tag{5}$$

In fact, this is the algorithm in [13] for additive modifications which we present in Algorithm 2. If we didn't care about the increasing rank, then we would simply omit the truncating step yielding increased computational cost for every rank-increasing update.

Algorithm 2 Additive modifications to LRA

1: **procedure** COMPUTE r-LRA OF $Z_0 + CD^T$
2: Given: r-LRA $Z_0 = U_0 S_0 V_0^T$, update matrices C and D
3: $PR_X \leftarrow \mathtt{qr}((I - U_0 U_0^T)X)$ (thin variant), $QR_Y \leftarrow \mathtt{qr}((I - V_0 V_0^T)D)$ (thin variant)
4: $K \leftarrow \begin{bmatrix} S_0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} U_0^T C \\ R_C \end{bmatrix} \begin{bmatrix} V_0^T D \\ R_D \end{bmatrix}^T$, $U'S'V'^T \leftarrow \mathtt{svd}(K)$
5: $U_1 \leftarrow [U_0, P]U_r'$, $V_1 \leftarrow [V_0, Q]V_r'$, $S_1 \leftarrow S_r'$
6: output: U_1, S_1, V_1

4.1.1 Computational Complexity

Algorithm 2 is only cheaper than recomputing the SVD if C and D are small. With the best ordering, computing the matrices in line 3 costs $\mathcal{O}(mrc)$ and $\mathcal{O}(nrc)$, respectively, and its QR-factorizations cost $\mathcal{O}(mc^2)$ and $\mathcal{O}(nc^2)$. Line 4 requires computing the matrix product $U_r^T X$ and $V_r^T Y$ in mrc and nrc operations and finally computing the matrix product of $(r + c \times c)$-matrices, i.e., $\mathcal{O}((r + c)^2 c)$ operations. The SVD of a $(r + c \times r + c)-$matrix costs $\mathcal{O}((r+c)^3)$ operations. Finally, rotating the singular vector matrices in line 5 costs $m(r+c)r$ and $n(r+c)r$, respectively. Clearly, if $r + c \geq m, n$ it is not advisable to use this algorithm to update the SVD but rather compute a new one or possibly use any of the algorithms that follow. We give a more precise operation count over all of the Algorithms in Table 1 right before the numerical testing section.

4.2 Appending Columns

The additive modification identities give rise to appending and deleting columns as described in this and the next section.

Suppose again we're given a r-LRA $Z_0 = U_0 S_0 V_0^T$ of A_0. Furthermore, suppose we want to add c columns, denoted as $C \in \mathbb{R}^{m \times c}$ to A_0. We're looking for a LRA of $[Z_0, C]$. Establishing an updated LRA for this matrix can be interpreted as a special case of the additive modification problem in the section before. Here, we just add c many zero columns to V_0^T and choose C from before as the C here and $D = [0_{c \times n}, I_c]^T \in \mathbb{R}^{n+c \times c}$, where $0_{c \times n}$ is the $c \times n$-matrix consisting of all zeros. Because then it holds

$$[Z_0, C] = U_0 S_0 [V_0^T, 0_{m \times c}] + CD^T.$$

We restrict our analysis to columns that are to be appended at the end of A_0 (or Z_0) since one can always apply a permutation matrix P from the right to change the column order of the resulting matrix. That change in the decomposition only affects the order of the rows in V.
With the factors U_0, S_0 and $[V_0^T, 0_{m\times c}]$ we can follow the lines from the previous section. More precisely, let $P \in \mathbb{R}^{m\times r_1}$ denote an orthonormal basis of $(I_m - U_0U_0^T)C \in \mathbb{R}^{m\times c}$ where r_1 denotes the rank of the orthogonal projection onto the complement space of C. Let $R_C = P^T(I_m - U_0U_0^T)C \in \mathbb{R}^{r_1\times c}$ and note again that $PR_C = PP^T(I_m - U_0U_0^T)C = (I_m - U_0U_0^T)C$, since PP^T is the orthogonal projection onto $(I_m - U_0U_0^T)C$. These formulas yield

$$[U_0, P]\underbrace{\begin{bmatrix} S_0 & U_0^TC \\ 0 & R_C \end{bmatrix}}_{:=K}\begin{bmatrix} V_0^T & 0 \\ 0 & I_c \end{bmatrix} = [U_0, P]\begin{bmatrix} S_0V_0^T & U_0^TC \\ 0 & R_C \end{bmatrix} = [U_0S_0V_0^T, U_0U_0^TC + PR_C] = [Z_0, C],$$

where again $[U_0, P] \in \mathbb{R}^{m,r+r_1}$ was constructed to be orthonormal. Likewise the last matrix of above decomposition is trivially seen to be orthonormal as well since V_0 is. Computing a SVD of K as $K = U'S'V'^T$, where $U' \in \mathbb{R}^{r+r_1\times r+r_1}, S' \in \mathbb{R}^{r+r_1\times r+c}$, and $V' \in \mathbb{R}^{r+c\times r+c}$ yields the exact compact SVD which after truncation yields the updated r-LRA

$$[Z_0, C] = ([U_0, P]U')S'\left(\begin{bmatrix} V_0 & 0 \\ 0 & I_c \end{bmatrix}V'\right)^T \overset{\text{truncate}}{\to} ([U_0, P]U_r')S_r'\left(\begin{bmatrix} V_0 & 0 \\ 0 & I_c \end{bmatrix}V_r'\right)^T =: U_1S_1V_1^T =: Z_1.$$

Algorithm 3 Appending columns to LRA
1: **procedure** COMPUTE r-LRA OF $[Z_0, C]$
2: Given: r-LRA $Z_0 = U_0S_0V_0^T$, column updates C
3: $PR_C \leftarrow \texttt{qr}((I - U_0U_0^T)C)$ (thin variant)
4: $K \leftarrow \begin{bmatrix} S_0 & U_0^TC \\ 0 & R_C \end{bmatrix}$, $\quad U'S'V'^T \leftarrow \texttt{svd}(K)$
5: $U_1 \leftarrow [U_0, P]U_r'$, $\quad V_1 \leftarrow \begin{bmatrix} V_0 & 0 \\ 0 & I_c \end{bmatrix}V_r'$, $\quad S_1 \leftarrow S_r'$
6: output: U_1, S_1, V_1

4.2.1 Computational Complexity

The computational complexity is given by the matrix products in line 3 ($\mathcal{O}(mrc)$ operations) and the computation of the QR-factorization of a $m\times c$-matrix, i.e., $\mathcal{O}(mc^2)$ steps. Computing line 4 requires the SVD of a $r+c\times r+c$-matrix, that is, $\mathcal{O}((r+c)^3)$ steps. Finally, rotating the truncated inner singular vector matrices is done in $m(r+c)r$ and $n(r + c)r$ steps, respectively.

4.3 Deleting Columns

Similarly to adding columns, we can delete columns in the LRA via the additive modification procedure. Suppose we are given a r-LRA $[Z_0^1, Z_0^2] = U_0S_0V_0^T$ of $[A_0, C]$ and want to delete the last c (similar derivation for arbitrary columns) columns of $[A_0, C]$ for $C \in \mathbb{R}^{m\times c}$ and therefore downdate the corresponding LRA. In order to delete the columns, we first set them to zero which can be done exactly as there is no rank increase in doing so. For doing that, we choose the C from the additive modification section as $-C$ and $D = [0_{c\times n}, I_c]^T$ because then it holds

$$[Z_0^1, 0_{m\times c}] = [Z_0^1, Z_0^2] - [0_{m\times n}, C] = [Z_0^1, Z_0^2] + CD^T = U_0 S_0 V_0^T + CD^T.$$

With the additive modification procedure this representation then yields a decomposition

$$[Z_0^1, 0_{m\times c}] = U_1 S_1 V_1^T,$$

where the last c columns of V_1^T (or last c rows of $V_1 \in \mathbb{R}^{n+c\times r}$) are zero and truncating them yields the desired downdated LRA

$$Z_0^1 = U_1 S_1 (V_1^T)_{:,1:n-c}.$$

In this case, P is unused (since $-C$ is already in the span of the column space of $U_0 U_0^T$ and therefore P would be zero) and therefore the K matrix simplifies to

$$K = [S_0 + U_0^T C D^T V_0, U_0^T C R_D^T] \in \mathbb{R}^{r\times r+c}.$$

The algorithm is summarized in Algorithm 4 which yields the exact solution to the problem.

Algorithm 4 Downdating columns in LRA

1: **procedure** COMPUTE r-LRA OF SUBMATRIX
2: Given: r-LRA $[Z_0^1, Z_0^2] = U_0 S_0 V_0^T$, columns to be downdated C
3: $D \leftarrow \begin{bmatrix} 0_{n\times c} \\ I_c \end{bmatrix}$, $QR_D \leftarrow \texttt{qr}((I - V_0 V_0^T)D)$ (thin variant)
4: $K \leftarrow [S_0 - U_0^T C D^T V_0 - U_0^T C R_D^T]$, $U' S' V'^T \leftarrow \texttt{svd}(K)$
5: $V_1 \leftarrow [V_0, Q]V_r'$, $U_1 \leftarrow U_0 U_r'$, $S_1 \leftarrow S_r'$
6: output: $U_1, S_1', (V_1)_{1:n-c,:}$

Note that this Algorithm requires the representation of C. However, a similar Algorithm can be derived without one having to know C explicitily. This follows immediately from $C = U_0 S_0 V_0^T D \Rightarrow U_0^T C = S_0 V_0^T D$. Replacing $U_0^T C$ by $S_0 V_0^T Y$ yields an equivalent algorithm. Furthermore, note that the rank possibly decreases and we could further truncate the resulting singular vector and value matrices in that case.

4.3.1 Computational Complexity

The matrix product in line 3 is computed in $\mathcal{O}(nrc)$ and its QR decomposition in $\mathcal{O}(nc^2)$ operations. The SVD in line 4 costs $\mathcal{O}((r+c)^3)$ operations and the subspace rotations of line 5 $n(r+c)r$ and mr^2 operations.

4.4 Usage of Old LRA

The established identities for updating the LRA by changing or adding columns are exact. The algorithms, however, truncate the smaller singular vector and value matrices of the SVD in order to obtain a fixed-rank updating procedure over periods of updates. Over subsequent different updates this introduces an error since we don't take into account the whole update at once but have it splitted into several smaller updates as is the case, e.g., in online settings or series of moving images. In [13] it is investigated how well these algorithms fit the original matrix. We are going to discuss the usage of the old best rank-r approximation X_0 as opposed to the full matrix A_0 in the problem of appending columns.

4.4.1 Usage of Old LRA for Column Appending

In the following subsection, we present a class of matrices for which we can use the old low-rank approximation to update by appending columns without loss of exactness, that is, under which conditions the best rank-r approximations of $[X_0, C]$ and $[A_0, C]$ coincide where X_0 is the best rank-r approximation of A_0. First, we present several Lemmas.

Lemma 4.4.1. *Let $A \in \mathbb{R}^{m\times n}$. Let $Q \in \mathbb{R}^{n\times n}$ be orthonormal. Then*

$$\sigma_i(AQ^T) = \sigma_i(A) \text{ for } i = 1, \dots, \min(m, n).$$

Proof. Let $A = USV^T$ be the full SVD of A for $U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}$ orthonormal and $S \in \mathbb{R}^{m\times n}$ with singular values ordered non-increasingly. Postmultiplying this SVD by Q^T yields $USV^TQ^T = US(QV)^T$ where QV is again orthonormal as a product of two orthonormal matrices. The matrix S stays the same, thus $\sigma_i(A) = \sigma_i(AV^T)$ for all $i = 1, \dots, \min(m, n)$. □

Lemma 4.4.2. *Let $A = [A_1, A_2]$ for some matrices $A_1 \in \mathbb{R}^{m\times a}$ and $A_2 \in \mathbb{R}^{m\times b}$. Then*

$$\sigma_i(A_1) \leq \sigma_i(A).$$

Proof. Instead of looking at the singular values of $[A_1, A_2]$ and A_1, respectively, we can look at the eigenvalues of $AA^T = [A_1, A_2][A_1, A_2]^T = A_1A_1^T + A_2A_2^T$ and $A_1A_1^T$ due to the one-to-one correspondence of singular values of A and eigenvalues of AA^T. Looking at Courant-Fischer's minmax principle, we obtain an identity for the i-th largest eigenvalue for a given symmetric matrix $B \in \mathbb{R}^{n\times n}$ denoting the eigenvalues as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$:

$$\lambda_i(B) = \min_{X \in X_i} \max_{0 \neq x \in X} \frac{x^T Bx}{x^T x},$$

where X_i denotes the set of all i-dimensional vector subspaces of $\mathbb{R}^n$. With that, we immediately see that

$$\lambda_i(AA^T) = \lambda_i(A_1A_1^T + A_2A_2^T) = \min_{X \in X_i} \max_{0 \neq x \in X} \frac{x^T(A_1A_1^T + A_2A_2^T)x}{x^T x} \geq \min_{X \in X_i} \max_{0 \neq x \in X} \frac{x^T A_1A_1^T x}{x^T x},$$

since $\frac{x^T A_2A_2^T x}{x^T x} = \frac{\langle Ax, Ax\rangle}{\langle x, x\rangle} \geq 0$. Hence, also the singular values of A are pairwise greater or equal to those of A_1. □

With the help of these Lemmas, we can formulate a theorem that says that our singular values in our truncated update always remain smaller or equal to the ones in the exact update.

Theorem 4.4.3. *Let $A \in \mathbb{R}^{m\times n}$, X its best rank-r approximation and $C \in \mathbb{R}^{m\times c}$. Then*

$$\sigma_i([X, C]) \leq \sigma_i([A, C]) \text{ for } i = 1, \dots, m$$

Proof. Let the SVD of A be

$$A = [U_r, U_r^\perp] \begin{bmatrix} S_r & 0 \\ 0 & S_r^\perp \end{bmatrix} [V_r, V_r^\perp]^T,$$

where $U_r \in \mathbb{R}^{m\times r}, U_r^\perp \in \mathbb{R}^{m\times n-r}, S_r \in \mathbb{R}^{r\times r}, S_r^\perp \in \mathbb{R}^{n-r\times n-r}, V_r \in \mathbb{R}^{n\times r}$ and $V_r^\perp \in \mathbb{R}^{n\times n-r}$. Then it holds for all $i = 1, \dots m$,

$$\sigma_i\left([A, C]\right) = \sigma_i\left(\left[[U_r, U_r^\perp]\begin{bmatrix} S_r & 0 \\ 0 & S_r^\perp \end{bmatrix}[V_r, V_r^\perp]^T, C\right]\right) \tag{6}$$

$$= \sigma_i\left(\left[[U_r, U_r^\perp]\begin{bmatrix} S_r & 0 \\ 0 & S_r^\perp \end{bmatrix}[V_r, V_r^\perp]^T, C\right]\begin{bmatrix} [V_r, V_r^\perp]^T & 0 \\ 0 & I_c \end{bmatrix}\right) = \tag{7}$$

since $Q = \begin{bmatrix} [V_r, V_r^\perp] & 0 \\ 0 & I_c \end{bmatrix}$ is orthonormal and thus we can apply Lemma (4.4.1). Evaluating the matrix-matrix product yields

$$(7) = \sigma_i\left(\left[[U_r, U_r^\perp]\begin{bmatrix} S_r & 0 \\ 0 & S_r^\perp \end{bmatrix}, C\right]\right) = \sigma_i\left([U_r S_r, U_r^\perp S_r^\perp, C]\right). \tag{8}$$

Naturally, changing the order of the columns doesn't change the singular values which can be seen by post-multiplying by P^T where $P^T \in \mathbb{R}^{n\times n}$ is the (orthonormal) permutation matrix $\begin{bmatrix} I_r & 0 & 0 \\ 0 & 0 & I_{n-r} \\ 0 & I_c & 0 \end{bmatrix}$ and again applying Lemma (4.4.1):

$$(8) = \sigma_i\left([U_r S_r, U_r^\perp S_r^\perp, C]\begin{bmatrix} I_r & 0 & 0 \\ 0 & 0 & I_{n-r} \\ 0 & I_c & 0 \end{bmatrix}\right) = \sigma_i\left([U_r S_r, C, U_r^\perp S_r^\perp]\right). \tag{9}$$

Using Lemma (4.4.1) yet one last time and post-multiplying by $\begin{bmatrix} V_r^T & 0 & 0 \\ 0 & I_c & 0 \\ 0 & 0 & I_{n-r} \end{bmatrix}$ yields

$$(9) = \sigma_i\left([U_r S_r, C, U_r^\perp S_r^\perp]\begin{bmatrix} V_r^T & 0 & 0 \\ 0 & I_c & 0 \\ 0 & 0 & I_{n-r} \end{bmatrix}\right) = \sigma_i([U_r S_r V_r^T, C, U_r^\perp S_r^\perp]) = \sigma_i([X, C, U_r^\perp S_r^\perp]).$$

Noting that $[X, C]$ is a submatrix of $[X, C, U_r^\perp S_r^\perp]$ and applying Lemma (4.4.2) we obtain our desired result.

□

Before we will compare the singular values of $[X_0, C]$ with those of $[A_0, C]$, we formulate one last simple Lemma.

Lemma 4.4.4. *Let the SVD of $A \in \mathbb{R}^{m\times n}$ be $A = \sum_{i=1}^m \sigma_i u_i v_i^T$ with the corresponding singular values σ_i and the left and right singular vectors u_i and v_i, respectively. Then for $p \geq r$ it holds*

$$X = best\left(A - \sum_{i=p+1}^{m} \sigma_i u_i v_i^T\right),$$

where X is the best rank-r approximation of A and best$(\cdot)$ also denotes the best rank-r approximation.

Proof. Using the definition of the best rank-r-approximation of A immediately yields

$$\text{best}\left(A - \sum_{i=p+1}^{m} \sigma_i u_i v_i^T\right) = \text{best}\left(\sum_{i=1}^{m} \sigma_i u_i v_i^T - \sum_{i=p+1}^{m} \sigma_i u_i v_i^T\right) = \text{best}\left(\sum_{i=1}^{p} \sigma_i u_i v_i^T\right) = X,$$

since $p \geq r$. □

The following result presents under which conditions there is no approximation degradation due to the truncation. It is stated for the column updating problem. It is taken from [13].

Theorem 4.4.5. *Let $A_1 := [A_0, C]$ and $\tilde{A}_1 := [X_0, C]$ where $A_0 \in \mathbb{R}^{m \times n}$ is the original data matrix and X_0 be its best rank-r-approximation and $C \in \mathbb{R}^{m \times c}$ is the collection of columns to be added. Assume that $m \geq n + c$ and*

$$A_1^T A_1 = F + \sigma^2 I \tag{10}$$

where F is symmetric and positive semi-definite with rank(F)$= r$ and $\sigma \neq 0$. Then it holds

$$\textit{best}(A_1) = \textit{best}(\tilde{A}_1).$$

Proof. From the definition of A_1 we obtain

$$A_1^T A_1 - \sigma^2 I_{n+c} = \begin{bmatrix} A_0^T A_0 - \sigma^2 I_n & A_0^T C \\ C^T A_0 & C^T C - \sigma^2 I_c \end{bmatrix} \in \mathbb{R}^{n+c \times n+c}$$

and furthermore from our assumption (10) and since F has rank at r we know that $\text{rank}(A_0^T A_0 - \sigma^2 I_n) \leq r$ and $\text{rank}(C^T C - \sigma^2 I_c) \leq r$. Since both of the following matrices are symmetric, their eigenvalue decomposition can be chosen with orthonormal eigenvector matrices;

$$A_0^T A_0 - \sigma^2 I_n = V_A \text{diag}(S_A^2, 0) V_A^T \in \mathbb{R}^{n \times n},$$

where $V_A \in \mathbb{R}^{n \times n}$ is orthonormal with corresponding eigenvalues listed in $S_A^2 \in \mathbb{R}^{r_1 \times r_1}$, where $0 \leq r_1 \leq r$, and which are non-zero. Likewise, we obtain the eigendecomposition

$$C^T C - \sigma^2 I_c = V_C \text{diag}(S_C^2, 0) V_C^T \in \mathbb{R}^{c \times c},$$

where $V_C \in \mathbb{R}^{c \times c}$ is orthonormal with corresponding eigenvalues listed in $S_C^2 \in \mathbb{R}^{r_2 \times r_2}$ where $0 \leq r_2 \leq r$. From those decompositions we obtain the SVD of A and C:

$$A_0 = U_A \text{diag}(S_A, \sigma I_{n-k_1}) V_A^T = [U_A^{(1)}, U_A^{(2)}] \text{diag}(S_A, \sigma I_{n-k_1}) [V_A^{(1)}, V_A^{(2)}]^T \tag{11}$$

$$C = U_C \text{diag}(S_C, \sigma I_{c-k_2}) V_C^T = [U_D^{(1)}, U_D^{(2)}] \text{diag}(S_D, \sigma I_{c-k_2}) [V_D^{(1)}, V_D^{(2)}]^T. \tag{12}$$

where $U_A^{(1)} \in \mathbb{R}^{m \times r_1}, U_A^{(2)} \in \mathbb{R}^{m \times n - r_1}, V_A^{(1)} \in \mathbb{R}^{n \times r_1}, V_A^{(2)} \in \mathbb{R}^{n \times n - r_2}, U_C^{(1)} \in \mathbb{R}^{m \times r_2}, U_C^{(2)} \in \mathbb{R}^{m \times c - r_2}, V_C^{(1)} \in \mathbb{R}^{c \times r_1}$ and $V_C^{(2)} \in \mathbb{R}^{c \times c - r_1}$. Now we partition $V_A^T A_0^T C V_C$ into

$$V_A^T A_0^T C V_C = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \tag{13}$$

where $S_{11} \in \mathbb{R}^{r_1 \times r_2}, S_{12} \in \mathbb{R}^{r_1 \times c-r_2}, S_{21} \in \mathbb{R}^{n-r_1 \times r_2}$ and $S_{22} \in \mathbb{R}^{n-r_1 \times c-r_2}$. Since F is symmetric and positive semi-definite with $\text{rank}(F) = r$, we know that $S_{12} = 0, S_{21} = 0, S_{22} = 0$ and $r_1 + r_2 = \text{rank}(F) = r$. Now, we can use the SVDs of A_0 and C to rewrite (13) as

$$\begin{aligned}
\begin{bmatrix} S_{11} & 0 \\ 0 & 0 \end{bmatrix} &= V_A^T (U_A \text{diag}(S_A, \sigma I_{n-r_1}) V_A^T)^T U_C \text{diag}(S_C, \sigma I_{c-r_2}) V_C^T V_C \\
&= \text{diag}(S_A, \sigma I_{n-r_1}) U_A^T U_C \text{diag}(S_C, \sigma I_{c-r_2}) \\
&= \text{diag}(S_A, \sigma I_{n-r_1}) [U_A^{(1)}, U_A^{(2)}]^T [U_C^{(1)}, U_C^{(2)}] \text{diag}(S_C, \sigma I_{p-r_2}) \\
&= [U_A^{(1)} S_A, \sigma U_A^{(2)}]^T [U_C^{(1)} S_C, \sigma U_C^{(2)}],
\end{aligned}$$

which implies

$$(U_A^{(1)})^T U_C^{(1)} = 0, \;\; (U_A^{(2)})^T U_C^{(1)} = 0, \;\; (U_A^{(2)})^T U_C^{(2)} = 0,$$

since S_A and S_C are non-singular. Let $\hat{U} \in \mathbb{R}^{m \times k}$ denote the matrix consisting of orthonormal basis vectors of the space

$$\mathcal{R}([U_A^{(1)}, U_C^{(1)}]) \cap \mathcal{R}([U_A^{(2)}, U_C^{(2)}])^{\perp},$$

where $\mathcal{R}(\cdot)$ denotes the column space of a given matrix and $\mathcal{R}(\cdot)^{\perp}$ denotes the orthogonal complement of that space. By the SVD of A_0 and C from (11) and (12), we obtain

$$[A_0, C] = \left[[U_A^{(1)}, U_A^{(2)}] \begin{bmatrix} S_A & \\ & \sigma I_{n-r_1} \end{bmatrix} [V_A^{(1)}, V_A^{(2)}]^T, [U_C^{(1)}, U_C^{(2)}] \begin{bmatrix} S_D & \\ & \sigma I_{c-r_2} \end{bmatrix} [V_C^{(1)}, V_C^{(2)}]^T \right] \tag{14}$$

$$= \left[[U_A^{(1)}, U_A^{(2)}], [U_C^{(1)}, U_C^{(2)}] \right] \begin{bmatrix} \begin{bmatrix} S_A & \\ & \sigma I_{n-r_1} \end{bmatrix} & \\ & \begin{bmatrix} S_C & \\ & \sigma I_{c-r_2} \end{bmatrix} \end{bmatrix} \begin{bmatrix} (V_A^{(1)})^T & 0 \\ (V_A^{(2)})^T & 0 \\ 0 & (V_C^{(1)})^T \\ 0 & (V_C^{(2)})^T \end{bmatrix} \tag{15}$$

$$= \left[[U_A^{(1)}, U_C^{(1)}], [U_A^{(2)}, U_C^{(2)}] \right] \begin{bmatrix} \begin{bmatrix} S_A & \\ & S_C \end{bmatrix} & \\ & \begin{bmatrix} \sigma I_{n-r_1} & \\ & \sigma I_{c-r_2} \end{bmatrix} \end{bmatrix} \begin{bmatrix} (V_A^{(1)})^T & 0 \\ 0 & (V_C^{(1)})^T \\ (V_A^{(2)})^T & 0 \\ 0 & (V_C^{(2)})^T \end{bmatrix}. \tag{16}$$

Choosing $\tilde{B} := \hat{U}^T [U_A^{(1)}, U_C^{(1)}] \begin{bmatrix} S_A & 0 \\ 0 & S_C \end{bmatrix} \in \mathbb{R}^{r \times r}$ yields

$$\hat{U}\tilde{B} = \hat{U}\hat{U}^T [U_A^{(1)}, U_C^{(1)}] \begin{bmatrix} S_A & 0 \\ 0 & S_C \end{bmatrix} = [U_A^{(1)}, U_C^{(1)}] \begin{bmatrix} S_A & 0 \\ 0 & S_C \end{bmatrix},$$

since $\hat{U}\hat{U}^T$ is the orthogonal projection on the space spanned by the columns of $\hat{U}$, which was constructed in the way that it already is the span of the columns of $[U_A^{(1)}, U_C^{(1)}]$. Using $\tilde{B}$ and this equation yields

$$(16) = \left[\hat{U}, [U_A^{(2)}, U_C^{(2)}] \right] \begin{bmatrix} \tilde{B} & \\ & \begin{bmatrix} \sigma I_{n-r_1} & \\ & \sigma I_{c-r_2} \end{bmatrix} \end{bmatrix} \begin{bmatrix} (V_A^{(1)})^T & 0 \\ 0 & (V_C^{(1)})^T \\ (V_A^{(2)})^T & 0 \\ 0 & (V_C^{(2)})^T \end{bmatrix} \tag{17}$$

with all the singular values of $\tilde{B}$ greater than σ. Therefore, we can split the decomposition of (17) to obtain

$$A_1 = [A_0, C] = \left[\hat{U}, U_C^{(2)}\right] \begin{bmatrix} \tilde{B} & 0 \\ 0 & \sigma I_{c-r_2} \end{bmatrix} \begin{bmatrix} (V_A^{(1)})^T & 0 \\ 0 & (V_C^{(1)})^T \\ 0 & (V_D^{(2)})^T \end{bmatrix} + U_A^{(2)} \sigma I_{n-r_1} [(V_A^{(2)})^T, 0]. \tag{18}$$

The right hand side of (18) is $[X_0, C] + U_A^{(2)} \sigma I_{n-r_1} [(V_A^{(2)})^T, 0]$ since it contains only the first r singular vectors and values of A_0 and is concatenated with C. Therefore, we have

$$A_1 = [X_0, C] + U_A^{(2)} \sigma I_{n-r_1} [(V_A^{(2)})^T, 0] = [X_0, C] + \sigma U_A^{(2)} [(V_A^{(2)})^T, 0] = [X_0, C] + \sum_{i=p+1}^{m} \sigma_i u_i v_i^T,$$

where for all $i = p+1, \ldots, m$ $\{\sigma_i = \sigma,\ u_i,\ v_i^T\}$ is the i-th singular triplet of $[A_0, C]$ for some $p \geq r$. Applying Lemma (4.4.4) to A_1 then yields

$$\mathrm{best}(A_1) = \mathrm{best}\left(A_1 - \sum_{i=p+1}^{m} \sigma_i u_i v_i^T\right) = \mathrm{best}([X_0, C]) = \mathrm{best}(\tilde{A}_1),$$

which concludes the claim. □

4.5 Rank-1 Modifications

In this section, we modify our r-LRA $Z_0 = U_0 S_0 V_0^T$ of A_0 by using rank-1 modifications. Instead of rank-c-matrices as used in the chapter before, we use rank-1 matrices xy^T where $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$. Basically, (2) then can be obtained by following the lines using the new matrices (or vectors) x and y. It turns out that sequentially using rank-one updates is more efficient.

4.5.1 Additive Modifications with Rank-1 Matrix

We would like to obtain an updated LRA of $Z_0 + xy^T = U_0 S_0 V_0^T + xy^T$. The derivation works analogous to the derivation for arbitrary additive modifications since this is just a special case. Unlike there, here we use the modified Gram-Schmidt Algorithm: First, define

$$m \coloneqq U_0^T x \in \mathbb{R}^r \ \text{ and } \ p \coloneqq x - U_0 m. \tag{19}$$

With $R_x = \|p\|$, our new orthonormal basis vector is $P = R_x^{-1} p$ (if $R_x \neq 0$, zero vector otherwise) and likewise define

$$n \coloneqq V_0^T y \in \mathbb{R}^r \ \text{ and } \ q \coloneqq y - V_0 n. \tag{20}$$

With $R_y = \|q\|$, our new orthonormal basis vector is $Q = R_y^{-1} q$ (if $R_y \neq 0$, zero vector otherwise). Note, that m and n also denote the dimensions of A_0, but from context they are distinguished.
Then equation (2) turns into

$$[U_0, x] = [U_0, P] \begin{bmatrix} I_r & m \\ 0 & \|p\| \end{bmatrix} \ \text{ and } \ [V_0, y] = [V_0, Q] \begin{bmatrix} I_r & n \\ 0 & \|q\| \end{bmatrix}.$$

Finally, we obtain

$$Z_0 + xy^T = [U_0, x] \begin{bmatrix} S_0 & 0 \\ 0 & 1 \end{bmatrix} [V_0, y]^T = [U_0, P] \begin{bmatrix} I_r & m \\ 0 & \|p\| \end{bmatrix} \begin{bmatrix} S_0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_r & n \\ 0 & \|q\| \end{bmatrix}^T [V_0, Q]^T \tag{21}$$

$$= [U_0, P] \underbrace{\left(\begin{bmatrix} S_0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} m \\ \|p\| \end{bmatrix} \begin{bmatrix} n \\ \|q\| \end{bmatrix}^T \right)}_{=:K} [V_0, Q]^T \tag{22}$$

by similar computations to (3). Finally as before, computing a full SVD of K yields the updated LRA which can be truncated for constant rank. If we want to modify the i-th column $(Z_0)_i$ to $(\tilde{Z}_0)_i \in \mathbb{R}^m$ we choose $x = (\tilde{Z}_0)_i - (Z_0)_i$ and $y = e_i \in \mathbb{R}^n$. In that case, $n = V_0^T e_i = (V_0^T)_i$ is the i-th row of V_0. For that, the norm of $Q = (y - V_0 n)$ is given by

$$R_y^2 = \|y - V_0 n\|^2 = \langle y, y \rangle - 2\langle b, V_0 n \rangle + \langle V_0 n, V_0 n \rangle = 1 - 2n^T n + n^T n = 1 - n^T n,$$

so that $Q = (y - V_0 n)/(\sqrt{1 - n^T n})$. P and m are given as above.

4.5.2 Appending a Column Using Rank-1 Additive Modifications

Given the r-LRA $Z_0 = U_0 S_0 V_0^T$ of A_0, we want to append a column $z \in \mathbb{R}^m$ to A_0 and obtain an updated LRA. For that, we append a column of zeros to V_0^T to obtain

$$[Z_0, z] = [U_0 S_0 V_0^T, z] = U_0 S_0 [V_0^T, 0] + [0_{m \times n}, z] = U_0 S_0 [V_0^T, 0] + xy^T \in \mathbb{R}^{m \times n+1}, \tag{23}$$

where $x = z$ and $y^T = [0, \cdots, 0, 1] =: e_{n+1}^T$. With the choice of such a vector y, we obtain

$$n = [V_0^T, 0] y = [V_0^T, 0] e_{n+1} = 0 \text{ and } \|q\| = \|y\| = 1.$$

Hence our middle matrix K in (22) simplifies to

$$K = \begin{bmatrix} S_0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} m \\ \|p\| \end{bmatrix} [0, \cdots, 0, 1] = \begin{bmatrix} S_0 & m \\ 0 & \|p\| \end{bmatrix}, \tag{24}$$

which is to be SVD-decomposed yielding an updated LRA as before. This matrix is called a broken-arrowhead matrix. [21] presents an algorithm that computes the SVD of (24) in approximately $2 \log_2^2(\varepsilon)(r+1)^2$ steps where ε denotes the machine precision. This constant factor is not negligible as, for example, for a machine with 64-bit IEEE floats (machine precision $\approx 10^{-16}$) it is roughly 5650. Dependent on the size of r we have to decide whether this approach is tangible. However, in our numerical tests, we use the traditional SVD.

4.5.3 Deleting a Column

Given the r-LRA $Z_0 = U_0 S_0 V_0^T$ of A_0, we want to obtain the LRA of A_0 with the i-th column deleted. For that, again by means of additive modifications, we first set it to zero:

$$[(Z_0)_1, \ldots, (Z_0)_{i-1}, 0, (Z_0)_{i+1}, \ldots, (Z_0)_n] = Z_0 + [0, \ldots, 0, -(Z_0)_i, 0, \ldots, 0] = Z_0 + xy^T,$$

where $x = -(Z_0)_i$ and $y = e_i \in \mathbb{R}^n$. With that choice, we obtain once again $n = V_0^T y = (V_0^T)_i$ and $q = y - V_0 n$ and like before $Q = (y - V_0 n)/\sqrt{1 - n^T n}$ if $q \neq 0$ and the zero vector otherwise. Here, again P is unused and $m = U_0^T (Z_0)_i = S_0 n$ since

$$(Z_0)_i = U_0 S_0 V_0^T y = U_0 S_0 (V_0^T)_i = U_0 S_0 n \ \ \Rightarrow \ \ U_0^T (Z_0)_i = S_0 n.$$

For this downdating procedure, note that in order to delete the i-th column of Z_0, we only need to know the decomposition and the i-th row of V_0. Our K matrix then simplifies to

$$[S_0, 0] + S_0 n [n^T, R_y] = [S_0 + S_0 n n^T, S_0 n R_y],$$

which is to be decomposed via SVD to yield the final updated LRA. Rotating the singular vector matrices yields the new singular vectors and in this procedure, similar to the general downdating procedure, we're going to delete the i-th row of V to obtain our final decomposition.

4.6 Extended Decomposition Update via Rank-1 Modifications

Up until now, using rank-one update matrices for updating hasn't improved the time complexity. We merely split up the possible greater update step in smaller rank-1 updates. However, with the help of the rank-one updates, we can consider a more computationally efficient extended decomposition. In the following we introduce an improved scheme for the case of rank-1 updates. From equation (5) in the general update scheme or its rank-1 analogue, we see that in order to obtain our orthonormal singular vector matrices we have to rotate U_0 by U' and V_0 by V' in each update which costs $\mathcal{O}((m+n)r^2)$ in the case of rank-1 updates. Instead, it is sufficient to rotate the smaller inner singular vector matrices occurring at every update step. In each update $n \in \mathbb{N}_0$ we leave our LRA decomposed into the following five matrices

$$U_n U_n' S_n V_n'^{T} V_n^T, \tag{25}$$

where for $n = 0$ we have our original LRA ($U_n = U_0, U_n' = I_r, S_n = S_0, V_n'^{T} = I_r, V_n^T = V_0^T$) and $U_n U_n'$ and $V_n V_n'$ are orthonormal while we don't require U_n, V_n, U_n' and V_n' to be orthonormal but U_n' and V_n' are invertible. The inner singular vector and value matrices U_n', S_n, V_n' will remain at most of size $r \times r$ where r is our desired approximation rank. The outer singular vector matrices only contain the span of the left and right subspaces. The transformations that make S_n diagonal in each update are contained in the much smaller U_n' and V_n' matrices. This way, we avoide the subspace rotations in each update deferring computational complexity and decreasing numerical error that would accumulate if we rotated the big subspace matrices in each update. Hereby, it is important that we use rank-one updates since rank-increasing updates cause the matrix dimensions to go beyond r. Therefore, we have to introduce a truncating scheme that goes into effect after each rank-1 update such that the complexity stays constant. In the following, we detail the update of the left and right subspaces. This approach was taken by Brand [12] for the column appending problem. We extend it to the column modification problem in the following and later describe its usage for LRAs. However, we keep the order of the updating schemes and present the more general additive modification scheme first.

4.6.1 Extended Decomposition Update for Additive Modifications

Consider a given extended r-LRA $Z_0 = U_0 U_0' S_0 V_0'^{T} V_0^T$ of A_0 with $U_0 U_0'$ and $V_0 V_0'$ orthonormal and U_0' and V_0' regular and we want to perturb/update the i-the column of A_0 and therefore also of its LRA Z_0 via a rank-one update as $Z_0 + xy^T$ where x is the update vector and $y = e_i$ indicates which column is to be updated. We only show the analysis for column updates rather than general rank-1 updates. However, the analysis can be done similarly for the more general case. Following the rank-1 updating procedure as described in (21) yields

$$\begin{aligned} Z_0 + xy^T &= [U_0 U_0', P] K [V_0 V_0', Q]^T = [U_0 U_0', P] L S_1 R^T [V_0 V_0', Q]^T \\ &= [U_0, P] \begin{bmatrix} U_0' & 0 \\ 0 & 1 \end{bmatrix} L S_1 R^T \begin{bmatrix} V_0'^T & 0 \\ 0 & 1 \end{bmatrix} [V_0, Q]^T, \end{aligned} \tag{26}$$

where K is defined as above and was decomposed via SVD as $K = LS_1R^T$ with $L, S_1, R \in \mathbb{R}^{r+1\times r+1}$ and the singular values denoted by $\sigma_i, i = 1, \ldots, r+1$. The challenge is to update U_0, U_0', V_0' and V_0 such that we can rewrite this expression in the form of the extended SVD in (25) without increasing the dimensions when unnecessary. When changing a column of Z_0 we have to make a case distinction. An update can cause the rank to increase, stay the same or decrease.

1. **Rank increasing update:**
   We replace a column that lies in the span of the others with a column that is linearly independent of the others. This is the case if and only if $\sigma_{r+1} > 0$. To obtain a valid decomposition, the dimensions of the inner matrices have to increase by one so that our update simply is

   $$S_1 \leftarrow S_1, \quad U_1 \leftarrow [U_0, P], \quad U_1' \leftarrow \begin{bmatrix} U_0' & 0 \\ 0 & 1 \end{bmatrix} L, \quad V_1 \leftarrow [V_0, Q], \quad V_1' \leftarrow \begin{bmatrix} V_0' & 0 \\ 0 & 1 \end{bmatrix} R$$

   with U_1' and V_1' regular as a product of a regular and an orthonormal matrix and U_1U_1' and V_1V_1' are orthonormal because U_0U_0' and V_0V_0' are orthonormal and P and Q, respectively, were built to be orthonormal to U_0U_0' and V_0V_0', respectively. This can be seen by substituting the updated versions and comparing to (26):

   $$U_1 U_1' S_1 V_1'^T V_1^T = [U_0, P] \begin{bmatrix} U_0' & 0 \\ 0 & 1 \end{bmatrix} L S_0 R^T \begin{bmatrix} V_0'^T & 0 \\ 0 & 1 \end{bmatrix} [V_0, Q]^T.$$

   For the other cases, we need the inverses of U_1' and V_1', so in every update we keep an updated inverse of both of these matrices. The inverses in this case then are simply updated as

   $$U_1'^{-1} \leftarrow L^T \begin{bmatrix} U_0'^{-1} & 0 \\ 0 & 1 \end{bmatrix}, \quad V_1'^{-1} \leftarrow R^T \begin{bmatrix} V_0'^{-1} & 0 \\ 0 & 1 \end{bmatrix},$$

   because L and R are orthonormal and we're given the inverses of U_0' and V_0' from the step before.

2. **Rank stays the same:**
   This is the case if and only if $\sigma_r > 0$ and $\sigma_{r+1} = 0$. The last singular value is zero and therefore we can truncate it. In all subcases of this case S_1 is updated by taking the r-th leading principal submatrix of S_1. This case actually has two subcases: either the new column is linearly independent of the others or it is not.

   (a) **New column is linearly independent:**
   This is the case if and only if $P \neq 0$ since P is constructed by the Modified Gram-Schmidt step to

be the normalized vector that was projected onto the orthogonal complement of the column space of Z_0. More precisely, iff our column update x is linearly independent to all other columns, then $x \neq \lambda_1 (Z_0)_1 + \dots \lambda_n (Z_0)_n$ for all choices of $\lambda_1, \dots, \lambda_n \in \mathbb{R}$. This holds if and only if

$$p = (I - U_0'^T U_0^T U_0 U_0')x \neq \lambda_1 (I - U_0'^T U_0^T U_0 U_0')(Z_0)_1 + \dots + \lambda_n (I - U_0'^T U_0^T U_0 U_0')(Z_0)_n = 0,$$

because the orthogonal projection of $(Z_0)_i, i = 1, \dots, n$ onto the complement of the column space of Z_0 is zero.

In this subcase we further obtain $Q = e_i - V_0 V_0' n = 0$ as the i-th row of $V_0 V_0'$, namely n, is already orthogonal to all others and is of length 1, as the following Lemma shows.

Lemma 4.6.1. *Consider the compact SVD of the rank-r matrix $A = USV^T$ with $U \in \mathbb{R}^{m \times r}$, $S \in \mathbb{R}^{r \times r}$ and $V \in \mathbb{R}^{n \times r}$. For the i-th column of A, A_i, and the i-th row of V, v_i, then it holds that $A_i \notin$ Span$(A_1, \dots, A_{i-1}, A_{i+1}, \dots, A_n)$ if and only if $\langle v_i, v_j \rangle = \delta_{ij} \ \ \forall j = 1, \dots, n$.*

Proof. See appendix. □

That means, K is of the form

$$K = \begin{bmatrix} S_0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} m \\ \|p\| \end{bmatrix} \begin{bmatrix} n \\ \|q\| \end{bmatrix} = \begin{bmatrix} S_0 + mn^T & 0 \\ \|p\|\, n & 0 \end{bmatrix}$$

and therefore, R is of the the form

$$R = \begin{bmatrix} R_{1:r,1:r} & 0 \\ 0 & 1 \end{bmatrix},$$

which means we can truncate R to the r-th leading principal submatrix to obtain the right singular vector updates:

$$V_1' \leftarrow V_0' R_{1:r,1:r}, \quad V_1 \leftarrow V_0.$$

This can be verified again by comparing to (26):

$$V_1 V_1' = V_0 V_0' R_{1:r,1:r} = [V_0, Q] \begin{bmatrix} V_0' & 0 \\ 0 & 1 \end{bmatrix} R_{:,1:r}.$$

Due to the simple form of V_1' and $R_{1:r,1:r}$ being regular, its inverse is easily updated and given by

$$V_1'^{-1} \leftarrow R_{1:r,1:r}^T V_0'^{-1}.$$

The left singular vector updates are more difficult to establish as we append a column to U_0 while U_1' has a somewhat more complex structure. In its original update form the left singular vectors are given by

$$[U_0, P] \begin{bmatrix} U_0' & 0 \\ 0 & 1 \end{bmatrix} L = [U_0, P] \begin{bmatrix} U_0' & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} W & y \\ w & x \end{bmatrix} = [U_0, P] \begin{bmatrix} U_0' W & U_0' y \\ w & x \end{bmatrix}$$

where $W \in \mathbb{R}^{r \times r}$ is the r-the leading principle submatrix and $y \in \mathbb{R}^{r \times 1}$, $w \in \mathbb{R}^{1 \times r}$ and $x \in \mathbb{R}$ (not to

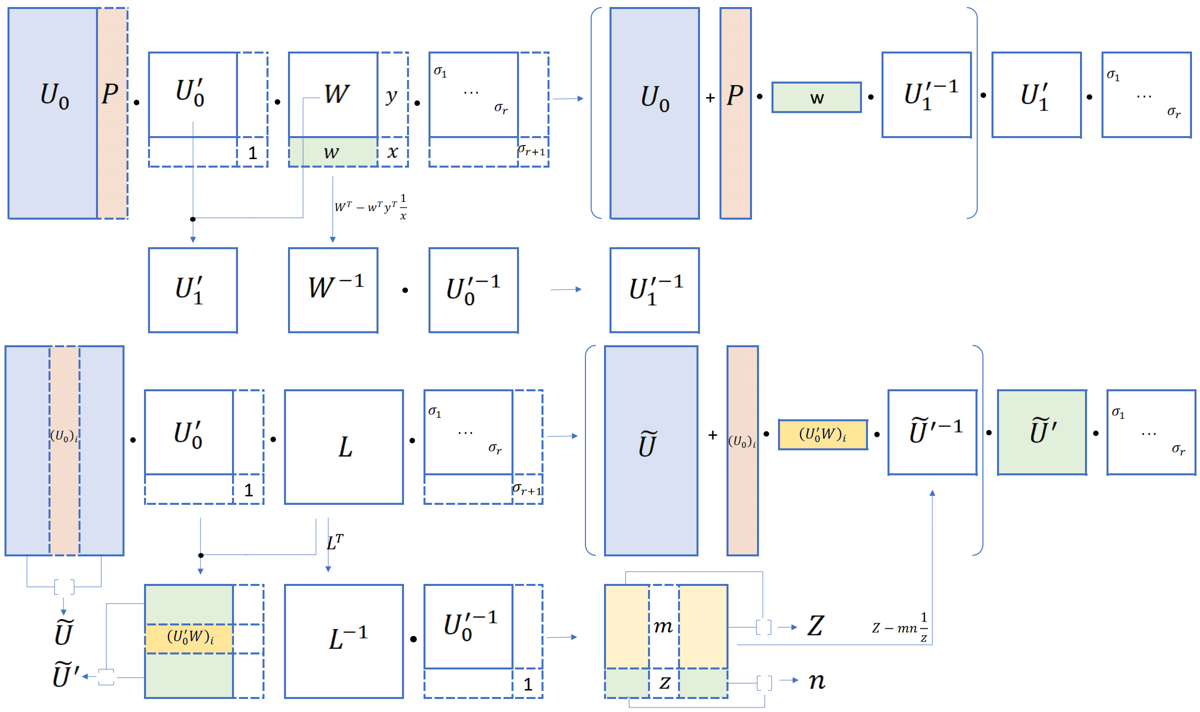

Figure 2: Scheme for updating the extended SVD. Top: Update if $x \neq 0$, bottom: Update if $x = 0$.

be confused with the update vector x). Now, the last column of L (and therefore also of the product of the extended U_0'-matrix and L) represents an unused subspace dimension (i.e. it would be multiplied by zero in the diagonal matrix S_1) and therefore should be suppressed which we will do in the update later. If W is regular, we can further rewrite the left singular vectors (with the last column suppressed) as

$$U_0 U_0' W + Pw = (U_0 + PwW^{-1}U_0'^{-1})U_0'W.$$

such that our update is given by

$$U_1' \leftarrow U_0'W, \quad U_1'^{-1} \leftarrow W^{-1}U_0'^{-1}, \quad U_1 \leftarrow U_0 + PwU_1'^{-1},$$

where the inverse W^{-1} exists iff $x \neq 0$ and as a submatrix of an orthonormal matrix can efficiently be computed by the following Lemma (proof in appendix) which is just an application of the Shermann-Woodburry formula.

Lemma 4.6.2. *Let* $L = \begin{bmatrix} W & y \\ w & x \end{bmatrix} \in \mathbb{R}^{r+1\times r+1}$ *be an orthonormal matrix, where* $w \in \mathbb{R}^{1\times r}$ *and* $y \in \mathbb{R}^{r\times 1}$*. Then the* r*-th leading principle submatrix* W *is invertible if and only if* $x \neq 0$ *and in this case the inverse is given by*

$$W^{-1} = W^T - \frac{w^T y^T}{x}.$$

In the unlikely event of W being singular, we can still extract a subspace dimension using the following Lemma, which is a just a generalization of the previous Lemma.

Lemma 4.6.3. *Let $L \in \mathbb{R}^{r+1 \times r+1}$ be regular. Then we can always remove the last column and at least one row such that the remaining submatrix is regular. The submatrix obtained by deleting the last column and the i-th row is regular if and only if $L_{i,r+1} \neq 0 \neq (L^{-1})_{r+1,i}$. The inverse is given by*

$$Z - \frac{mn}{z}$$

where $Z \in \mathbb{R}^{r \times r}$ is the submatrix of L^{-1} obtained by deleting the i-th column and the last row, $m \in \mathbb{R}^{r \times 1}$ is the i-th column of L^{-1} with the last entry deleted and $n \in \mathbb{R}^{1 \times r}$ is the last row of L^{-1} with the i-th entry deleted and $z := (L^{-1})_{r+1,r+1}$.

Proof. See appendix. □

Instead of extracting the last row, we extract the i-th row for an i that fulfills $L_{i,r+1} \neq 0 \neq (L^{-1})_{r+1,i}$. Let $\tilde{U}$ denote the matrix $[U_0, P]$ whose i-th column $(U_0)_i$ is deleted and likewise let $\tilde{U}'$ denote the matrix $\begin{bmatrix} U_0'W \\ w \end{bmatrix}$ whose i-th row $(U_0'W)_i$ is deleted. With the inverse $\tilde{U}'^{-1}$ of $\tilde{U}'$ we then obtain

$$[U_0, P] \begin{bmatrix} U_0'W \\ w \end{bmatrix} = \tilde{U}\tilde{U}' + (U_0)_i (U_0'W)_i = (\tilde{U} + (U_0)_i (U_0'W)_i \tilde{U}'^{-1})\tilde{U}'$$

such that the update becomes

$$U_1' \leftarrow \tilde{U}', \qquad U_1'^{-1} \leftarrow \tilde{U}', \qquad U_1 \leftarrow \tilde{U} + (U_0)_i (U_0'W)_i \tilde{U}'^{-1}.$$

Algorithm 5 Extract subspace dimension in extended decomposition routine

procedure Extract a subspace dimension and return inverse of submatrix
  Given: $[U_0, P] \in \mathbb{R}^{m \times r+1}, U_0', U_0'^{-1} \in \mathbb{R}^{r \times r}, L \in \mathbb{R}^{r+1 \times r+1}$ with L orthogonal
  if $L_{r+1,r+1} >$ tol
    $U_1' \leftarrow U_0' L_{1:r,1:r},\ \ U_1'^{-1} \leftarrow L_{1:r,1:r}^T U_0'^{-1},\ \ U_1 \leftarrow U_0 + P L_{r+1,1:r} U_1'^{-1}$
  else
    $U_0' \leftarrow \begin{bmatrix} U_0' & 0 \\ 0 & 1 \end{bmatrix} L,\ \ U_0'^{-1} \leftarrow L^T \begin{bmatrix} U_0'^{-1} & 0 \\ 0 & 1 \end{bmatrix}$
    for $j = 1$ until r do
      if $(U_0')_{j,r+1} >$ tol and $(U_0'^{-1})_{r+1,j} >$ tol do
        $k \leftarrow j$
        break
      end if
    end while
    $U_0' \leftarrow (U_0')_{:,1:r}$, $u \leftarrow (U_0')_{k,:}$, $U_1' \leftarrow (U_0')_{1:k-1 \wedge j+1,r+1,:}$,
    $U_1'^{-1} \leftarrow (U_0'^{-1})_{1:k-1 \wedge k+1:r+1,1:r} - (U_0'^{-1})_{1:k-1 \wedge k+1,r+1} (U_0'^{-1})_{k,1:r} / (U_0'^{-1})_{r+1,r+1}$
    $U_1 \leftarrow [U_0, P]_{:,1:k-1 \wedge k+1:r+1} + [U_0, P]_{:,k} u U_1'^{-1}$
  end if
  Output: $U_1, U_1', U_1'^{-1}$

(b) **New column is linearly dependent:**

In this case, we have $P = 0$ and K is of the form

$$\begin{bmatrix} S_0 + mn^T & \|q\| \, m \\ 0 & 0 \end{bmatrix}$$

and therefore the left singular vector matrix L is of the form

$$\begin{bmatrix} L_{1:r,1:r} & 0 \\ 0 & 1 \end{bmatrix}$$

so that our left singular vectors satisfy

$$[U_0, P] \begin{bmatrix} U_0' & 0 \\ 0 & 1 \end{bmatrix} L = U_0 U_0' L_{1:r,1:r}$$

such that our update simply is

$$U_1 \leftarrow U_0, \qquad U_1' \leftarrow U_0' L_{1:r,1:r}, \qquad U_1'^{-1} \leftarrow L_{1:r,1:r}^T U_0'^{-1}.$$

The right singular vectors need to be treated similar to the left singular vectors from previous section. Again, we're suppressing the last column of $R = \begin{bmatrix} X & g \\ v & z \end{bmatrix}$ and if the r-th leading principal submatrix X (not to be confused with the best approximation) is invertible (i.e. $z \neq 0$) we obtain the identity

$$[V_0, Q] \begin{bmatrix} V_0' & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ v \end{bmatrix} = V_0 V_0' X + Qv = (V_0 + QvX^{-1} V_0'^{-1}) V_0' X$$

such that the updates are

$$V_1 \leftarrow V_0 + QvV_1'^{-1}, \qquad V_1' \leftarrow V_0' X, \qquad V_1'^{-1} \leftarrow X^{-1} V_0'^{-1}.$$

If X is not invertible, we look for a row to extract similar to the previous section, say the i-th row can be extracted such that the remaining matrix is regular. If $\tilde{V}$ denotes $[V, Q]$ with the i-th column $(V_0)_i$ deleted and $\tilde{V}'$ denotes $\begin{bmatrix} V_0' X \\ v \end{bmatrix}$ with the i-th row $(V_0' X)_i'$ deleted, then we again obtain the identity

$$[V_0, Q] \begin{bmatrix} V_0' X \\ v \end{bmatrix} = \tilde{V} \tilde{V}' + (V_0)_i (V_0' X)_i' = (\tilde{V} + (V_0)_i (V_0' X)_i' \tilde{V}'^{-1}) \tilde{V}'$$

such that the updates are given by

$$V_1 \leftarrow \tilde{V} + (V_0)_i (V_0' X)_i' \tilde{V}'^{-1}, \qquad V_1' \leftarrow \tilde{V}', \qquad V_1'^{-1} \leftarrow \tilde{V}'^{-1}.$$

3. **Rank is decreasing:**
In this case, we replace a column that is linearly independent of the others with a column that is linearly dependent of the others. Thus, $P = 0$ and $Q = 0$ similar to previous cases. Hence, it suffices to compute

the SVD of $S_0 + mn^T$ which is the truncated K-matrix (instead of having the matrix with appended zero row and column). With the SVD $S_0 + mn^T = \tilde{L}\tilde{S}_1\tilde{R}^T$ where $\tilde{L}, \tilde{R} \in \mathbb{R}^{r\times r}$ are orthonormal and $\tilde{S}_1 \in \mathbb{R}^{r\times r}$ is diagonal, we immediately obtain

$$[U_0, P]\begin{bmatrix} U_0' & 0 \\ 0 & 1 \end{bmatrix} LS_1R^T \begin{bmatrix} V_0' & 0 \\ 0 & 1 \end{bmatrix}^T [V_0, Q]^T = U_0U_0'\tilde{L}\tilde{S}_1\tilde{R}^T V_0'^T V_0^T.$$

In the following, we write L instead of $\tilde{L}$ and R instead of $\tilde{R}$ and the new singular value matrix is immediately updated by taking the $(r-1)$-th leading principal submatrix of $\tilde{S}_1$. As we are in the rank-decreasing case, there is a yet a dimension to extract. This will be done similarly as before. We omit a detailed analysis and just state the update. If we can extract the i-th row from $U_0'L$ such that the matrix obtained from deleting the i-th row $u := (U_0'L)_i$ and the last column (call the matrix $\tilde{U}'$) is invertible (finding i and computing the inverse by means of Lemma 4.6.3) then with $\tilde{U}$ denoting the matrix U_0 with the i-th column $(U_0)_i$ deleted, we obtain the identity

$$U_0U_0'L = \tilde{U}\tilde{U}' + (U_0)_iu = (\tilde{U} + (U_0)_iu\tilde{U}'^{-1})\tilde{U}'$$

such that the left singular vector update is

$$U_1 \leftarrow \tilde{U} + (U_0)_iu, \qquad U_1' \leftarrow \tilde{U}', \qquad U_1'^{-1} \leftarrow \tilde{U}'^{-1}.$$

Finally, with $\tilde{V}'$ denoting $V_0'R$ with the k-th row v and the last column deleted (such that it becomes invertible by means of Lemma 4.6.3) and $\tilde{V}$ denoting V_0 with the k-th column $(V_0)_k$ deleted, we obtain the updates

$$V_1 \leftarrow \tilde{V} + (V_0)_kv, \qquad V_1' \leftarrow \tilde{V}', \qquad V_1'^{-1} \leftarrow \tilde{V}'^{-1}.$$

4.6.2 Extended Decomposition for Appending a Column

This problem is closely related to the rank-1 additive modification problem. As in the normal LRA decomposition, we append a row of zeros to V_0 to obtain the decomposition

$$U_0S_0[V_0^T, 0] = [Z_0, 0]$$

that is to be updated via a rank-1 additive modification. In the rank appending problem, we only have the cases that the rank increases or stays the same.

1. **Rank increasing update:**
   In the rank increasing case, the update is the same as in the section before except that we append a row to V_0 such that the update for the outer right singular vector matrix is

   $$V_1 \leftarrow \begin{bmatrix} V_0 & 0 \\ 0 & 1 \end{bmatrix}.$$

   since $Q = e_{n+1}$.

2. **Rank stays the same:**
   The only other case is that the appended column already lies in the same subspace yielding $P = 0$ which

brings us to case 2 (b) of last section. Again, the updates are the same except for the right outer singular vector matrix. With the same notation as before, for $R = \begin{bmatrix} X & g \\ v & z \end{bmatrix}$ we already know the invertibility of X as the r-th principle leading submatrix of K, S_0, is also regular:

$$K = \begin{bmatrix} S_0 & m \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} (S_1)_{1:r,1:r} \end{bmatrix} \begin{bmatrix} X^T & v^T \\ g^T & z \end{bmatrix} = \begin{bmatrix} W(S_1)_{1:r,1:r}X^T & W(S_1)_{1:r,1:r}wT \\ 0 & 0 \end{bmatrix}.$$

Therefore, the update is

$$V_1 \leftarrow \begin{bmatrix} V_0 \\ 0 \end{bmatrix} + QvV_1'^{-1} = \begin{bmatrix} V_0 \\ vV_1'^{-1} \end{bmatrix}.$$

4.6.3 Remarks on the Extended Decomposition Updating Procedure

- The theoretical algorithm relies on rank-detection via modified Gram-Schmidt. That this can become troublesome for badly conditioned matrices is well-known. We will have to say more on choosing the criterion for which we decide linear dependence later in the chapter Numerical Experiments.

- The theoretical orthonormality of the product U_1U_1' and V_1V_1' is simply seen by comparing to the general additive modification identity (21). However, note that in every update the inner singular vector matrices are multiplied with L and R, respectively, which causes loss of orthogonality over thousands of updates. Brand [12] suggests to occasionally form the product $U_0'S_0V_0'^T$ and recompute the SVD. This does not increase the overall computational complexity. However, it is not clear how often this should be done to obtain a satisfying numerical precision.

4.6.4 Computational Complexity of the Extended Decomposition Updating Procedure

The modified Gram-Schmidt part essentially is the multiplication of $U_0U_0'^T x$ which takes $\mathcal{O}(mr)$ operations. The rediagonalization of K in the column appending procedure costs $\mathcal{O}(r^2)$ steps using the sparse diagonalization described in [21] or $\mathcal{O}(r^3)$ in the column modification procedure using traditional SVD. Finally the subspace rotations only cost $\mathcal{O}(r^3)$ since we leave out the large outer matrices and reduce the updating process to rotating the smaller inner matrices which have dimension $r \times r$. So the complexity in each update is $\mathcal{O}(mr + r^3)$. If we want to compute a complete compact SVD of $A_0 \in \mathbb{R}^{m \times n}$ from scratch we can do this in $\mathcal{O}(mnr + nr^3)$ steps via successively adding the columns of A_0 to our LRA up to that point. For the assumption that the rank r of A_0 is small relative to the matrix dimensions, that is, we assume $r = \mathcal{O}(\sqrt{m})$, then we have a linear-time (in the rank) LRA algorithm, that is, it takes $\mathcal{O}(mnr)$ steps to compute a compact singular value decomposition/LRA. However, in practice we face full rank matrices and also over periods of updates the rank of our LRA becomes too large and the computational complexity disadvantageous. Therefore, the next section deals with truncation strategies.

4.6.5 Truncation Strategies for the Extended LRA

The linear time in the previous LRA algorithm can only be achieved if the rank of our sequentially obtained compact SVD/LRA will not go beyond $r = \sqrt{m}$. For sequential updates that increase the rank beyond $\sqrt{m}$, we may have to truncate rank-increasing updates. Two strategies that immediately come to mind are the following:

1. Ignore any new subspace components,
2. Optimal Greedy Tactic.

1. Ignore the components of an update that lie outside the current subspace (column space of LRA) by setting $\|p\| = 0$ for every update we encounter. This strategy takes into account only the components that lie in the current subspace already. If the component that lies outside would make a large contribution, then clearly this approach will capture the update quite badly as will be seen in Numerical Tests later. The error of such an update is simply seen to be

$$\begin{aligned} &\left\| [U_0, P] \begin{bmatrix} S_0 + mn^T & \|q\|\, m \\ \|p\|\, n^T & \|p\|\, \|q\| \end{bmatrix} [V_0, Q]^T - [U_0, P] \begin{bmatrix} S_0 + mn^T & \|q\|\, m \\ 0 & 0 \end{bmatrix} [V_0, Q]^T \right\|_2 \\ = &\left\| \begin{bmatrix} 0 & 0 \\ \|p\|\, n^T & \|p\|\, \|q\| \end{bmatrix} \right\|_2 = \|p\| \left\| [n^T, \|q\|] \right\| = \|p\| \sqrt{n^T n + 1 - n^T n} = \|p\| \,. \end{aligned}$$

Although $\|p\|$ can be quite large, it is still better than ignoring the update entirely: Let $Z_0 = U_0 S_0 V_0^T$ denote the LRA whose i-th column $(Z_0)_i$ was changed to $(Z_1)_i$ and Z_1 denote the corresponding updated LRA. Then,

$$\|p\| = \left\| (I - U_0 U_0^T)((Z_1)_i - (Z_0)_i) \right\| \leq \left\| I - U_0 U_0^T \right\|_2 \|(Z_1)_i - (Z_0)_i\| = \underbrace{\left\| I - U_0 U_0^T \right\|_2}_{=1} \|Z_1 - Z_0\|_2 \,.$$

For example, for the following minimalistic example,

$$A_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = e_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} e_1^T, \quad A_1 = \begin{bmatrix} 1 & 10 \\ 0 & 1 \end{bmatrix},$$

it holds a strict inequality since $\|A_1 - A_0\|_2 = \left\| [10, 1]^T \right\| = \sqrt{101}$ and

$$\|p\| = \left\| (I - e_1 e_1^T) \begin{bmatrix} 10 \\ 1 \end{bmatrix} \right\| = \left\| \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\| = 1.$$

Numerical Experiments will illustrate the accumulation of errors for different classes of matrices later. The same results also hold for the column appending problem.

2. The optimal greedy tactic is to allow a rank-increasing update but immediately truncate the smallest singular value triplet to retain a rank-r-approximation. This truncation strategy can retain a good approximation for certain classes of matrices as we have seen from Theorem 5. The algorithm changes only slightly: If the rank does not increase, there is no change. If it increases, we set the smallest singular value σ_{r+1} to zero and suppress the last column of L and R. Then again, we extract a subspace dimension by means of the sequential additive modification section. This is basically the strategy already introduced in Algorithm 4, just with higher efficiency. The computational complexity also stays at $\mathcal{O}(mr + r^3)$. If our data matrix is of the form as in the first theorem, then this procedure stays exact over the course of all column appends. Naturally, if it is not of that form, over the period of updates we will not obtain the best rank-r approximation. It remains to be shown in numerical tests later how bad it will influence the approximation.

Both, the stopping and the truncation criterion effect the computational complexity to stay at $\mathcal{O}(mr + r^3)$ for an arbitrary desired rank r or $\mathcal{O}(mnr)$ for $r \leq \sqrt{m}$. We summarized the extended LRA additive modification algorithm in Algorithm 6 with the two possible "stopping" criteria. It is stated such that the criterion goes into effect as soon as $r > \sqrt{m}$. However, other thresholds are possible as well. Algorithm 7 is the column appending analog. Algorithm 8 describes a complete matrix update $Z_0 + \Delta A$ via sequentially adding the columns of the update. Finally, Algorithm 9 computes a LRA from scratch by means of sequentially appending columns. Each of these algorithms can either be called with the stopping criterion or with the truncation criterion. We extend the respective Algorithm number to distinguish, which Algorithm we mean (e.g. Algorithm 6.1 vs Algorithm 6.2). We analyze and compare their behavior in numerical tests later.

Algorithm 6 Extended LRA Additive Modification

procedure COMPUTE UPDATED EXTENDED LRA OF $Z_0 + xy^T$
  Given: extended r-LRA $U_0', V_0', U_0, V_0, U_0'^{-1}, V_0'^{-1}$, S_0, column update x, column index j
  if not set, initialize with $U_0' = V_0' = U_0'^{-1} = V_0'^{-1} = I_{r\times r}$,
  $m \leftarrow U_0'^T U_0^T x, \quad p \leftarrow x - U_0 U_0' m, \quad n \leftarrow V_0'^T V_0^T e_j, \quad q \leftarrow e_j - V_0 V_0' e_j$
  if $r+1 > \sqrt{m}$, $\|p\| >$ tol, $\|q\| >$ tol do —— % Stopping criterion:
    $\|p\| \leftarrow 0$ —— % ignore new
  end if —— % subspace information
  if $\|p\|, \|q\| >$ tol do
    Set $K \leftarrow \begin{bmatrix} S_0 + mn^T & m\|q\| \\ n^T\|p\| & \|p\|\|q\| \end{bmatrix}, \quad L, S_1, R \leftarrow \texttt{svd}(K)$
    $U_1 \leftarrow [U_0, p/\|p\|]$, $V_1 \leftarrow [V_0, q/\|q\|]$, $U_1' \leftarrow \begin{bmatrix} U_0' & 0 \\ 0 & 1 \end{bmatrix} L$, $V_1' \leftarrow \begin{bmatrix} V_0' & 0 \\ 0 & 1 \end{bmatrix} R$
    $U_1'^{-1} \leftarrow L^T \begin{bmatrix} U_0'^{-1} & 0 \\ 0 & 1 \end{bmatrix}$, $V_1'^{-1} \leftarrow R^T \begin{bmatrix} V_0'^{-1} & 0 \\ 0 & 1 \end{bmatrix}$
  if $r+1 > \sqrt{m}$ do —— % Truncation criterion:
    $U_1, U_1', U_1'^{-1} \leftarrow$ Algorithm 5($[U_0, p/\|p\|], U_0', U_0'^{-1}, L$) —— % truncate smallest singular
    $V_1, V_1', V_1'^{-1} \leftarrow$ Algorithm 5($[V_0, q/\|q\|], V_0', V_0'^{-1}, R$) —— % value triplet
    $S_1 \leftarrow (S_1)_{1:r,1:r}$ —— %
  end if —— %
  else if $\|p\| >$ tol, $\|q\| <$ tol do
    $K \leftarrow \begin{bmatrix} S + mn^T \\ n^T\|p\| \end{bmatrix}, \quad L, S_1, R \leftarrow \texttt{svd}(K)$
    $U_1, U_1', U_1'^{-1} \leftarrow$ Algorithm 5($[U_0, p/\|p\|], U_0', U_0'^{-1}, L$)
    $V_1 \leftarrow V_0$, $V_1' \leftarrow V_0' R$, $V_1'^{-1} \leftarrow R^T V_0'^{-1}$, $S_1 \leftarrow (S_1)_{1:r,:}$
  else if $\|p\| <$ tol, $\|q\| >$ tol do
    $K \leftarrow [S_0 + mn^T, \|q\| m], \quad L, S_1, R \leftarrow \texttt{svd}(K)$
    $V_1, V_1', V_1'^{-1} \leftarrow$ Algorithm 5 ($[V_0, q/\|q\|], V_0', V_0'^{-1}, R$)
    $U_1 \leftarrow U_0$, $U_1' \leftarrow U_0' L$, $U_1'^{-1} \leftarrow L^T U_0'^{-1}$, $S_1 \leftarrow (S_1)_{:,1:r}$
  else do
    $K \leftarrow S_0 + mn^T, \quad L, S_1, R \leftarrow \texttt{svd}(K)$
    $U_1, U_1', U_1'^{-1} \leftarrow$ Algorithm 5 ($[U_0, p\|p\|], U_0', U_0'^{-1}, L$)
    $V_1, V_1', V_1'^{-1} \leftarrow$ Algorithm 5 ($[V_0, q\|q\|], V_0', V_0'^{-1}, R$)
  end if
  output: $U_1, U_1', U_1'^{-1}, S_1, V_1, V_1', V_1'^{-1}$

Algorithm 7 Extended LRA Column Appending

procedure COMPUTE UPDATED EXTENDED LRA OF $[Z_0, x]$
    Given: extended r-LRA $U_0, U_0', U_0'^{-1}, S_0, V_0, V_0', V_0'^{-1}$, column update x
    $V_0 \leftarrow \begin{bmatrix} V_0 \\ 0_{1\times r} \end{bmatrix}$
    $U_1, U_1', U_1'^{-1}, S_1, V_1, V_1', V_1'^{-1} \leftarrow$ Algorithm 6$(U_0, U_0', U_0'^{-1}, S_0, V_0, V_0', V_0'^{-1}, x, n+1)$
    output: $U_1, U_1', U_1'^{-1}, S_1, V_1, V_1', V_1'^{-1}$

Algorithm 8 Complete Additive Modification of LRA via sequentially adding columns

procedure COMPUTE UPDATED LOW RANK-APPROXIMATION OF $Z_0 + \Delta A$
    Given: r-LRA $Z_0 = U_0 S_0 V_0^T$, update matrix $\Delta A \in \mathbb{R}^{m\times n}$
    $U_0', V_0', V_0'^{-1}, U_0'^{-1} \leftarrow I_{r\times r}$
    for $j = 1 : n$
        $U_0, U_0', U_0'^{-1}, S_0, V_0, V_0', V_0'^{-1} \leftarrow$ Algorithm 6$(U_0, U_0', U_0'^{-1}, S_0, V_0, V_0', V_0'^{-1}, (\Delta A)_j, j)$
    end for
    $U_1 \leftarrow U_0 U_0', V_1 \leftarrow V_0 V_0'$
    output: U_1, S_0, V_1

Algorithm 9 Compute LRA via sequentially appending columns

procedure COMPUTE r-LRA OF A
    Given: $A \in \mathbb{R}^{m\times n}$
    $U_0', V_0', V_0'^{-1}, U_0'^{-1}, V_0 \leftarrow 1$, $U_0 \leftarrow (A)_1 / \|(A)_1\|$, $S_0 \leftarrow \|(A)_0\|$,
    for $j = 1 : n$
        $U_0, U_0', U_0'^{-1}, S_0, V_0, V_0', V_0'^{-1} \leftarrow$ Algorithm 7$(U_0, U_0', U_0'^{-1}, S_0, V_0, V_0', V_0'^{-1}, (A)_j, j)$
    end for
    $U_1 \leftarrow U_0 U_0', V_1 \leftarrow V_0 V_0'$
    output: U_1, S_0, V_1

5 Dynamical Low-Rank Approximation

The previous algorithms were discrete algorithms that simply modified the LRA successively. The following approach by [15], however, considers a smooth matrix and its derivative. We will specify the nomenclature needed for this approach. Instead of discrete LRAs Z_0 and Z_1, where the latter matrix is obtained by performing one step with one of the previous algorithms, we consider a matrix $A(t) \in \mathbb{R}^{m \times n}$ depending on a parameter $t \in \mathbb{R}$, which we will call time t, which is to be approximated continuously for every t. With that notation a best rank-r-approximation of $A(t)$ for every t is given by

$$\underset{X(t) \in \mathcal{M}_r}{\arg\min} \|X(t) - A(t)\|,$$

where $\mathcal{M}_r = \mathcal{M}_r^{m \times n}$ denotes the manifold of rank-r matrices of dimensions $m \times n$. As before, we refer to $X(t)$ as a best rank-r-approximation of $A(t)$. As seen before $X(t)$ is given by the truncated SVD of $A(t)$. In this approach, we consider the rank-r-approximation $Y(t) \in \mathcal{M}_r$ that is determined from the condition that the component-wise time derivative $\dot{Y}(t) = \frac{d}{dt} Y(t) \in \mathcal{T}_{Y(t)} \mathcal{M}_r$ (where $\mathcal{T}_{Y(t)} \mathcal{M}_r$ is the tangent space of M_r at $Y(t)$) is given by

$$\underset{\dot{Y}(t) \in \mathcal{T}_{Y(t)} \mathcal{M}_r}{\arg\min} \left\| \dot{Y}(t) - \dot{A}(t) \right\|. \tag{27}$$

We will establish matrix differential equations to solve this problem. For that we need an initial condition for Y, ideally $Y(t_0) = X(t_0)$. Problem (27) yields an initial value problem of nonlinear ordinary differential equations on $\mathcal{M}_r$ that will turn out to be efficiently solvable. The solution $Y(t)$ will be called Dynamical LRA (DLRA). This approach lacks the precision of a direct best rank-r-approximation. However, we will show that this approach has many advantages compared to previous methods and furthermore we will show, under some conditions, good error behavior.

5.1 Unique Decomposition in Tangent Space

Recall that we can rewrite every $Y \in \mathcal{M}_r$ as $Y = USV^T$ using the compact SVD where $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$ are orthonormal and $S \in \mathbb{R}^{r \times r}$ is diagonal and invertible. Whenever, we are given such a decomposition in the following, we will only assume S to be invertible. This decomposition is not unique as for any orthonormal matrices $P, Q \in \mathbb{R}^{r \times r}$, choosing $\hat{U} = UP \in \mathbb{R}^{m \times r}$, $\hat{V} = VQ \in \mathbb{R}^{n \times r}$ and $\hat{S} = P^T SQ \in \mathbb{R}^{r \times r}$ yields a different decomposition

$$\hat{U}\hat{S}\hat{V}^T = UPP^T SQQ^T V^T = USV^T = Y.$$

However, we will obtain uniqueness in the tangent space. Let $\mathcal{V}_{m,r} = \{U \in \mathbb{R}^{m \times r} \mid \text{U orthonormal}\}$ denote the Stiefel manifold ([22]) with dimension $mr - \frac{1}{2} r(r+1)$. For the tangent space at $U \in \mathcal{V}_{m,r}$, consider the curve $\gamma : (-\varepsilon, \varepsilon) \to \mathcal{V}_{m,r}$ where $\gamma(0) = U$. Differentiating both sides of the orthonormal condition $\gamma(t)\gamma(t)^T = I$ then yields $U_\delta^T U + U^T U_\delta = 0$ where $U_\delta = \frac{d\gamma}{dt}(0)$ by the chain rule. A detailed proof can be found in e.g. [22]. Hence we arrive at the tangent space

$$\mathcal{T}_U V_{m,r} = \{U_\delta \in \mathbb{R}^{m \times r} \mid U_\delta^T U + U^T U_\delta = 0\} \ \ (= \{U_\delta \in \mathbb{R}^{m \times r} : U^T U_\delta \in \mathrm{so}(r)\})$$

where $\mathrm{so}(r)$ denotes the space of skew-symmetric real $r \times r$ matrices. Now, we consider the extended tangent map of $(S, U, V) \mapsto Y = USV^T$,

$$\begin{aligned}\mathbb{R}^{r\times r} \times \mathcal{T}_U\mathcal{V}_{m\times r} \times \mathcal{T}_V\mathcal{V}_{n\times r} &\to \mathcal{T}_Y\mathcal{M}_r \times \mathrm{so}(r) \times \mathrm{so}(r),\\ (S_\delta, U_\delta, V_\delta) &\mapsto (U_\delta SV^T + US_\delta V^T + USV_\delta^T, U^TU_\delta, V^TV_\delta).\end{aligned}$$

This map is well-defined since $U^TU_\delta \in \mathbb{R}^{r\times r}$ and $V^TV_\delta \in \mathbb{R}^{r\times r}$ both are seen to be skew-symmetric and matrices of the form $U_\delta SV^T + US_\delta V^T + USV_\delta^T$ for a given decomposition $Y = USV^T$ are contained in the tangent space of $\mathcal{M}_r$ at Y(analogous to the tangent space of the Stiefel manifold).
Furthermore, the null-space is zero: Suppose

$$U^TU_\delta = 0, \tag{28}$$
$$V^TV_\delta = 0, \tag{29}$$
$$U_\delta SV^T + US_\delta V^T + USV_\delta^T = 0. \tag{30}$$

Multiplying (28) and (29) by U and V from the left, respectively, and (30) by U^T from the left and by V from the right yields $U_\delta = V_\delta = S_\delta = 0$. Finally, the dimensions of the vector spaces agree: Since the manifold dimension is the same as the dimension of its tangent space at any point, we can see that the dimension of the left vector space is

$$r^2 + mr - \frac{1}{2}r(r+1) + nr - \frac{1}{2}r(r+1) = r^2 + (m+n)r - r(r+1) = (m+n-1)r.$$

The dimension of $\mathrm{so}(r)$ is $\frac{1}{2}r(r-1)$ and of $\mathcal{T}_Y\mathcal{M}_r$ (or simply $\mathcal{M}_r$) is $(m+n)r - r^2$. A proof can be found in the appendix. Thus, the dimension of the right hand side vector space is $r(r-1) + (m+n-r)r = (m+n-1)r$ yielding the equality of the dimensions. Conclusively, we created an isomorphism between the two vector spaces above and hence every tangent matrix $Y_\delta \in \mathcal{T}_Y\mathcal{M}_r$ is of the form

$$Y_\delta = U_\delta SV^T + US_\delta V^T + USV_\delta^T, \tag{31}$$

where $S_\delta \in \mathbb{R}^{r\times r}$, $U_\delta \in \mathcal{T}_U\mathcal{V}_{m,r}$ and $V_\delta \in \mathcal{T}_V\mathcal{V}_{n\times r}$. What is more, $U_\delta, S_\delta, V_\delta$ are uniquely determined by Y_δ if we impose the following constraints

$$U^TU_\delta = 0, \;\; V^TV_\delta = 0, \tag{32}$$

since multiplying (31) by U^T from the left and by V from the right using (32) yields

$$S_\delta = U^TY_\delta V. \tag{33}$$

Likewise, using (33), and multiplying (31) by V and S^{-1} from the right yields

$$U_\delta = Y_\delta VS^{-1} - UU^TY_\delta VS^{-1} = (I_m - UU^T)Y_\delta VS^{-1}. \tag{34}$$

Lastly, using (33), transposing and multiplying (31) by U and S^{-T} from the right yields

$$V_\delta = Y_\delta^TUS^{-T} - VV^TY_\delta^TUS^{-T} = (I_n - VV^T)Y_\delta^TUS^{-T}, \tag{35}$$

so that every representation of (31) yields the same factors. Analoguous to the previous algorithms, we denote the orthogonal projections onto the column spaces of U and V, and onto their orthogonal complements, respectively, by

$$P_U = UU^T, \;\; P_V = VV^T, \;\; P_U^{\perp} = I_m - P_U, \;\; P_V^{\perp} = I_n - P_V.$$

With that notation, (34) and (35) simplify to

$$\begin{aligned} U_\delta &= P_U^{\perp} Y_\delta V S^{-1}, \\ V_\delta &= P_V^{\perp} Y_\delta^T U S^{-T}. \end{aligned}$$

Finally, we have established an isomorphism between the subspace

$$\{(S_\delta, U_\delta, V_\delta) \in \mathbb{R}^{r\times r} \times \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r} : U^T U_\delta = 0, V^T V_\delta = 0\}$$

and $\mathcal{T}_Y\mathcal{M}_r$ since we could uniquely describe an element $Y_\delta \in \mathcal{T}_Y\mathcal{M}_r$ by the three matrices S_δ, U_δ and V_δ using (32). Hence, we have a unique decomposition in the tangent space, which we use in the following chapter.

5.2 Establishing the Differential Equations

Our minimization condition can equivalently be solved by finding the orthogonal projection using Hilbert's Projection Theorem. We omit the argument t in the following Lemma.

Lemma 5.2.1. *Given $\dot{A} \in \mathbb{R}^{m\times n}$, for $\dot{Y} \in \mathbb{R}^{m\times n}$ it holds that*

$$\dot{Y} = \underset{Y_\delta \in \mathcal{T}_Y\mathcal{M}_r}{\arg\min} \left\| Y_\delta - \dot{A} \right\| \;\;\Leftrightarrow\;\; \langle \dot{Y} - \dot{A}, Y_\delta \rangle = 0 \;\; \textit{for all} \;\; Y_\delta \in \mathcal{T}_Y\mathcal{M}_r. \tag{36}$$

Proof. ($\Leftarrow$) Consider a $\dot{Y} \in \mathcal{T}_Y\mathcal{M}_r$ that satisfies

$$\langle \dot{Y} - \dot{A}, Y_\delta \rangle = 0 \;\text{ for all }\; Y_\delta \in \mathcal{T}_Y\mathcal{M}_r.$$

Then we obtain for all $Y_\delta \in \mathcal{T}_Y\mathcal{M}_r$

$$\left\| Y_\delta - \dot{A} \right\|^2 = \left\| Y_\delta - \dot{Y} + \dot{Y} - \dot{A} \right\|^2 = \left\| Y_\delta - \dot{Y} \right\|^2 + \left\| \dot{Y} - \dot{A} \right\|^2,$$

which yields, after taking the root,

$$\left\| \dot{Y} - \dot{A} \right\| \le \left\| Y_\delta - \dot{A} \right\| \;\text{ for all }\; Y_\delta \in \mathcal{T}_Y\mathcal{M}_r$$

and therefore the claim.

($\Rightarrow$) Conversely, let $\dot{Y} \in \mathcal{T}_Y\mathcal{M}_r$ be a minimizer of the norm. Then for any $Y_\delta \in \mathcal{T}_Y\mathcal{M}_r$ and $\lambda \in \mathbb{R}$ it holds

$$0 \le \left\| (\dot{Y} + \lambda Y_\delta) - \dot{A} \right\|^2 - \left\| \dot{Y} - \dot{A} \right\|^2 = 2\lambda \langle \dot{Y} - \dot{A}, Y_\delta \rangle + \lambda^2 \left\| Y_\delta \right\|^2.$$

Suppose $C := \langle \dot{Y} - \dot{A}, Y_\delta \rangle > 0$, then we obtain for all $\lambda < 0$

$$2\lambda C + \lambda^2 \left\| Y_\delta \right\|^2 \ge 0 \;\;\Leftrightarrow\;\; 2C \le -\lambda \left\| Y_\delta \right\|^2,$$

which yields a contradiction for λ of small magnitude. Analoguosly, suppose $C := \langle \dot{Y} - \dot{A}, Y_\delta \rangle < 0$, then we obtain for all $\lambda > 0$

$$2\lambda C + \lambda^2 \|Y_\delta\|^2 \geq 0 \;\; \Leftrightarrow \;\; 2C \geq -\lambda \|Y_\delta\|^2 ,$$

which yields a contradiction for λ of small magnitude once again. Therefore, $\langle \dot{Y} - \dot{A}, Y_\delta \rangle = 0$ for all $Y_\delta \in \mathcal{T}_Y \mathcal{M}_r$. □

Usually, this identity is referred to to as Galerkin orthogonality on the tangent space $\mathcal{T}_Y \mathcal{M}_r$. Using this identity, we derive differential equations for the factors U, S and V^T that yield our desired Y.

Theorem 5.2.2 ([15]). *Given $\dot{A} \in \mathbb{R}^{m\times n}$, for $Y = USV^T \in \mathcal{M}_r$, where $U \in \mathbb{R}^{m\times r}, V \in \mathbb{R}^{n\times r}$ both are orthonormal and $S \in \mathbb{R}^{r\times r}$ is invertible with $U^T\dot{U} = V^T\dot{V} = 0$, any of the statements of Lemma 5.2.1 is equivalent to*

$$\dot{Y} = \dot{U}SV^T + U\dot{S}V^T + US\dot{V}^T, \tag{37}$$

where

$$\begin{aligned} \dot{S} &= U^T \dot{A} V, \\ \dot{U} &= P_U^{\perp} \dot{A} V S^{-1}, \\ \dot{V} &= P_V^{\perp} \dot{A}^T U S^{-T}. \end{aligned}$$

Proof. ($\Rightarrow$) Suppose for $\dot{Y} \in \mathcal{T}_Y \mathcal{M}_r$ it holds $\langle \dot{Y} - \dot{A}, Y_\delta \rangle = 0$ for all $Y_\delta \in \mathcal{T}_Y \mathcal{M}_r$. Then it holds for $Y_\delta = u_i v_j^T$, for all $i, j = 1, ..., r$, where u_i, v_j denote the columns of U, V, respectively, because it is of the form (31) with $U_\delta = V_\delta = 0$ and one entry with value 1 in S_δ. What is more, by the product rule, we obtain from $Y = USV^T$ that

$$\dot{Y} = \dot{U}SV^T + U\dot{S}V^T + US\dot{V}^T. \tag{38}$$

We use the identity $\langle uv^T, B \rangle = u^T B v$ for $u \in \mathbb{R}^m, v \in \mathbb{R}^n$ and $B \in \mathbb{R}^{m\times n}$ in the following. With that, it follows that

$$0 = \langle \dot{Y} - \dot{A}, u_i v_j^T \rangle = u_i^T \dot{Y} v_j - u_i^T \dot{A} v_j \;\text{ for all }\; i, j = 1, ..., r. \tag{39}$$

Now, $u_i^T \dot{Y} v_j = (\dot{S})_{ij}$ which follows from mutliplying (38) by U^T and V from the left and right, respectively, and just looking at the ij-th component. Since this holds for all i, j, with (39) we infer

$$\dot{S} = U^T \dot{A} V.$$

Similarly, choosing $Y_\delta = \sum_{j=1}^r u_\delta s_{ij} v_j^T, i = 1, ..., r$, where $u_\delta \in \mathbb{R}^m$ is arbitrary with $U^T u_\delta = 0$ (which condition can be fulfilled as we can always find another orthogonal vector for a given set) is also of the mandatory form, which can be seen after choosing $V_\delta = S_\delta = 0$ and U_δ only having the i-th column non-zero, called u_δ. Substituting Y_δ yields

$$0 = \langle \dot{Y} - \dot{A}, \sum_{j=1}^r u_\delta s_{ij} v_j^T \rangle = \sum_{j=1}^r s_{ij} \langle \dot{Y} - \dot{A}, u_\delta v_j^T \rangle = \sum_{j=1}^r s_{ij} u_\delta^T \left(\dot{Y} - \dot{A} \right) v_j \;\text{ for all }\; i = 1, ..., r.$$

Substituting $\dot{Y}$ from (38) and using $u_\delta^T U = 0$ and $V^T v_j = e_j$ yields

$$0 = \sum_{j=1}^{r} s_{ij} u_\delta^T \left(\dot{U} S V^T - \dot{A} \right) v_j = \sum_{j=1}^{r} s_{ij} u_\delta^T \left(\dot{U} (S)_j - \dot{A} v_j \right).$$

This holds for all i and therefore we can rewrite it in matrix form as

$$0 = u_\delta^T (\dot{U} S^2 - \dot{A} V S) = u_\delta^T (\dot{U} S - \dot{A} V) S \quad \Leftrightarrow \quad 0 = u_\delta^T (\dot{U} S - \dot{A} V),$$

since S is regular. As u_δ is an arbitrary vector from the orthogonal complement of the column space of U, we can apply the projection operator to yield

$$0 = P_U^\perp (\dot{U} S - \dot{A} V) = \dot{U} S - U U^T \dot{U} S - \dot{A} V + U U^T \dot{A} V \quad \Leftrightarrow \quad \dot{U} S = P_U^\perp \dot{A} V \quad \Leftrightarrow \quad \dot{U} = P_U^\perp \dot{A} V S^{-1}.$$

The derivation for $Y_\delta = \sum_{j=1}^{r} u_j s_{ji} v_\delta^T$ with $V^T v_\delta = 0$ is analogous.
($\Leftarrow$) Since the different choices of Y_δ span the tangent space, by reversing the above steps we obtain the other implication and therefore the equivalence.

$\square$

5.3 Behaviour of the Problem

It is not yet clear, how well the solution $Y(t)$ approximates $A(t)$ or how close it is to the best-approximation $X(t)$. An easy to see characteristic is that the U and V stay orthonormal, since we imposed the constraints $U^T \dot{U} = V^t \dot{V} = 0$. Namely, because of them we obtain $\frac{d}{dt} U^T \dot{U} = \dot{U}^T U + U^T \dot{U} = 0$, meaning no change in the initial orthogonality.
The overall analytic approximation quality, however, can suffer from discontinuities of $X(t)$ as the following simple example shows: Consider the problem of finding a rank-1 approximation of

$$A(t) = \begin{bmatrix} e^{-t} & 0 \\ 0 & e^t \end{bmatrix}.$$

Starting from $t_0 < 0$ and setting our initial condition as $Y(t_0) = X(t_0)$ we obtain for $t < 0$ that

$$X(t) = \begin{bmatrix} e^{-t} & 0 \\ 0 & 0 \end{bmatrix},$$

since for negative t it holds that e^{-t} is the greatest singular value. Similarly, we find that the best-approximation in the tangent space of $Y(t)$ to $\dot{A} = \mathrm{diag}(-e^{-t}, e^t)$ is $\mathrm{diag}(-e^{-t}, 0)$ and therefore $Y(t) = \mathrm{diag}(e^{-t}, 0)$. However, for $t > 0$ we find that $X(t) = \mathrm{diag}(0, e^t)$ since $e^t > e^{-t}$ but $Y(t) = \mathrm{diag}(e^{-t}, 0)$. This difference is due to the discontinuity of X in $t = 0$ where singular values cross from inside to outside of the approximation of $Y(t)$. In the following we provide an analysis that shows that $Y(t)$ yields a near-optimal solutions on intervals where a good smooth approximation exists.

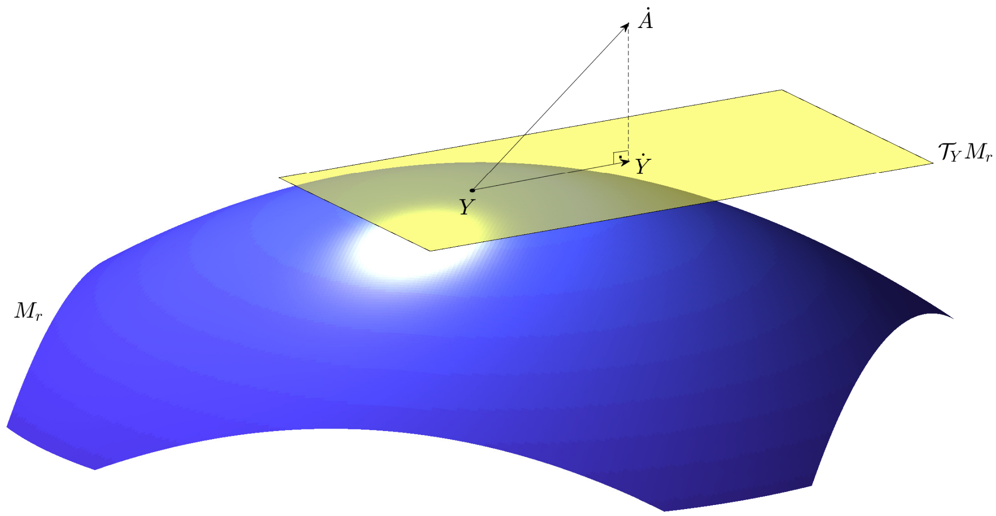

Figure 3: By projecting $\dot{A}$ onto the tangent space $\mathcal{T}_Y\mathcal{M}_r$ at Y we obtain $\dot{Y}$.

5.3.1 Curvature Bounds

In this section, we present some results about the error behavior in different scenarios that will also be looked at in numerical tests later. Recall that $\dot{Y}$ is given by the orthogonality condition

$$\langle \dot{Y} - \dot{A}, Y_\delta \rangle = 0 \ \text{ for all } \ Y_\delta \in \mathcal{T}_Y\mathcal{M}_r. \tag{40}$$

In other words, we can express $\dot{Y}$ in terms of the orthogonal projection of $\dot{A}$:

$$\dot{Y} = P(Y)\dot{A}, \tag{41}$$

where $P(Y)$ is the orthogonal projection onto the tangent space $\mathcal{T}_Y\mathcal{M}_r$ at Y.
The following Lemma gives an representation of this projection:

Lemma 5.3.1 ([15])**.** *The orthogonal projection onto the tangent space $\mathcal{T}_Y\mathcal{M}_r$ at $Y = USV^T \in \mathcal{M}_r$ is given by*

$$P(Y) = I - P^\perp(Y) \ \textit{ with } \ P^\perp(Y)Z = P_U^\perp Z P_V^\perp$$

for $Z \in \mathbb{R}^{m\times n}$.

Proof. From Theorem (5.2.2) we know that

$$\dot{Y} = \dot{U}SV^T + U\dot{S}V^T + US\dot{V}^T = P_U^\perp \dot{A}VV^T + \dot{A} + UU^T\dot{A}P_V^\perp = \dot{A} - P_U^\perp\dot{A}P_V^\perp = (I - P^\perp(Y))\dot{A},$$

with the corresponding representations for $\dot{U}, \dot{V}$ and $\dot{S}$. This holds for every matrix $\dot{A}$ and hence we conclude the claim. ☐

The following Lemma is a preparatory tool used for the approximation theorem later. It provides bounds for the curvature of the manifold $\mathcal{M}_r$ at Y.

Lemma 5.3.2 ([15])**.** *Consider the rank-r matrix $X \in \mathcal{M}_r$ such that its smallest nonzero singular value satisfies $\sigma_r(X) \geq \rho > 0$. Furthermore, let $Y \in \mathcal{M}_r$ with $\|Y - X\| \leq \frac{1}{8}\rho$. Then for all $B \in \mathbb{R}^{m\times n}$:*

$$\|(P(Y) - P(X))B\| \leq 8\rho^{-1} \|Y - X\| \|B\|_2 , \tag{42}$$

$$\left\|P^{\perp}(Y)(Y - X)\right\| \leq 4\rho^{-1} \|Y - X\|^2 \tag{43}$$

Proof. We're going to split up the proof in sections (a)-(e).
(a)
Decomposing $X = U_0 S_0 V_0^T \in \mathcal{M}_r$ using compact SVD and using the condition that the smallest singular value is bounded from below by ρ, the greatest singular value of the matrix formed by the inverses (if zero, then leave it zero) is bounded by ρ^{-1} from above, namely $\left\|S_0^{-1}\right\|_2 \leq \rho^{-1}$. From [23], Corollary 7.3.5, we know

$$|\sigma_r(Y) - \sigma_r(X)| \leq \|Y - X\|_2 \leq \|Y - X\| .$$

Furthermore, using the condition $\|Y - X\| \leq \frac{1}{8}\rho$ and $\sigma_r(X) \geq \rho$, we obtain

$$\sigma_r(Y) \geq \sigma_r(X) - |\sigma_r(Y) - \sigma_r(X)| \geq \rho - \frac{1}{8}\rho = \frac{7}{8}\rho.$$

This in turn yields via SVD $Y = U_1 S_1 V_1^T$ with $\left\|S_1^{-1}\right\|_2 \leq \frac{8}{7}\rho^{-1}$ where we formed S_1^{-1} again by inverting all non-zero diagonal entries.
(b)
We decompose every matrix on the line connecting X and Y into a sum of a rank-r matrix and a matrix orthogonal to the tangent space of $\mathcal{M}_r$ at X as such:

$$X + \tau(Y - X) = M(\tau) + N(\tau) \quad \text{for all } 0 \leq \tau \leq 1 \text{ with } N(\tau) \perp \mathcal{T}_X\mathcal{M}_r.$$

The existence of such a decomposition will become clear by the arguments that follow. For that, denote

$$\Delta = P(X)(Y - X) \in \mathcal{T}_X\mathcal{M}_r \text{ while } \|\Delta\| = \|P(X)(Y - X)\| \leq \|P(X)\| \|Y - X\| = \|Y - X\| =: \delta$$

with submultiplicativity and the unitary boundedness of the orthogonal projection. Applying the orthogonal projector yields

$$P(X)(M(\tau) - X) = P(X)[\tau(Y - X) + N(\tau)] = P(X)(\tau(Y - X)) = \tau\Delta,$$

as $N(\tau)$ lies in the orthogonal complement. Deriving with respect to τ yields

$$P(X)\dot{M}(\tau) = \Delta.$$

By Lemma 5.3.1 the orthogonal projector can be rewritten as $\Delta = P(X)\dot{M} = \dot{M} - P_{U_0}^{\perp}\dot{M}P_{V_0}^{\perp}$ and thus, after applying the projectors P_{U_0} and P_{V_0}, we obtain

$$P_{U_0}\Delta = P_{U_0}\dot{M}, \quad \Delta P_{V_0} = \dot{M}P_{V_0}.$$

Since the orthogonal projectors are given easily given as $P_{U_0} = U_0U_0^T$ and $P_{V_0} = V_0V_0^T$, we obtain $U_0U_0^T\Delta = U_0U_0^T\dot{M}$ and $\Delta V_0V_0^T = \dot{M}V_0V_0^T$ and premultiplying and postmultiplying the respective equation by U and V, respectively, this implies

$$U_0^T \dot{M}(\tau) = U_0^T \Delta, \quad \dot{M}(\tau) V_0 = \Delta V_0. \tag{44}$$

(c)
Using Theorem 5.2.2 with $M(\tau) \in \mathcal{M}_r$ in place of $A(\tau)$ and $Y(\tau)$ we obtain $M(\tau) = U(\tau)S(\tau)V(\tau)^T$, where S, U, V are given as the solutions of the differential equations

$$\begin{aligned} \dot{S} &= U^T \dot{M} V, \\ \dot{U} &= P_U^\perp \dot{M} V S^{-1}, \\ \dot{V} &= P_V^\perp \dot{M}^T U S^{-T}. \end{aligned} \tag{45}$$

Since $\dot{M} = \Delta + P_{U_0}^\perp \dot{M} P_{V_0}^\perp$ and by using (44) and the projector property $P_{U_0}^\perp U_0 = 0$ (or $U_0^T P_{U_0}^\perp = (P_{U_0}^\perp U_0)^T = 0$) and $P_{V_0}^\perp V_0$, equations (45) equal to

$$\begin{aligned} U^T \dot{M} V &= U^T \Delta V + U^T P_{U_0}^\perp \dot{M} P_{V_0}^\perp V = U^T \Delta V + (U - U_0)^T P_{U_0}^\perp \dot{M} P_{V_0}^\perp (V - V_0), \\ P_U^\perp \dot{M} V S^{-1} &= P_U^\perp \dot{M}(V - V_0 + V_0) S^{-1} = P_U^\perp \Delta V_0 S^{-1} + P_U^\perp \dot{M}(V - V_0) S^{-1}, \\ P_V^\perp \dot{M}^T U S^{-T} &= P_V^\perp \dot{M}^T (U - U_0 + U_0) S^{-T} = P_V^\perp \Delta^T U_0 S^{-T} + P_V^\perp \dot{M}^T (U - U_0) S^{-T}. \end{aligned} \tag{46}$$

At the same time, deriving M yields

$$\dot{M} = U \dot{S} V^T + (\dot{U} S) V^T + U (\dot{V} S^T)^T. \tag{47}$$

and therefore (by using the projector property again)

$$\dot{M} = \Delta + P_{U_0}^\perp \dot{M} P_{V_0}^\perp = \Delta + P_{U_0}^\perp \left((U - U_0)\dot{S}(V - V_0)^T + (\dot{U}S)(V - V_0)^T + (U - U_0)(\dot{V}S^T)^T \right) P_{V_0}^\perp. \tag{48}$$

In the following we will show that these differential equations have a solution for $\tau \leq 1$. For that, note that in a neighborhood around t_0 it holds

$$\|U - U_0\| \leq \frac{1}{4} \quad \text{and} \quad \|V - V_0\| \leq \frac{1}{4}.$$

Using (45) within this neighborhood we obtain the bounds

$$\begin{aligned} \left\|\dot{S}\right\| &\leq \|\Delta\| + \frac{1}{16}\left\|\dot{M}\right\| \leq \delta + \frac{1}{16}\left\|\dot{M}\right\|, \\ \left\|\dot{U}S\right\| &\leq \|\Delta\| + \frac{1}{4}\left\|\dot{M}\right\| \leq \delta + \frac{1}{4}\left\|\dot{M}\right\|, \\ \left\|\dot{V}S^T\right\| &\leq \|\Delta\| + \frac{1}{4}\left\|\dot{M}\right\| \leq \delta + \frac{1}{4}\left\|\dot{M}\right\|. \end{aligned}$$

by using triangular inequality and unitary invariance. Inserting these bounds into (47) yields

$$\left\|\dot{M}\right\| \leq \left\|\dot{S}\right\| + \left\|\dot{U}S\right\| + \left\|\dot{V}S^T\right\| = 3\delta + \frac{9}{16}\left\|\dot{M}\right\| \;\Rightarrow\; \left\|\dot{M}\right\| \leq 2\delta,$$

and therefore

$$\left\|\dot{S}\right\| \leq \frac{9}{8}\delta, \quad \left\|\dot{U}S\right\| \leq \frac{3}{2}\delta, \quad \left\|\dot{V}S^T\right\| \leq \frac{3}{2}\delta. \tag{49}$$

The bound for $\left\|\dot{S}\right\|$ yields

$$\|S(\tau) - S_0\| = \left\|\int_0^\tau \dot{S}(t)dt\right\| \leq \int_0^\tau \left\|\dot{S}(t)\right\| dt \leq \frac{9}{8}\delta, \tag{50}$$

for $\tau \leq 1$ where the integral is pointwise and we used the fundamental theorem of calculus and the "triangular inequality" of the norm. Using $\delta \leq \frac{1}{8}\rho$ and part (a), it therefore holds $\left\|S(\tau)^{-1}\right\|_2 \leq \frac{4}{3}\rho^{-1}$ for $\tau \leq 1$. From (49) we then obtain with $(\dot{U}S)_j$ denoting the j-th row of $\dot{U}S$

$$\left\|\dot{U}\right\|^2 = \left\|\dot{U}SS^{-1}\right\|^2 = \sum_{j=1}^m \left\|(\dot{U}S)_j S^{-1}\right\|^2 \leq \sum_{j=1}^m \left\|(\dot{U}S)_j\right\|^2 \|S^{-1}\|_2^2 = \left\|\dot{U}S\right\|^2 \|S^{-1}\|_2^2 \leq \frac{3}{2}\delta\frac{4}{3}\rho^{-1} \leq \frac{1}{4}.$$

The bound for $\dot{V}$ can be derived in the same way. Now, since the right hand side of (45) is continuous and bounded for all $\tau \leq 1$, we conclude that the differential equation for M has a solution by Peano's Existence Theorem up to $\tau = 1$ with the bounds

$$\|S_1 - S_0\| \leq \frac{9}{8}\delta, \quad \|U_1 - U_0\| \leq 2\rho^{-1}\delta, \quad \|V - V_0\| \leq 2\rho^{-1}\delta. \tag{51}$$

(d)
From these bounds, we immediately obtain

$$\left\|P_{U_1}^\perp - P_{U_0}^\perp\right\| = \left\|U_0U_0^T - U_1U_1^T\right\| = \left\|U_0U_0^T - U_1U_0^T + U_1U_0^T - U_1U_1^T\right\| \leq \|U_0 - U_1\| + \left\|U_0^T - U_1^T\right\| \leq 4\rho^{-1}\delta, \tag{52}$$

$$\left\|P_{V_1}^\perp - P_{V_0}^\perp\right\| = \left\|V_0V_0^T - V_1V_1^T\right\| = \left\|V_0V_0^T - V_1V_0^T + V_1V_0^T - V_1V_1^T\right\| \leq \|V_0 - V_1\| + \left\|V_0^T - V_1^T\right\| \leq 4\rho^{-1}\delta. \tag{53}$$

From our orthogonal projection representation in Lemma 5.3.1, we see that

$$(P(Y) - P(X))B = (I - P_{U_1}^\perp BP_{V_1}^\perp) - (I - P_{U_0}^\perp BP_{V_0}^\perp) = P_{U_0}^\perp BP_{V_0}^\perp - P_{U_1}^\perp BP_{V_1}^\perp.$$

Combining this with (52) yields

$$\begin{aligned}\|(P(Y) - P(X))B\| &= \left\|P_{U_0}^\perp BP_{V_0}^\perp - P_{U_1}^\perp BP_{V_1}^\perp\right\| = \left\|P_{U_0}^\perp BP_{V_0}^\perp - P_{U_1}^\perp BP_{V_0}^\perp + P_{U_1}^\perp BP_{V_0}^\perp - P_{U_1}^\perp BP_{V_1}^\perp\right\| \\ &\leq \left\|(P_{U_0}^\perp - P_{U_1}^\perp)BP_{V_0}^\perp\right\| + \left\|P_{U_1}^\perp B(P_{V_0}^\perp - P_{V_1}^\perp)\right\| \leq 8\rho^{-1}\delta\,\|B\|_2 = 8\rho^{-1}\,\|Y - X\|\,\|B\|_2\,,\end{aligned}$$

which is the first desired inequality.
(e)
Using $P_U^\perp U = 0$ and $P_V^\perp V = 0$, we obtain

$$\begin{aligned}P^\perp(Y)(Y - X) &= P_{U_1}^\perp(U_1S_1V_1^T - U_0S_0V_0^T)P_{V_1}^\perp = -P_{U_1}^\perp U_0S_0V_0^TP_{V_1}^\perp \\ &= -P_{U_1}^\perp(U_1 - U_0)S_0(V_1 - V_0)^TP_{V_1}^\perp.\end{aligned} \tag{54}$$

Furthermore, we rewrite using the Fundamental Theorem of Calculus component-wise

$$(U_1 - U_0)S_0 = \int_0^1 \dot{U}(\tau)S_0 d\tau = \int_0^1 \dot{U}(\tau)S(\tau)d\tau - \int_0^1 \dot{U}(\tau)(S(\tau) - S_0)d\tau.$$

Thus, taking the norm and using (50), (49) and the bound on $\dot{U}$ yields

$$\|(U_1 - U_0)S_0\| \leq \int_0^1 \left\|\dot{U}(\tau)S(\tau)\right\| d\tau + \int_0^1 \left\|\dot{U}(\tau)\right\| \|(S(\tau) - S_0)\| \, d\tau \leq \frac{3}{2}\delta + \frac{1}{4}\frac{9}{8}\delta \leq 2\delta. \tag{55}$$

Using this bound and from (51) the bound for V to plug both into (54) finally yields

$$\left\|P^{\perp}(Y)(Y - X)\right\| = \left\|P_{U_1}^{\perp}(U_1 - U_0)S_0(V_1 - V_0)^T P_{V_1}^T\right\| \leq 4\delta^2\rho^{-1} = 4\rho^{-1}\|Y - X\|.$$

□

With the help of this Lemma, we can prove different approximation properties of the Dynamical Low-Rank Approximation scheme.

5.3.2 The Case of Continuously Differentiable Best Approximation

If the best-approximation $X(t) \in \mathcal{M}_r$ to $A(t)$ is continuously differentiable, then we can bound the error of the dynamical low-rank approximation in terms of the best-approximation error $\|X(t) - A(t)\|$. For the following theorem, we also assume that

$$\left\|\dot{A}\right\|_2 \leq \mu \quad \text{for} \quad 0 \leq t \leq \bar{t}, \tag{56}$$

where we choose the inital time $t_0 = 0$ for convenience.

Theorem 5.3.3 ([15]). *Suppose there exists a continuously differentiable best-approximation $X(t) \in \mathcal{M}_r$ to $A(t)$ for $0 \leq t \leq \bar{t}$. Denote a lower bound of the r-th singular value of $X(t)$ by $\sigma_r(X(t)) \geq \rho > 0$ and furthermore assume that the best-approximation error is bounded by $\|X(t) - A(t)\| \leq \frac{1}{16}\rho$ for $0 \leq t \leq \bar{t}$. Then the Dynamical Low-Rank Approximation $Y(t)$, given as the solution of (27), with initial value $Y(0) = X(0)$ is bounded by*

$$\|Y(t) - X(t)\| \leq 2\beta e^{\beta t} \int_0^t \|X(s) - A(s)\| \, ds,$$

where $\beta = 8\mu\rho^{-1}$ for $0 \leq t \leq \bar{t}$ and as long as the right-hand side is bounded by $\frac{1}{8}\rho$.

Proof. For readibility, the argument t is sometimes omitted. As X is the best-approximation, $A - X$ is orthogonal to the tangent space $\mathcal{T}_X\mathcal{M}_r$ because as seen before $\|A - X\| = \min!$ is equivalent to $\langle A - X, X_\delta\rangle = 0$ for all $X_\delta \in \mathcal{T}_X\mathcal{M}_r$. Another equivalent reformulation is

$$P(X)(A - X) = 0.$$

Differentiating this expression with respect to t and denoting $(P'(X)B)\dot{X} = \frac{d}{dt}P(X(t))B$ yields

$$P(X)(\dot{A} - \dot{X}) + (P'(X)(A - X))\dot{X} = 0,$$

by product and chain rule. Since the derivative of X, $\dot{X}$, is in the tangent space of X, $\mathcal{T}_X\mathcal{M}_r$, it holds $P(X)\dot{X} = \dot{X}$. Using this in the previous equation yields

$$(I - P'(X)(X - A))\dot{X} = P(X)\dot{A}. \tag{57}$$

With Lemma 14 and the condition $d := \|A - X\| \leq \frac{1}{16}\rho$ we obtain

$$\|P'(X)(X - A)\| \leq 8\rho^{-1}d \leq \frac{1}{2}.$$

Therefore, we rewrite (57) and apply the Neumann series to obtain

$$\dot{X} = \left[\sum_{k=0}^{\infty}(P'(X)\cdot(X - A))^k\right] P(X)\dot{A} = P(X)\dot{A} + \underbrace{\left[\sum_{k=1}^{\infty}(P'(X)(X - A))^k\right] P(X)\dot{A}}_{:=D} \tag{58}$$

where the norm of the latter summand is bounded by

$$\|D\| \leq \mu\sum_{k=1}^{\infty}\|(P'(X)(X - A))\|^k \leq \mu\left(\frac{8\rho^{-1}d}{1 - 8\rho^{-1}d}\right) \leq \mu 16\rho^{-1}d = 2\beta d.$$

By subtracting (58) from $\dot{Y} = P(Y)\dot{A}$ and integrating from 0 to t, we obtain

$$Y - X = \int_0^t (P(Y) - P(X))\dot{A} + Dds. \tag{59}$$

What is more, using the condition $e := \|Y - X\| \leq \frac{1}{8}\rho$ enables us to apply Lemma 5.3.2 to obtain

$$\left\|(P(Y) - P(X))\dot{A}\right\| \leq 8\rho^{-1}e\mu = \beta e.$$

Taking the norm and substituting this inequality in (59) therefore yields

$$e(t) \leq \beta\int_0^t e(s)ds + 2\beta\int_0^t d(s)ds.$$

Now, since the second integral is a monotone function in t, we conclude with the Gronwall inequality

$$e(t) \leq 2\beta e^{\beta t}\int_0^t d(s)ds$$

which is the desired bound. □

5.3.3 Error Bounds for Longer Intervals

The last error bound grew with greater t fast due to the integral and the exponential dependence. In the following, we're going to establish a bound that only grows linearly in t but demands the derivative of $X - A$ to be small.

Assume that $A(t)$ is of the form

$$A(t) = X(t) - E(t), \quad 0 \leq t \leq \bar{t}, \tag{60}$$

where $X(t) \in \mathcal{M}_r$ need not necessarily be the best-approximation. We assume that the derivatives exist and are bounded as

$$\left\|\dot{X}(t)\right\|_2 \leq \mu, \qquad \left\|\dot{E}(t)\right\| \leq \varepsilon, \tag{61}$$

where $\varepsilon \leq \frac{1}{8}\mu$ is assumed. With these conditions we can formulate the next Theorem.

Theorem 5.3.4 ([15]). *In addition to the above assumptions, suppose that the r-th singular value of $X(t)$ satisfies $\sigma_r(X(t)) \geq \rho > 0$. Then the DLRA $Y(t)$ to $A(t)$ with initial value $Y(0) = X(0)$ is bounded by*

$$\|Y(t) - X(t)\| \leq 2t\varepsilon \quad \text{for} \quad 0 \leq t \leq \min(\bar{t}, \frac{\rho}{4\sqrt{2\mu\varepsilon}}).$$

Proof. Again, note that $P(X)\dot{X} = \dot{X}$ and rewrite $\dot{Y} = P(Y)\dot{A}$ as

$$\dot{Y} = P(Y)\dot{X} + P(Y)\dot{E}.$$

Subtracting these two equations yields

$$\dot{Y} - \dot{X} = (P(Y) - P(X))\dot{X} + P(Y)\dot{E}. \tag{62}$$

Furthermore, note that

$$(P(Y) - P(X))\dot{X} = -(P^{\perp}(Y) - P^{\perp}(X))\dot{X}.$$

Taking the inner product of this expression with $Y - X$, yields

$$\begin{aligned}\langle Y - X, (P(Y) - P(X))\dot{X}\rangle = -\langle Y - X, P^{\perp}(Y)\dot{X}\rangle &= -\langle P^{\perp}(Y)(Y - X), P^{\perp}(Y)\dot{X}\rangle \\ &\langle P^{\perp}(Y)(Y - X), (P(Y) - P(X))\dot{X}\rangle,\end{aligned}$$

where we used $-P^{\perp}(Y)\dot{X} = -P^{\perp}(Y)^2\dot{X}$ to include a projection in the first argument of the inner product. Combining (62) with this equality yields

$$\langle Y - X, \dot{Y} - \dot{X}\rangle = \langle P^{\perp}(Y)(Y - X), (P(Y) - P(X))\dot{X}\rangle + \langle Y - X, P(Y)\dot{E}\rangle. \tag{63}$$

Applying Lemma (5.3.2) and the conditions on $\dot{X}$ and $\dot{E}$ then yield with the Cauchy-Schwarz inequality

$$\langle Y - X, \dot{Y} - \dot{X}\rangle \leq 32\mu\rho^{-2}\|Y - X\|^3 + \|Y - X\|\varepsilon.$$

At the same time, it holds

$$\langle Y - X, \dot{Y} - \dot{X}\rangle = \frac{1}{2}\frac{d}{dt}\|Y - X\|^2 = \|Y - X\|\frac{d}{dt}\|Y - X\|$$

by the chain rule. Dividing by $e(t) = \|Y - X\|$ yields the differential inequality

$$\dot{e}(t) \leq \gamma e(t)^2 + \varepsilon, \qquad \text{with initial value} \quad e(0) = 0,$$

where $\gamma = 32\mu\rho^{-2}$.

Therefore, we can majorize $e(t)$ by the solution of the differential equation

$$\dot{y}(t) = \gamma y(t)^2 + \varepsilon, \qquad \text{with intial value} \quad y(0) = 0.$$

The solution is seen to be $y(t) = \sqrt{\varepsilon/\gamma}\tan(t\sqrt{\gamma\varepsilon})$ as $y(0) = 0$ and

$$\dot{y}(t) = \frac{1}{\cos^2(t\sqrt{\gamma\varepsilon})}\varepsilon = \frac{\sin^2(t\sqrt{\gamma\varepsilon})}{\cos^2(t\sqrt{\gamma\varepsilon})}\varepsilon + \varepsilon = \varepsilon\tan^2(t\sqrt{\gamma\varepsilon}) + \varepsilon = \gamma y(t)^2 + \varepsilon$$

while using $1 = \sin^2(x) + \cos^2(x)$. As long as $t\sqrt{\gamma\varepsilon} \leq 1$ we can bound $y(t)$ by $2t\varepsilon$. We can use Lemma 5.3.2 as long as $2t\varepsilon \leq \frac{1}{8}\rho$ which is satisfied on the given interval under the assumption $\varepsilon \leq \frac{1}{8}\mu$ since

$$2t\varepsilon \leq \frac{\rho\varepsilon}{2\sqrt{2\mu\varepsilon}} = \frac{\rho\sqrt{\varepsilon}}{2\sqrt{2\mu}} \leq \frac{\rho}{8}.$$

□

5.3.4 Over-Approximation

The bound established above is only valid for times $t \leq \frac{\rho}{4\sqrt{2\mu\varepsilon}}$. If ρ, the smallest non-zero singular value of $X(t)$, is smaller than ε, this time intervall is very short. This occurs if $A(t)$ has an effective rank (ε-pseudorank) of $q < r$ but the appoximation is done with a rank-r matrix $Y(t)$. This means that $S(t)$ is ill-conditioned as we deal with very small singular values and since its inverse appears in the differential equations for the factors one could expect severe consequences on the approximation properties. The following result, however, shows that this is not the case.

Theorem 5.3.5 ([15]). *Let (60) and (61) hold for $X(t) \in \mathcal{M}_q$ for $q < r$. Suppose furthermore for the smallest singular value of $X(t)$ that $\sigma_q(X(t)) \geq \rho > 0$. Let the initial matrix $Y(0) \in \mathcal{M}_r$ be $Y(0) = X(0) + E_0$ with $\operatorname{Im} E_0 \perp \operatorname{Im} X(0)$ and $\operatorname{Im} E_0^T \perp \operatorname{Im} X(0)^T$ and $\|E_0\| \leq \varepsilon_0 \leq \frac{1}{16}\rho\mu^{-1}\varepsilon$. Finally, assume that the differential equation (37) for the factors has a solution for all $0 \leq t \leq t^*$. Then the DLRA approximation of (27) is bounded by*

$$\|Y(t) - X(t)\| \leq \varepsilon_0 + 6t\varepsilon \quad \textit{for} \quad t \leq \min\left(\bar{t}, t^*, \frac{\rho}{16\mu}\right).$$

Proof. See [15] □

The existence of a the solution had to be assumed as $S(t)$ might become singular. Note that the orthogonality condition is satisfied if $Y(0)$ is the best rank-r approximation: If $A(0) = USV^T$ is the SVD and $Y(0) = U_{1:r}S_{1:r}V_{1:r}^T$ for $U_{1:r} \in \mathbb{R}^{m\times r}$, $S_{1:r} \in \mathbb{R}^{r\times r}$ and $V_{1:r} \in \mathbb{R}^{n\times r}$ is the best-rank-r-approximation and $X(0)$ is the best rank-q approximation $X(0) = U_{1:q}S_{1:q}V_{1:q}^T$, then $E_0 = U_{1:r}S_{1:r}V_{1:r}^T - U_{1:q}S_{1:q}V_{1:q}^T = U_{q+1:r}S_{q+1:r}V_{q+1:r}^T$ and the orthogonality condition is fulfilled: $E_0^T X(0) = V_{q+1:r}S_{q+1:r}U_{q+1:r}^T U_{1:q}S_{1:q}V_{1:q} = 0$ as the columns of U are orthogonal. Likewise for the transposed condition.

5.3.5 Continuous-like Distribution of Singular Values

The error bounds established so far yield good results for systems where there is a gap in the distribution of the singular values such that measurement data and noise are separated. In this subsection we consider the case where the singular values are allowed to spread more evenly and a gap need not exist. For the following theorem, we need the assumption that $X(t)\mathcal{M}_r$ with $\sigma_r(X(t)) \geq \rho > 0$ has a decomposition

$$X(t) = U_0(t)S_0(t)V_0(t)^T \quad \text{for} \quad 0 \leq t \leq \bar{t} \tag{64}$$

with regular $S_0(t) \in \mathbb{R}^{r\times r}$ and with $U_0(t) \in \mathbb{R}^{m\times r}$ and $V_0(t) \in \mathbb{R}^{n\times r}$ orthonormal such that for $0 \leq t \leq \bar{t}$ it holds:

$$\left\|\frac{d}{dt}S_0^{-1}(t)\right\|_2 \leq c_1\rho^{-1}, \quad \left\|\dot{U}_0(t)\right\|_2 \leq c_2, \quad \left\|\dot{V}_0(t)\right\|_2 \leq c_2. \tag{65}$$

Under this condition we formulate the next theorem.

Theorem 5.3.6 ([15])**.** *With the assumptions of Theorem 5.3.4 and above condition (64) and (65), the DLRA approximation error of (27) with initial value $Y(0) = X(0)$ is bounded by*

$$\|Y(t) - X(t)\| \leq 2t\varepsilon \quad \textit{for} \quad t \leq \min\left(\bar{t}, \frac{1}{16c_2^{1/2}}(\frac{\rho}{\varepsilon})^{1/2}, \frac{1}{8c_1^{1/3}}(\frac{\rho}{\varepsilon})^{2/3}, \frac{1}{16}\frac{\rho}{\varepsilon}\right).$$

Proof. From the proof in Theorem 5.3.4, equation (63), we know that

$$\langle Y - X, \dot{Y} - \dot{X}\rangle = -\langle P^\perp(Y)(Y - X), P^\perp(Y)\dot{X}\rangle + \langle Y - X, P(Y)\dot{E}\rangle. \tag{66}$$

With $e = \|Y - X\| \leq \frac{1}{8}\rho$ and using Lemma 5.3.2, inequality (55) shows that Y can be decomposed as $Y = U_1S_1V_1^T$ such that

$$\|(U_1 - U_0)S_0\| \leq 2e, \quad \left\|S_0(V_1 - V_0)^T\right\| \leq 2e \tag{67}$$

(the inequality from Lemma 5.3.2 containing V can be shown analogously to the one containing U). Using Lemma 5.3.1 we can rewrite

$$\begin{aligned} P^\perp(Y)\dot{X} &= P_{U_1}^\perp(\dot{U}_0S_0V_0^T + U_0\dot{S}_0V_0^T + U_0S_0\dot{V}_0^T)P_{V_1}^\perp \\ &= P_{U_1}^\perp(-\dot{U}_0S_0(V_1 - V_0)^T + (U_1 - U_0)S_0S_0^{-1}\dot{S}_0S_0^{-1}S_0(V_1 - V_0)^T - (U_1 - U_0)S_0\dot{V}_0^T)P_{V_1}^\perp, \end{aligned}$$

as the newly introduced terms inside the brackets become zero when multiplied with the projection onto the orthogonal complements of the column spaces U_1 and V_1, respectively. Similarly to Theorem 5.3.4, we obtain by the Cauchy-Schwartz inequality and (66) that

$$\langle Y - X, \dot{Y} - \dot{X}\rangle = e\dot{e} \leq \left\|P^\perp(Y)(Y - X)\right\| \left\|P^\perp(Y)\dot{X}\right\| + e\left\|P(Y)\dot{E}\right\|$$

and by applying Lemma 5.3.2 to the first factor and using the bound on $\dot{E}$ we obtain as long as $e \leq \frac{1}{8}\rho$ that

$$e\dot{e} \leq 4\rho^{-1}e^2\left\|P^\perp(Y)\dot{X}\right\| + e\varepsilon.$$

Finally, by combining the equality for $P^\perp(Y)\dot{X}$ and the bounds (65), (67) we obtain after dividing by e

$$\dot{e} \leq 4\rho^{-1}e(2c_2e + 4c_1\rho^{-1}e^2 + 2c_2e) + \varepsilon,$$

where we used that

$$\frac{d}{dt}S_0^{-1} = -S_0^{-1}\dot{S}_0 S_0^{-1},$$

which can be seen from

$$S_0 S_0^{-1} = I \quad \Rightarrow \quad \frac{d}{dt}(S_0 S_0^{-1}) = \dot{S}_0 S_0^{-1} + S_0 \frac{d}{dt}S_0^{-1} = \dot{I} = 0 \quad \Rightarrow \frac{d}{dt}S_0^{-1} = -S_0^{-1}\dot{S}_0 S_0^{-1}.$$

As long as the first term on the right hand side of the differential inequality is bounded by ε, the overall error is bounded by $2t\varepsilon$ since integrating yields

$$e \leq \int_0^t 2\varepsilon dt = 2t\varepsilon.$$

It is bounded by ε which is satisfied for $16c_2\rho^{-1}(2t\varepsilon)^2 \leq \frac{1}{2}\varepsilon$ and $16c_1\rho^{-2}(2t\varepsilon)^3 \leq \frac{1}{2}\varepsilon$ which itself is satisfied if t fulfills the stated bounds.

□

5.4 Numerical Integration of Differential Equations

As discussed earlier, numerically integrating the differential equations can become computationally very expensive when S is nearly singular. Also in that case, the algorithm suffers from rounding errors due to the ill-conditioned problem. Therefore, in ([24]) an integrator for exactly this problem was introduced that is fully explicit and robust. This integrator has several properties that make it preferable over traditional Runge-Kutta-Solvers; see [24] for a detailed analysis. Therefore, we will solely focus on this integrator. In the following, we detail the projector splitting integrator.

5.4.1 Establishing the Projector Splitting Integrator

Recall that $\dot{Y}$ is given by the orthogonality condition

$$\langle \dot{Y} - \dot{A}, Y_\delta \rangle = 0 \text{ for all } Y_\delta \in \mathcal{T}_Y\mathcal{M}_r. \tag{68}$$

In other words, we can express $\dot{Y}$ in terms of the orthogonal projection of $\dot{A}$:

$$\dot{Y} = P(Y)\dot{A}, \tag{69}$$

where $P(Y)$ is the orthogonal projection onto the tangent space $\mathcal{T}_Y\mathcal{M}_r$ at Y. Lemma 5.3.1 gave an representation of this projection. The orthogonal projection onto the tangent space $\mathcal{T}_Y\mathcal{M}_r$ at $Y = USV^T \in \mathcal{M}_r$ is given by

$$P(Y) = I - P^\perp(Y) \text{ with } P^\perp(Y)Z = P_U^\perp Z P_V^\perp$$

for $Z \in \mathbb{R}^{m\times n}$.
Furthermore, note that the range $\mathcal{R}(Y)$ of $Y = USV^T$ is equal to the range of U ($\mathcal{R}(U)$) since V^T consists of orthogonal rows and S is invertible. Therefore, the orthogonal projector $P_{\mathcal{R}(Y)}$ onto $\mathcal{R}(Y)$ is UU^T. Similarly,

VV^T is the orthogonal projector $P_{\mathcal{R}(Y^T)}$ onto $\mathcal{R}(Y^T)$. Hence we can rewrite $P(Y)Z$ for any $Z \in \mathbb{R}^{m\times n}$ as

$$\begin{aligned} P(Y)Z &= Z - P_U^{\perp} Z P_V^{\perp} = Z - (I - UU^T)Z(I - VV^T) = ZVV^T - UU^T ZVV^T + UU^T Z \\ &= ZP_{\mathcal{R}(Y^T)} - P_{\mathcal{R}(Y)} ZP_{\mathcal{R}(Y^T)} + P_{\mathcal{R}(Y)} Z. \end{aligned} \tag{70}$$

Having splitted the right hand side of (69) into a sum of three summands, we can perform traditional Lie-Trotter splitting. For that, let a rank-r approximation Y_0 to $A(t_0)$ be given and perform one step to $t_1 = t_0 + h$ for some step size h:

1. Solve the deq. $\dot{Y}_I = \dot{A} P_{\mathcal{R}(Y_I^T)}$ with initial value $Y_I(t_0) = Y_0$ on $[t_0, t_1]$.
2. Solve the deq. $\dot{Y}_{II} = -P_{\mathcal{R}(Y_{II})} \dot{A} P_{\mathcal{R}(Y_{II}^T)}$ with initial value $Y_{II}(t_0) = Y_I(t_1)$ on $[t_0, t_1]$.
3. Solve the deq. $\dot{Y}_{III} = P_{\mathcal{R}(Y_{III})} \dot{A}$ with initial value $Y_{III}(t_0) = Y_{II}(t_1)$ on $[t_0, t_1]$.

Our new approximation to the solution of the original problem, $Y(t_1)$, then is taken to be $Y_1 = Y_{III}(t_1)$. The new stated differential equations can be solved exactly in a heuristic way as the following Lemma shows.

Lemma 5.4.1 ([15])**.** *The solution of the first deq. is given by*

$$Y_I(t) = U_I(t) S_I(t) V_I(t)^T \ \ \textit{with} \ \ \frac{d}{dt}(U_I S_I) = \dot{A} V_I, \ \ \dot{V}_I = 0. \tag{71}$$

The solution of the second deq. is given by

$$Y_{II}(t) = U_{II}(t) S_{II}(t) V_{II}(t)^T \ \ \textit{with} \ \ \dot{S}_{II} = -\dot{U}_{II}^T \dot{A} V_{II}, \ \ \dot{U}_{II} = 0, \ \ \dot{V}_{II} = 0. \tag{72}$$

The solution of the third deq. is given by

$$Y_{III}(t) = U_{III}(t) S_{III}(t) V_{III}(t)^T \ \ \textit{with} \ \ \frac{d}{dt}(V_{III} S_{III}^T) = \dot{A}^T U_{III}, \ \ \dot{U}_{III} = 0. \tag{73}$$

Remark 5.4.2. *The differential equations in (71), (72) and (73) can be solved trivially. Their solutions are obtained by integrating both sides from t_0 to t yielding for all $t \in [t_0, t_1]$*

$$\begin{aligned} &U_I(t)S_I(t) - U_I(t_0)S_I(t_0) = (A(t) - A(t_0)V_I(t_0), \ \ V_I(t) = V_I(t_0), \\ &S_{II}(t) - S_{II}(t_0) = U_{II}(t_0)^T (A(t) - A(t_0)) V_{II}(t_0), \ \ U_{II}(t) = U_{II}(t_0), \ \ V_{II}(t) = V_{II}(t_0), \\ &V_{III}(t) S_{III}(t)^T - V_{III}(t_0) S_{III}(t_0)^T = (A(t) - A(t_0))^T U_{III}(t_0), \ \ U_{III}(t) = U_{III}(t_0). \end{aligned}$$

Proof of Lemma 5.4.1. Every summand of the splitted representation (70) is in the tangent space $\mathcal{T}_Y \mathcal{M}_r$. This can be seen by using the orthonormality of U and V, respectively, and applying the projector on each summand: For a $Z \in \mathbb{R}^{m\times n}$ we obtain for the first summand ZVV^T:

$$\begin{aligned} P(Y)(ZVV^T) &= ZVV^T VV^T - UU^T ZVV^T VV^T + UU^T ZVV^T \\ &= ZVV^T + UU^T ZVV^T - UU^T ZVV^T = ZVV^T, \end{aligned}$$

which means that the first summand ZVV^T already was in the tangent space $\mathcal{T}_Y \mathcal{M}_r$. Similarly, for the second summand UU^TZVV^T we obtain after applying the projector

$$\begin{aligned}P(Y)(UU^TZVV^T) &= UU^TZVV^TVV^T - UU^TUU^TZVV^TVV^T + UU^TUU^TZVV^T \\ &= UU^TZVV^T - UU^TZVV^T + UU^TZVV^T = UU^TZVV^T,\end{aligned}$$

which means that this summand also was in the tangent space. Lastly, the last summand UU^TZ is also in the tangent space which again can be seen by applying the projector:

$$\begin{aligned}P(Y)(UU^TZ) &= UU^TZVV^T - UU^TUU^TZVV^T + UU^TUU^TZ \\ &= UU^TZVV^T - UU^TZVV^T + UU^TZ = UU^TZ.\end{aligned}$$

That means that the solutions for 1., 2. and 3. stay in $\mathcal{M}_r$. For the first equation, we can therefore factorize Y_I as $Y_I(t) = U_I(t)S_I(t)V_I^T(t)$ where $S_I \in \mathbb{R}^{r\times r}$ is invertible and $U_I \in \mathbb{R}^{m\times r}, V_I \in \mathbb{R}^{n\times r}$ are orthonormal. Differentiating Y_I yields (for representation purposes we use "$\frac{d}{dt}$" and "$\dot{}$" interchangably)

$$\dot{Y}_I = [\frac{d}{dt}(U_IS_I)]V_I^T + (U_IS_I)\dot{V}_I^T,$$

which is well-defined since the differential equations for U_IS_I and V_I were seen to be solvable. By the first differential equation it must also hold that $\dot{Y}_I = \dot{A}V_IV_I^T$. Comparing these equations, we see that the right hand sides are equal if $[\frac{d}{dt}(U_IS_I)] = \dot{A}V_I$ and $\dot{V}_I = 0$ because then

$$\dot{A}V_IV_I^T = [\frac{d}{dt}(U_IS_I)]V_I^T + (U_IS_I)\dot{V}_I^T.$$

For the second differential equation, again we can factorize its solution as $Y_{II}(t) = U_{II}S_{II}V_{II}^T$ for orthonormal U, V and invertible S. Differentiating Y_{II} (which again is well-defined by the same reason) yields

$$\dot{Y}_{II} = \dot{U}_{II}S_{II}V_{II} + U_{II}\dot{S}_{II}V_{II}^T + U_{II}S_{II}\dot{V}_{II}^T.$$

At the same time, by the second differential equation Y_{II} must also satisfy $\dot{Y}_{II} = -U_{II}U_{II}^T\dot{A}V_{II}V_{II}^T$. They are equal if $\dot{S}_{II} = -U_{II}^T\dot{A}V_{II}$ and $\dot{U}_{II} = \dot{V}_{II} = 0$. Similarly, we can factorize the last solution as $Y_{III} = U_{III}S_{III}V_{III}^T$ for orthonormal U, V and invertible S. Differentiating yields

$$\dot{Y}_{III} = \dot{U}_{III}S_{III}V_{III}^T + U_{III}\frac{d}{dt}(S_{III}V_{III}^T).$$

From the third differential equation, Y_{III} must also satisfy $\dot{Y}_{III} = UU^T\dot{A}$ and both expressions are equal if $\frac{d}{dt}(S_{III}V_{III}^T) = U_{III}^T\dot{A}$ and $\dot{U}_{III} = 0$ which concludes the proof. □

5.4.2 Practical Algorithm with Splitted Integrator

With the Lie-Trotter splitting and its abstract formulation and the solutions of the differential equations given in Lemma 5.4.1, we propose the following algorithm. We consider one step from time t_0 to t_1. Given a factorization of the r-LRA $Y_0 = U_0S_0V_0^T$ that approximates A_0 and the increment $\Delta A = A_1 - A_0$ we obtain the updated DLRA $Y_1 = U_1S_1V_1^T$ by updating the factors as described in Algorithm 10. Since we could analytically solve the differential equations stated in Lemma 5.4.1 the output given in above algorithm is the exact solution of the abstract Lie-Trotter splitting algorithm. From the viewpoint of abstract splitting algorithms, this is a first-order splitting algorithm. It could be extended to higher order splitting algorithms (see [24]). However, here we restrict our analysis to the first-order algorithm as above.

Algorithm 10 Dynamical Low-Rank-Approximation - Additive Modifications

1: **procedure** COMPUTE r-DLRA OF $A_1 \coloneqq A_0 + \Delta A$
2: Given: r-LRA $Y_0 = USV^T$ and modification ΔA
3: $K \leftarrow US + \Delta AV$
4: $[U, \hat{S}] = \mathtt{qr}(K)$ (thin variant)
5: $\tilde{S} \leftarrow \hat{S} - U^T \Delta AV$
6: $K \leftarrow V\tilde{S}^T + \Delta A^T U$
7: $[V, S] = \mathtt{qr}(K)$ (thin variant)
8: output: U, S^T, V

5.4.3 Computational Complexity

The algorithm only relies on matrix multiplications and the QR decomposition. Computing ΔAV and $\Delta A^T U$ each take mnr operations. The other multiplications and the QR decomposition take $\mathcal{O}(mr^2)$ and $\mathcal{O}(nr^2)$ operations. The algorithm is stated for general ΔA matrices. However, if we're given updates of the form CD^T for $C \in \mathbb{R}^{m\times c}$ and $D \in \mathbb{R}^{n\times c}$, the complexity of the matrix multiplications ΔAV and $\Delta A^T U$ falls to $\mathcal{O}(mrc)$ and the overall complexity to $\mathcal{O}(mn \max(r, c))$. Furthermore, this algorithms can make use of sparse updates. The bottleneck is the multiplication of ΔAV and $\Delta A^T U$ which very much profits from sparse matrices ΔA. In the numerical tests, we illustrate this advantage.

5.4.4 The Case of Over-Approximation

As described earlier, solving the differential equations established in Theorem 5.2.2 can be problematic when S becomes singular. This is the case when the rank of $A(t)$ is smaller than the chosen approximation rank r because then the singular values of S become very small. However, the integrator we reviewed in the previous subsection does not suffer from this property. The following theorem only holds for the algorithm we reviewed. For a different ordering of the splitted projectors, the result does not hold.

Theorem 5.4.3 ([24])**.** *Suppose rank(A_0), rank$(A_1) \leq r$ for $A_1 = A_0 + \Delta A$. Given the exact initial value $Y_0 = A_0 = U_0 S_0 V_0^T$ with U_0, V_0 orthonormal and $S_0 \in \mathbb{R}^{r\times r}$ and assume for $A_1 = U_1 S_1 V_1^T$ with U_1, V_1 orthonormal and $S_1 \in \mathbb{R}^{r\times r}$ that $V_1^T V_0$ is regular, then the splitting algorithm DLRA given in Algorithm 10 is exact, i.e. $Y_1 = A_1$.*

Proof. We can decompose $A(t) = U(t)S(t)V(t)^T$ for $t = t_0, t_1$ where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ are orthonormal and $S \in \mathbb{R}^{r\times r}$ invertible with the compact singular value decomposition. As in the algorithm, computing the QR-decomposition of K and using the decompositions in the following yields

$$U_1\hat{S} = K = U_0 S_0 + \Delta AV_0 = U_0 S_0 + (A_1 - A_0)V_0 = A_1 V_0 = U(t_1)S(t_1)V(t_1)^T V_0.$$

Postmultiplying above equation by $(V_1 V_0)^{-1} V_1^T$, we obtain

$$U_1\hat{S}(V_1^T V_0)^{-1} V_1^T = A_1.$$

Pre-multiplying that by $U_1U_1^T$, we obtain

$$U_1U_1^TA_1 = A_1, \qquad \text{and likewise} \qquad A_0V_0V_0^T = A_0.$$

Finally, using the formulas of the splitting algorithm in reverse order yields

$$\begin{aligned}
Y_1 &= U_1S_1V_1^T \\
&= U_1\tilde{S}V_0^T + U_1U_1^T\Delta A \\
&= U_1\hat{S}V_0^T - U_1U_1^T\Delta AV_0V_0^T + U_1U_1^T\Delta A \\
&= U_0S_0V_0^T + \Delta AV_0V_0^T - U_1U_1^T\Delta AV_0V_0^T + U_1U_1^T\Delta A \\
&= A_0 + A_1V_0V_0^T - A_0V_0V_0^T - U_1U_1^TA_1V_0V_0^T - U_1U_1^TA_0V_0V_0^T + U_1U_1^TA_1 - U_1U_1^TA_0 \\
&= A_0 + A_1V_0V_0^T - A_0 - A_1V_0V_0^T - U_1U_1^TA_0 + A_1 - U_1U_1^TA_0 \\
&= A_1,
\end{aligned}$$

which concludes the claim. □

Although in practice mostly fulfilled, the condition $V_1^TV_0$ invertible cannot be dropped as for example for the following over-approximation example the DLRA projection splitting integrator fails to yield an exact update. Consider the 2-DLRA approximation to

$$A_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = U_0S_0V_0^T = Y_0, \qquad \Delta A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The exact solution naturally is $A_1 = A_0 + \Delta A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ whereas the projection splitting algorithm yields

$$K = U_0S_0 + \Delta AV_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = U_1\hat{S}, \quad \tilde{S} = \hat{S} - U_1^T\Delta AV_0 = \hat{S}$$

$$L = V_0\tilde{S}^T + (\Delta A)^TU_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = V_1S_1$$

and hence $Y_1 = U_1S_1V_1^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \neq A_1$. Also, in the over-approximation case it can still fail to yield the exact solution if $V_1^TV_0$ is not invertible. For that, consider again the 2-DLRA to

$$A_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = U_0S_0V_0^T = Y_0, \qquad \Delta A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

which yields $Y_1 = U_1 S_1 V_1^T = U_0 S_0 V_0^T \neq A_1$.
Although in practice we mostly encounter full rank matrices, Theorem 5.4.3 actually offers an efficient method to compute an exact compact SVD of low-rank $A \in \mathbb{R}^{m\times n}$ with $r = \text{rank}(A)$ by choosing our initial matrix $A_0 = I_{m\times r} 0_{r\times r} I_{n\times r}^T$ and $\Delta A = A$. This yields a decomposition $A = USV^T$ with $U \in \mathbb{R}^{m\times r}$, $V \in \mathbb{R}^{n\times r}$ orthonormal and $S \in \mathbb{R}^{r\times r}$ the diagonal singular value matrix in $\mathcal{O}(mnr)$ operations given $V_{1:r,1:r}$ is regular.

5.4.5 Invariance under Presence of Small Singular Values

We conclude the overview over this integrator by reviewing an error analysis for this integrator. In the theoretical analysis of the previous sections, we saw that from a theoretical standpoint small singular values do not have a dramatic effect on the theoretical DLRA solution. However, the presence of small singular values may cause the integrator to attain very small step sizes and may eventually cause standard numerical integrators to fail and/or become inefficient. Another remarkable property of the projector splitting integrator from last section is that it is insensitive to the presence of small singular values. In [25] it is shown theoretically and in numerical tests that the proposed integrator outperforms traditional integrators. In particular, it is proven the following bound on the DLRA solution which is a special case of a more general result.

Theorem 5.4.4 ([25]). *Let $A(t) = X(t) + R(t)$ with $X(t) \in \mathcal{M}_r$ and $\|R(t_0)\| \leq \delta$, $\left\|\dot{R}(t)\right\| \leq \varepsilon$. Then for some step size h the integrator from Algorithm 10 after n steps is bounded as*

$$\|Y_n - A(t_n)\| \leq \delta + 7(t_n - t_0)\varepsilon \quad \textit{for} \quad t_0 \leq t_n \leq T.$$

Proof. See [25]. □

Whenever we compare other low-rank updating methods to DLRA, we use this integrator as a representative.

Table 1: Computational Complexity Overview of all the algorithms: Throughout the table we assume $m \geq n$, $A \in \mathbb{R}^{m\times n}$ and $Z_0, Y_0 = U_0 S_0 V_0^T$ denote its r-(D)LRA

#	Given	Goal	$\mathcal{O}(\cdot)$
1	A	$A = USV^T$	$4mn^2 - \frac{4}{3}n^3$
2	$Z_0 = U_0 S_0 V_0^T$, $X \in \mathbb{R}^{m\times c}, Y \in \mathbb{R}^{n\times c}$	$Z_0 + XY^T = U_1 S_1 V_1^T$	$(m+n)(2rc + c^2 + (r+c)^2)$ $+\frac{8}{3}(r+c)^3$
3	$Z_0 = U_0 S_0 V_0^T$,$C \in \mathbb{R}^{m\times c}$	$[Z_0, C] = U_1 S_1 V_1^T$	$2mrc + mc^2$ $+(m + n + \frac{8}{3}(r+c))(r+c)^2$
4	$[Z_0, C] = U_0 S_0 V_0^T, C \in \mathbb{R}^{m\times c}$	$Z_0 = U_1 S_1 V_1^T$	$(m+n)r^2 + nc(r+c) + \frac{8}{3}(r+c)^3$
5	$U_0 \in \mathbb{R}^{m\times r+1}, U_0'$, $U_0'^{-1} \in \mathbb{R}^{r\times r}, L \in \mathbb{R}^{r+1\times r+1}$	$U_1 \in \mathbb{R}^{m\times r}, U_1', U_1'^{-1} \in \mathbb{R}^{r\times r}$	$2(r^3 + mr)$
6	$Z_0 = U_0 U_0' S_0 V_0'^T V_0^T, x, e_j \in \mathbb{R}^m$	$Z_0 + xe_j^T = U_1 U_1' S_1 V_1'^T V_1^T$	$(2m+n)r + \frac{20}{3}(r+1)^3$
7	$Z_0 = U_0 U_0' S_0 V_0'^T V_0^T, x \in \mathbb{R}^m$	$[Z_0, x] = U_1 U_1' S_1 V_1'^T V_1'^T$	$2mr + \frac{20}{3}(r+1)^3$
8	$Z_0 = U_0 S_0 V_0^T$, $\Delta A \in \mathbb{R}^{m\times n}$	$Z_0 + \Delta A = U_1 S_1 V_1^T$	$(2m+n)rn + \frac{20}{3}(r+1)^3 n$
9	A	$Z_0 = U_0 S_0 V_0^T$	$2mnr + \frac{20}{3}(r+1)^3 n$
10	$(A \approx) Y_0 = U_0 S_0 V_0^T$, $\Delta A \in \mathbb{R}^{m\times r}$	$(A + \Delta A \approx) Y_1 = U_1 S_1 V_1^T$	$2mnr$

6 Numerical Experiments

We will compare Algorithms 8.1, 8.2, 9.1, 9.2 and 10 with one another with respect to their approximation properties. $Z^{8.1}$, $Z^{8.2}$, $Z^{9.1}$, $Z^{9.2}$ and Y refer to the algorithms or the approximations we obtain by applying the respective algorithm. X denotes the best rank-r approximation. The approximation rank is $r = \sqrt{m}$ unless otherwise stated. We chose these algorithms as their complexity is similar ($\mathcal{O}(mnr)$).

6.1 Comments on Criterion for Linear Dependence

Although it is more stable than the Classical Gram Schmidt procedure (e.g. [26],[27]), the Modified Gram Schmidt procedure still depends on the condition number of our matrix [28]. To overcome the problems occuring in ill-conditioned matrices, re-orthogonalization was proposed (e.g. [26]). In those algorithms, the computed vector p will again be orthogonalized in the same way. Many different variations of the number of iteration steps and the stopping criterion exist and were examined. E.g., in [29] an error analysis for exactly two iteration steps was given. For other algorithms that iterate more often, a selective re-orthogonalization criterion is needed. In [30] a criterion was proposed for checking the quality of the computed basis which depends on a single parameter σ. In [31] a rounding error analysis shows under some technical assumptions the convergence of the iterative algorithm or alternatively that the criterion continually fails to be satisfied. For our purposes, this criterion would yield an alternative to simply taking the norm $\|p\|$ as a threshold for deciding whether we are in the rank-deficient case or not. Furthermore, for very ill-conditioned matrices there are methods to estimate the condition number of the extended orthonormal matrix $[U_0, p/\|p\|]$ and raise a flag if it is large (such as in [32]). In our numerical comparisons, we restrict our analysis on "not too ill-conditioned matrices" as for this class of matrices, stability remains a problem. In detecting linear dependence, we restrict the analysis to simply deciding whether the norm $\|p\|$ is below a certain threshold. For larger matrices, one may have to use the "twice is enough"-convention suggested in [33] and formally demonstrated in [34] meaning that one has to perform one re-orthogonalization step and decide that a vector is considered linearly dependent if

$$\|p'\| \leq c \|x\|$$

for p', the re-orthogonalized vector, and some constant c close to machine precision.

6.2 LRAs of Matrices with Low Numerical Rank

We follow a model problem from [15]. We construct time-dependent matrices

$$A(t) = Q_1(t)(A_1 + \exp(t)A_2)Q_2(t) \in \mathbb{R}^{m \times n},$$

where $Q_1(t)$ and $Q_2(t)$ are given as the solutions of the ordinary differential equations

$$\dot{Q}_i = T_i Q_i, \quad i = 1, 2$$

for some random skew-symmetric matrices T_i with values between -1 and 1. For A_1 and A_2 we first start by taking I_{10} plus a matrix with entries uniformly distributed over $[0, 1/2]$. This matrix is set as the leading principal submatrix of A_1 and similarly for A_2. To each A_1 and A_2 is added a (different) pertubation matrix generated as a random matrix with entries uniformly distributed over $[0, \varepsilon]$ for some small ε. In this way, for small ε, $A(t)$ is close to a rank-10 matrix. Figure 4 shows the distribution of the entries. The orthogonal

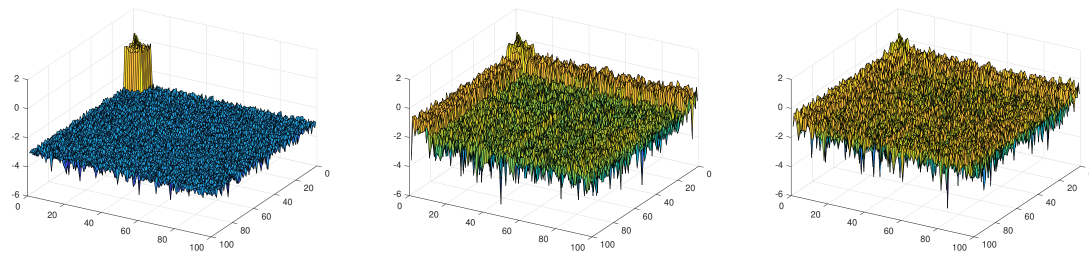

Figure 4: Size of the matrix elements of $\log_{10}(|A(t)|)$ for $t = 0, 0.5, 1$ and $\varepsilon = 10^{-3}$

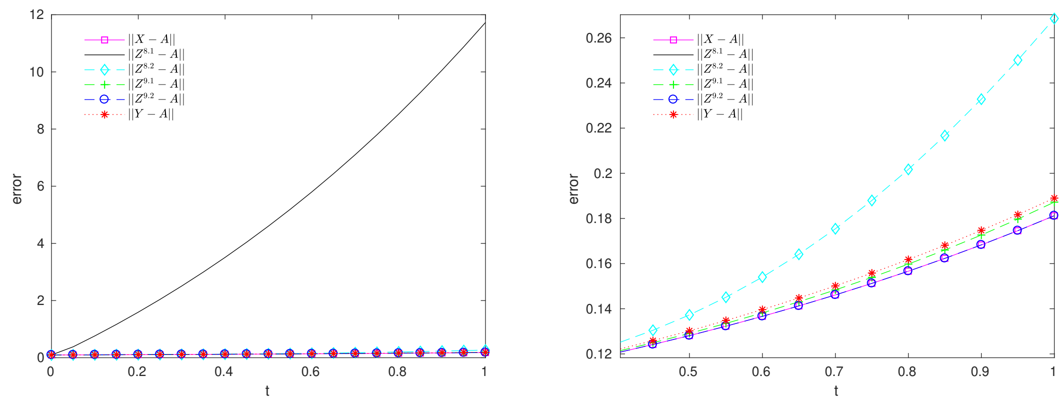

Figure 5: Reconstruction Frobenius error as function of t; left: zoomed out plot, right: zoomed in.

transformations spread the block over the whole matrix such that the size becomes approximately the same all over. At first, we choose $\varepsilon = 10^{-3}$ and $m = n = 100$ where we consider rank-10 approximations. In Figure 5 we plotted the reconstruction Frobenius error $\|A(t) - X(t)\|, \|A(t) - Y(t)\|$ and $\|A(t) - Z(t)\|$ for the various types of low-rank approximations $X, Y, Z^{8.1}, Z^{8.2}, Z^{9.1}$ and $Z^{9.2}$.

First, it is clear that over time the approximation quality becomes worse for all the algorithms as the number of significant singular values grow due to the influence of $\exp(t)$ which can't all be captured by a rank-10 approximation. For example, the number of singular values greater than 10^{-2} at $t = 0$ is 11 as opposed to 32 at $t = 1$. Furthermore, we immediately see that $Z^{8.1}$ performs much worse compared to all the other algorithms. This is expected as we start with the (best) rank-10 approximation and from here on, no more subspace information lying outside the current subspace is considered. However, $A(t)$ rotates the relevant subspace over time (due to Q_1 and Q_2) causing it to fail tracking the subspace. This observation matches the similar but far less severe error behavior of $Z^{8.2}$. It can cope with the small initial subspace changes but truncating the update modifications also causes it to accumulate errors very quickly. In contrast, the other algorithms perform similarly to the best rank-10 approximation. As we would expect, $Z^{9.2}$ yields better results than $Z^{9.1}$ for the same reason as mentioned before. Fortunately here, the LRA at a given time point is not based on the approximation at a time point before and therefore is flexible in its subspace orientation at every time point. Also, the DLRA approximation successfully tracks the 10 dominant singular values filtering out the noise or insignificant data. If we change ε by an order of magnitude, the reconstruction errors decrease somewhat proportionally. In Table 2 we see the maximal errors (at $t = 1$) for various choices of ε.

ε	$\|X-A\|$	$\|Y-A\|$	$\|Z^{9.2}-A\|$	$\|Z^{9.1}-A\|$	$\|Z^{8.2}-A\|$	$\|Z^{8.1}-A\|$
10^{-1}	$7.9805e+00$	$8.2200e+00$	$8.1074e+00$	$1.5589e+01$	$8.2220e+00$	$2.0141e+01$
10^{-2}	$1.8144e+00$	$1.9005e+00$	$1.8243e+00$	$1.9854e+00$	$1.9078e+00$	$1.2776e+01$
10^{-3}	$1.8169e-01$	$1.8654e-01$	$1.8170e-01$	$1.8911e-01$	$2.4736e-01$	$1.1431e+01$
10^{-4}	$1.8119e-02$	$1.8947e-02$	$1.8119e-02$	$1.9040e-02$	$2.0249e-01$	$1.1859e+01$
10^{-5}	$1.8139e-03$	$1.8693e-03$	$1.8139e-03$	$1.8753e-03$	$2.2443e-01$	$1.2530e+01$
0	$1.0412e-13$	$3.2185e-14$	$6.8827e-13$	$6.8827e-13$	$1.9874e-01$	$1.1549e+01$

Table 2: Reconstruction errors at $t=1$ for $m=n=100$, $r=10$ and various ε

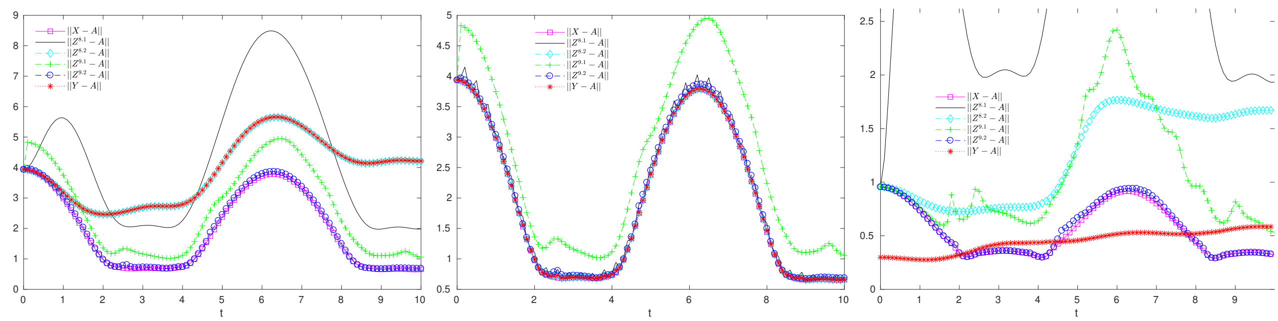

Figure 6: Error plots for different LRA algorithms over time. Left: 5-(D)LRA, Center: 5-(D)LRA with restart, Right: 10-LRA and 20-DLRA. For visibility, we cut the right plot. $Z^{8.1}$ behaves similarly to first plot.

It nicely illustrates the failure of $Z^{8.1}$ for every ε. Furthermore, $Z^{8.2}$ also has trouble readjusting to the subspace rotations. For $\varepsilon = 0$ X, Y, $Z^{9.1}$ and $Z^{9.2}$ are all almost exact (up to rounding errors) as expected. What can already be seen in a small scale, is that $Z^{9.1}$ and $Z^{9.2}$ yield worse results for $\varepsilon = 0$ than Y or X. This is partly due to the nature of the modified Gram-Schmidt procedure which is known to be not as stable as other methods such as SVD (for X) or QR decomposition (for Y), as its quality of creating an orthonormal matrix relies on the condition number σ_1/σ_n as discussed above. Additionally, in its core $Z^{9.1}$ and $Z^{9.2}$ also rely on SVD and therefore propagate possible errors from there even further.

6.3 LRAs of Matrix with Discontinuous X

This examples is also taken from [15]. It shows a weakness of the DLRA due to a discontinuous best-approximation, an assumption we made in the error analysis earlier. We construct matrices $A(t)$ similarly as before, yet with oscillating magnitudes of singular values;

$$A(t) = Q_1(t)(A_1 + \cos(t)A_2)Q_2(t)^T$$

with $A_1, A_2, Q_1(t), Q_2(t)$ given as before and $\varepsilon = 10^{-2}$, but now on the longer interval $[0, 10]$. In figure 6 we see the reconstruction error over time for all the algorithms. The left plot shows the error evolution for rank-5 approximations, the middle plot rank-5 approximations with restart (except of $Z^{9.1}, Z^{9.2}$) and the right plot rank-10 approximations for all the algorithms except Y which is a rank-20 approximation. The oscillating property of all the algorithms can immediately be seen. $\cos(t)$ causes the singular values of $A(t)$ to decrease around $t = \pi$ and increase around $t = 2\pi$. If our chosen approximation rank r is set too small, Y fails to approximate the largest r singular values over the whole interval as they change over time. In particular, if we choose our approximation rank $r = 5$, then Y does not capture the correct behavior of the 2 largest singular values at all times. For example, for $t \approx \pi$ or $t \approx 8$, one of the singular values, that are not approximated becomes larger than the one being approximated. Also note, that $Z^{8.2}$ has almost the same error evolution as Y for the rank-5 case. It is not clear why they behave similarly. For seeing the singular value matgnitudes, we plotted the singular value evolution for all the algorithms and compare them with the exact singular values in figure 7. Again, $Z^{8.1}$ performs very poorly as it can't cope with the subspace rotation which is similar but less severe in $Z^{8.2}$. In contrast, $Z^{9.1}$ almost captures the first few singular values while $Z^{9.2}$ has the best approximation quality and approximates the 5 largest singular values reasonably well despite some artifacts at around $t \approx 7$ for the second and fifth singular value. Clearly, $Z^{9.1}$ and $Z^{9.2}$ benefit in such scenarios from the non-dependence of previous LRAs. The bad error behavior of the other algorithms can be nullified by restarting the respective algorithms around 30 times in the interval as can be seen in the second plot of figure 6. Each algorithm captures the 5 largest singular values at all times. Alternatively, if we choose the approximation rank of Y to 20, it captures the 20 largest singular values at all times as here we approximate all the relevant singular values at all times. For comparison, if we choose the approximation rank of the other algorithms to 10, $Z^{8.1}$ still performs comparably bad to the rank-5 aproximation. However, $Z^{9.1}$ and $Z^{9.2}$ clearly profit from the increased rank as does $Z^{8.2}$. Nevertheless, $Z^{8.2}$ has trouble readjusting the dominant subspace after $t = 2\pi$ (when the influence of A_2 decreases again and the singular values decrease), an observation that can also be seen in Y but to a lesser extend.

6.4 LRAs of Matrices with High Numerical Rank

The preceding examples had a distinguishable gap between the singular values (simulating data versus noise for example). As such a gap might not exist, in this example we investigate the performance of the low-rank approximations of

$$A(t) = Q_1(t)e^t D Q_2(t)^T \in \mathbb{R}^{100\times 100}$$

for $t \in [0, 1]$ where D is a diagonal matrix with descending entries $(d)_{jj} = 2^{-j/10}, j = 1, \ldots 100$. In figure 8 we see the reconstruction errors of the respective rank-10 approximations of these matrices and the errors at $t = 1$ as a function of the approximation rank r from 3 to 97.

In the left plot we see that the errors of all algorithms behave similarly to example 1 just scaled slightly worse as there are more relevant singular values not being captured. The right plot shows that the errors of all the algorithms except of $Z^{8.1}$ decrease at the same rate as the best-approximation error. Again, $Z^{8.1}$ performs much worse because it can't capture the subspace rotations.

6.5 Application Example - Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a common technique in information retrieval which computes a low-rank approximation to a document-term matrix via a truncated SVD ([35]). Updating such a data collection is crucial to many dynamic data bases. In the following, we illustrate how (updated) LRA approaches yield a more efficient alternative to SVD. For that, we take the Classic4 collection containing the datasets CACM, CISI, CRAN, and MED consisting of 3204, 1460, 1398 and 1033 documents, respectively, and in total 5896 terms, yielding a 7095×5896 document-term matrix. This is a standard dataset for testing LSI-based algorithms [35]. We performed the traditional tfidf-weighting-scheme and normalized it to 1 (see e.g. [36]). This matrix has 247.158 non-zero entries. In order to consider an updating problem, we change 10.000 random entries to random (in $[0, 1]$) entries 10 times in a row. $A(t)$ then is the linear interpolation through the update points which yields

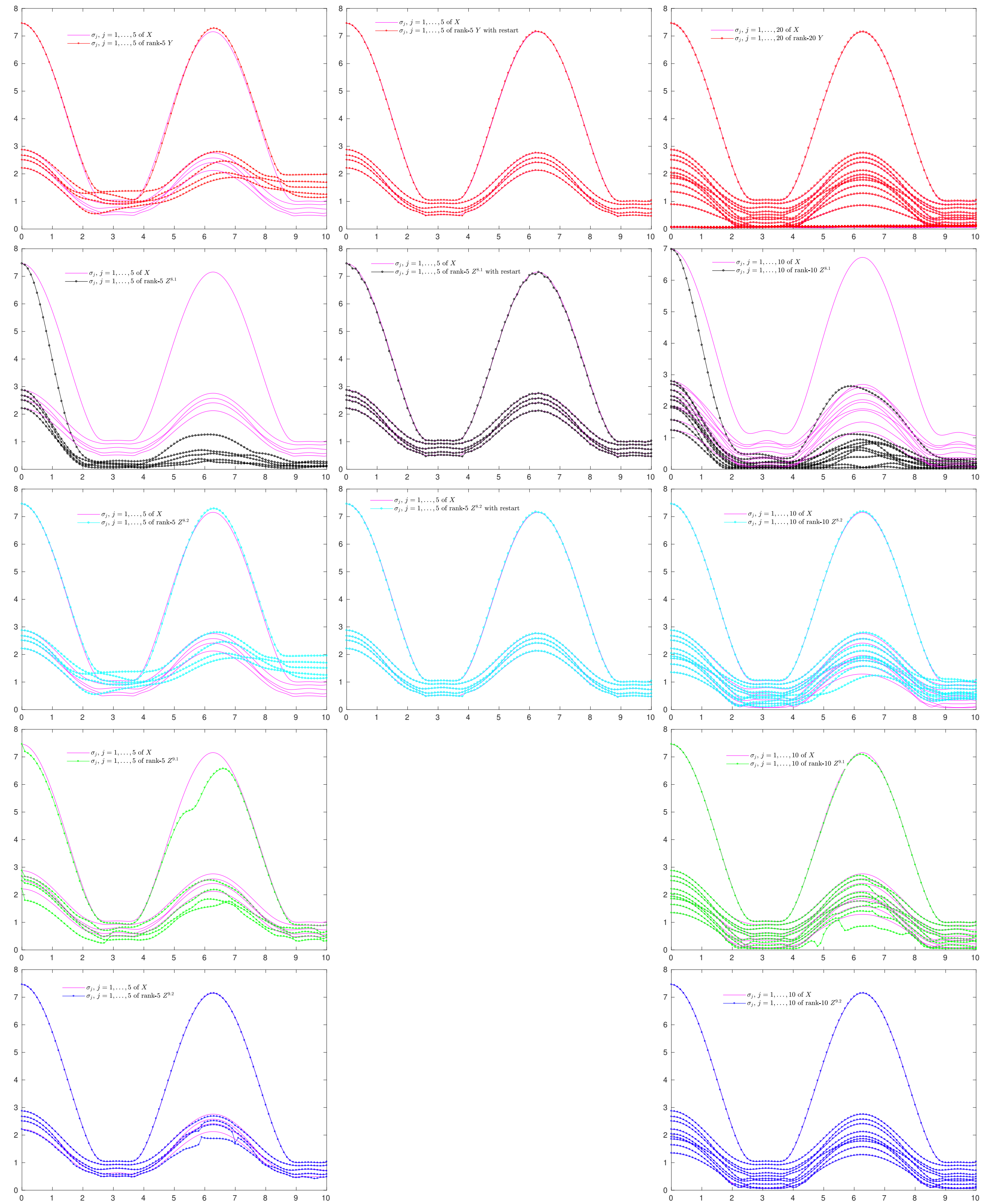

Figure 7: Evolution of magnitudes of exact singular values over time (solid lines) versus its rank-5 (left), rank-5 with restart (except $Z^{9.1}, Z^{9.2}$) (center) and rank-10 (for LRA) and rank-20 (for DLRA) approximations, respectively (dotted lines) (right)

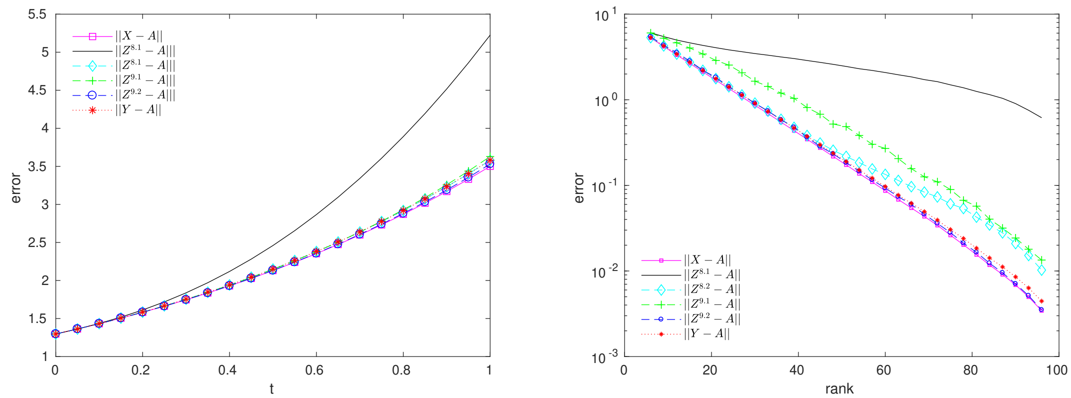

Figure 8: Rank-10 approximation reconstruction errors (left) and reconstruction errors as functions of r at $t = 1$ (right).

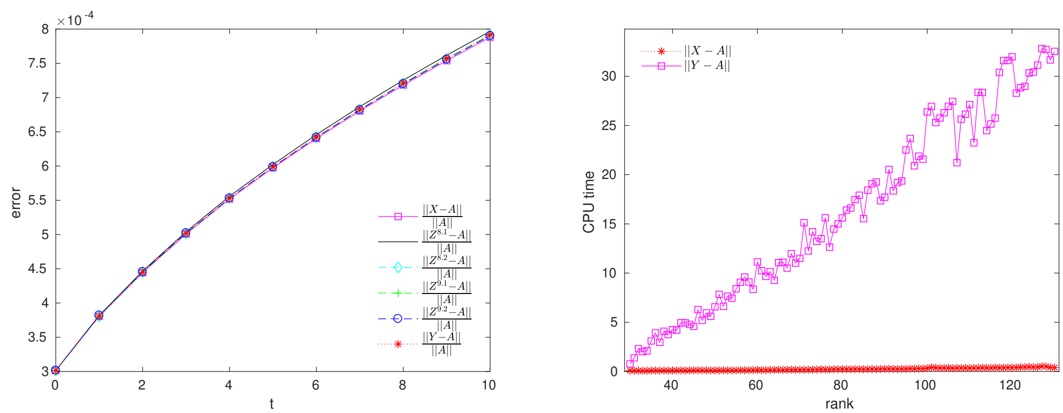

Figure 9: Rank-84 relative approximation reconstruction errors evolution for different LRAs (left) and CPU time in seconds for svds and DLRA as functions of the rank (right).

a constant derivative $\dot{A} = \Delta A$ on the subintervals. We approximate $A(t)$ by rank-$\lceil\sqrt{7095}\rceil$ $(= 84)$. The initial decomposition at $t = 0$ is computed via MATLAB's internal function `svds` (restarted Lanczos bidiagonalization). Figure 9(a) shows the relative error $\frac{\|A-X\|}{\|A\|}$ for the different low-rank approximations X. All the error curves are very similar. Also $Z^{8.1}$ and $Z^{8.2}$ perform well which is due to the sparsity of the updates. In figure 9(b) we compare the running times of `svds` and DLRA as a function of different approximation ranks to effect 10 consecutive updates at every point (as in Figure 9 (a)). As can be seen, the DLRA algorithm very much profits from the sparse ΔA updates. We don't compare it with the other algorithms as they lack an efficient C or Fortran implementation and thus don't show the theoretical computational complexity behavior showed earlier.

7 Conclusion

We reviewed and extended a SVD-based LRA updating technique in section 4 and the DLRA in section 5. The error analysis of the DLRA is well-studied in the literature and it is still an active research area. In particular, the DLRA scheme was extended to tensors and symmetric matrices (see [37] for the most recent publication). The error analysis of updating truncated SVDs (or LRAs) has also been discussed but only for a small class of matrices as stated in Theorem 4.4.5 which states sufficient conditions such that no approximation degradation occurs. Although, this analysis was extended to yield approximative results to column partitioned matrices in [38], it doesn't provide practically useful bounds.

However, the numerical tests showed a very good error behavior for $Z^{9.2}$ in all the test considered. It outperforms $Z^{9.1}$ in all the tests. The same holds for $Z^{8.2}$ versus $Z^{8.1}$. The latter two updating schemes should only be used (in comparion to $Z^{9.2}$) for very sparse updates where we are given the updates in a factorized form or only a few columns are to be modified, such that the computation time savings outweigh the worse error behavior. However, stability is a more critical issue in $Z^{8.2}$ and $Z^{9.2}$ than in $Z^{8.1}$ or $Z^{9.1}$. The truncation (as opposed to ignoring new subspace information) at every update is based on extracting a subspace dimension to yield an inner subspace matrix and its inverse. Similarly to the threshold for deciding when a vector is considered linearly dependent, here we need a threshold for deciding which dimension to extract. This can become troublesome especially for bad conditioned matrices.

Another source of potential instability is the rotations of the subspaces at every update which may erode orthogonality over time. However, in our numerical tests (for not too ill conditioned matrices) we did not encounter any stability issues.

For all the numerical tests, except of the second, the DLRA performs similarly well to $Z^{9.2}$. It does not have stability issues and is the fastest of all the algorithms considered (rougly three times as fast). Its approximation rank choice is not limited to $r \leq \sqrt{m}$ as opposed to the other algorithms which are only fast for these approximation rank magnitudes. Also, sparse updates (not in a factorized form) are much better exploited by the DLRA as the bottleneck here is the matrix multiplication with a sparse matrix. In contrast, in the other algorithms an immediate fill-in after the new orthonormal vector is found eliminates the sparse structure. Furthermore, the DLRA can be used in conjunction with matrix differential equations.

Lastly, the DLRA only requires the storage of an $m \times r$, an upper triangular $r \times r$ and an $n \times r$ matrix as opposed to the other algorithms which require on top of these matrices two inner $r \times r$ subspace matrices with their inverses.

Deutsche Zusammenfassung

In dieser Arbeit wird das Problem "Updates von Niedrigrang-Matrix-Approximationen" betrachtet:
Gegeben eine Niedrigrang-Approximation (NRA) Z_0 einer Matrix A_0, wie erhalten wir möglichst effizient und genau eine NRA der upgedateten Matrix $A_1 = A_0 + \Delta A$ oder $A_1 = [A_0, \Delta A]$ basierend auf der alten Approximation Z_0?
Da die beste NRA zu einem festen Rang gegeben ist durch die abgeschnittene Singulärwertzerlegung (SWZ), ist eine natürliche Herangehensweise an das Update-Problem einer NRA mittels Updates einer SWZ. Abgesehen von Algorithmen zum Updaten von SWZen, gibt es allerdings auch noch andere Verfahren zum Updaten von NRAen. Diese Arbeit versucht, einen Teilüberblick über verschiedene Algorithmen, die im Bereich der NRA existieren, zu geben und diese Algorithmen theoretisch und numerisch zu vergleichen. Es wird eine Klasse von SWZ-basierenden Verfahren aufgearbeitet und deren vorteilhafte Nutzung für NRAen erläutert und mit der Methode der Dynamischen NRA in numerischen Tests verglichen.
Im ersten Abschnitt wird eine Einleitung in das Thema und ein grober Literaturüberblick von themenverwandten und anwendungsbezogenen Arbeiten im Bereich der NRA gegeben.
Abschnitt zwei und drei klären und definieren die Nomenklatur und Konventionen beziehungsweise formalisieren das Problem mathematisch.
In Abschnitt vier beschäftigen wir uns mit den SWZ-basierenden Verfahren und der Rechtfertigung, NRAen basierend auf alten NRAen zu updaten. Wir präsentieren einige bekannte Algorithmen und erweitern alsdann ein Verfahren zum Updaten von SWZ, welches bisher nur für das Hinzufügen von Spalten beschrieben wurde, zu dem Fall, dass Spalten verändert werden können. Dabei gehen wir gesondert auf die Nützlichkeit dieser SWZ-Updates für NRAen ein.
Abschnitt fünf handelt von dem Gebiet der Dynamischen NRA. Zunächst wird die Problemstellung und damit einhergehend Differentialgleichungen zur Lösung dieses Problems formuliert. Es werden Fehlerschranken analysiert und zuletzt ein spezieller Integrator zur Lösung dieser Differentialgleichungen vorgestellt, welcher einen praktischen Algorithmus liefert.
Die Arbeit schließt mit numerischen Tests und Vergleichen der vorgestellten Algorithmen und einem Anwendungsbeispiel im Bereich des Latent Semantic Indexing ab.

References

[1] S. Friedland, V. Mehrmann, A. Miedlar, and M. Nkengla. Fast low rank approximations of matrices and tensors. *Electron. J. Linear Algebra*, 22:1031–1048, 2011.

[2] Lars Eldén. Numerical linear algebra in data mining. *Acta Numer.*, 15:327–384, 2006.

[3] David Skillicorn. *Understanding complex datasets*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC, Boca Raton, FL, 2007. Data mining with matrix decompositions.

[4] Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: a probabilistic analysis. *J. Comput. System Sci.*, 61(2):217–235, 2000. Special issue on the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (Seattle, WA, 1998).

[5] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philos. Trans. Roy. Soc. A*, 374(2065):20150202, 16, 2016.

[6] Haesun Park and Lars Eldén. Matrix rank reduction for data analysis and feature extraction. In *Handbook of parallel computing and statistics*, volume 184 of *Stat. Textb. Monogr.*, pages 415–447. Chapman & Hall/CRC, Boca Raton, FL, 2006.

[7] Joonseok Lee, Seungyeon Kim, Guy Lebanon, Yoram Singer, and Samy Bengio. LLORMA: local low-rank matrix approximation. *J. Mach. Learn. Res.*, 17:Paper No. 15, 24, 2016.

[8] Robert E. McCulloch. Machine learning: a probabilistic perspective [book review, MIT Press, Cambridge MA, 2012]. *J. Amer. Statist. Assoc.*, 110(512):1822, 2015.

[9] Xuefeng Duan, Jianchao Bai, Maojun Zhang, and Xinjun Zhang. On the generalized low rank approximation of the correlation matrices arising in the asset portfolio. *Linear Algebra Appl.*, 461:1–17, 2014.

[10] Ivan Markovsky. Structured low-rank approximation and its applications. *Automatica J. IFAC*, 44(4):891–909, 2008.

[11] James R. Bunch and Christopher P. Nielsen. Updating the singular value decomposition. *Numer. Math.*, 31(2):111–129, 1978/79.

[12] Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear algebra and its applications*, 415(1):20–30, 2006.

[13] Hongyuan Zha and Horst D Simon. On updating problems in latent semantic indexing. *SIAM Journal on Scientific Computing*, 21(2):782–791, 1999.

[14] Fahrettin Horasan, Hasan Erbay, Fatih Varçın, and Emre Deniz. Alternate low-rank matrix approximation in latent semantic analysis. *Scientific Programming*, 2019, 2019.

[15] Othmar Koch and Christian Lubich. Dynamical low-rank approximation. *SIAM Journal on Matrix Analysis and Applications*, 29(2):434–454, 2007.

[16] Charles F Van Loan. Matrix computations (johns hopkins studies in mathematical sciences), 1996.

[17] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[18] Rasmus Munk Larsen. Lanczos bidiagonalization with partial reorthogonalization. *DAIMI Report Series*, (537), 1998.

[19] Horst D Simon and Hongyuan Zha. Low-rank matrix approximation using the lanczos bidiagonalization process with applications. *SIAM Journal on Scientific Computing*, 21(6):2257–2274, 2000.

[20] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

[21] Ming Gu and Stanley C Eisenstat. A stable and fast algorithm for updating the singular value decomposition, 1994.

[22] I.M. James, N.J. Hitchin, and Cambridge University Press. *The Topology of Stiefel Manifolds*. Cambridge books online. Cambridge University Press, 1976.

[23] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[24] Christian Lubich and Ivan V Oseledets. A projector-splitting integrator for dynamical low-rank approximation. *BIT Numerical Mathematics*, 54(1):171–188, 2014.

[25] Emil Kieri, Christian Lubich, and Hanna Walach. Discretized dynamical low-rank approximation in the presence of small singular values. *SIAM Journal on Numerical Analysis*, 54(2):1020–1038, 2016.

[26] John R Rice. Experiments on gram-schmidt orthogonalization. *Mathematics of Computation*, 20(94):325–328, 1966.

[27] James H Wilkinson. Modern error analysis. *SIAM review*, 13(4):548–568, 1971.

[28] Åke Björck. Solving linear least squares problems by gram-schmidt orthogonalization. *BIT Numerical Mathematics*, 7(1):1–21, 1967.

[29] Nabih N Abdelmalek. Round off error analysis for gram-schmidt method and solution of linear least squares problems. *BIT Numerical Mathematics*, 11(4):345–367, 1971.

[30] Heinz Rutishauser and Friedrich L Bauer. *Handbook for Automatic Computation: Description of Algol 60*. Springer, 2013.

[31] James W Daniel, Walter Bill Gragg, Linda Kaufman, and Gilbert W Stewart. Reorthogonalization and stable algorithms for updating the gram-schmidt qr factorization. *Mathematics of Computation*, 30(136):772–795, 1976.

[32] William Gragg and Lother Reichel. Fortran subroutines for updating the qr decomposition. Technical report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 1988.

[33] Beresford N Parlett. *The symmetric eigenvalue problem*, volume 20. siam, 1998.

[34] Luc Giraud and Julien Langou. When modified gram–schmidt generates a well-conditioned set of vectors. *IMA Journal of Numerical Analysis*, 22(4):521–528, 2002.

[35] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[36] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[37] Gianluca Ceruti and Christian Lubich. Time integration of symmetric and anti-symmetric low-rank matrices and tucker tensors. *arXiv preprint arXiv:1906.01369*, 2019.

[38] Zhenyue Zhang and Hongyuan Zha. Structure and perturbation analysis of truncated svds for column-partitioned matrices. *SIAM Journal on Matrix Analysis and Applications*, 22(4):1245–1262, 2001.

Appendices

Lemma 4.6.1. *Consider the compact SVD of the rank-r matrix* $A = USV^T$ *with* $U \in \mathbb{R}^{m\times r}$*,* $S \in \mathbb{R}^{r\times r}$ *and* $V \in \mathbb{R}^{n\times r}$*. For the i-th column of A,* A_i*, and the i-th row of V,* v_i*, then it holds that* $A_i \notin$ *Span*$(A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_n)$ *if and only if* $\langle v_i, v_j\rangle = \delta_{ij} \ \ \forall j = 1, \ldots, n$.

Proof. $(\Rightarrow)$ Consider $k = [k_1, \ldots, k_n]^T \in \ker(A)$ for which holds

$$0 = Ak = \begin{bmatrix} a_{11}k_1 + \cdots + a_{1n}k_n \\ \vdots \\ a_{m1}k_1 + \cdots + a_{mn}k_n \end{bmatrix} = k_1A_1 + \cdots + k_iA_i + \cdots + k_nA_n. \tag{74}$$

Furthermore, the following implications hold:

$$\begin{aligned} &A_i \notin \operatorname{Span}(A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_n) \\ \Rightarrow &A_i \neq \sum_{j\neq i} \lambda_j A_j \ \ \forall \lambda_j \in \mathbb{R}, j = 1, \ldots, i-1, i+1, \ldots, n \\ \Rightarrow &\left(0 = \lambda A_i + \sum_{j\neq i}\lambda_j A_j, \ \ \lambda \in \mathbb{R} \ \ \Rightarrow \lambda = 0\right). \end{aligned}$$

Combining that with (74) we obtain that if A_i is not in the span of the other columns of A then k_i is zero for any arbitrary kernel element k. Thus, it also holds for any basis vector of any basis of the kernel of A. Therefore, if we were to complete V such that it becomes an orthonormal basis of $\mathbb{R}^n$ the vectors to be appended all have a zero entry in the i-th entry. This means that the i-th row of V already was of norm 1 and because the rows of the completed V are orthonormal to all other rows, the i-th row of the compact V must already be orthogonal to all others.

$(\Leftarrow)$ Now suppose that $A_i \in \operatorname{Span}(A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_n)$, then this implies that

$$\exists \lambda_1, \ldots, \lambda_{i-1}, \lambda_{i+1}, \ldots, \lambda_n \in \mathbb{R} \ \ : \ \ 0 = -A_i + \sum_{j\neq i} \lambda_j A_j.$$

Choosing $k = [\lambda_1, \ldots, \lambda_{i-1}, -1, \lambda_{i+1}, \ldots, \lambda_n]^T$ yields that $k \in \ker(A)$ with the i-th component non-zero. This means that for every basis of the kernel of A there must exist a basis vector with the i-th component non-zero. Again, if we were to complete V to be an orthonormal basis of $\mathbb{R}^n$ it would append an element to the i-th row of V that is non zero which means that v_i was not of norm 1. □

Lemma 4.6.2. *Let* $L = \begin{bmatrix} W & y \\ w & x \end{bmatrix} \in \mathbb{R}^{r+1\times r+1}$ *be an orthonormal matrix, where* $W \in \mathbb{R}^{r\times r}$*. Then the r-th leading principle submatrix W is invertible if and only if* $x \neq 0$ *and in this case the inverse is given by*

$$W^{-1} = W^T - \frac{w^T y^T}{x}.$$

Proof. The statement follows immediately from the next Lemma as it is a special case. □

Lemma 4.6.3. *Let $L \in \mathbb{R}^{r+1\times r+1}$ be regular. Then we can always delete/remove the last column and at least one row such that the remaining submatrix is regular. The submatrix obtained by deleting the last column and the i-th row is regular if and only if $L_{i,r+1} \neq 0 \neq (L^{-1})_{r+1,i}$. The inverse is given by*

$$Z - \frac{mn}{z}$$

where $Z \in \mathbb{R}^{r\times r}$ is the submatrix of L^{-1} obtained by deleting the i-th column and the last row, $m \in \mathbb{R}^{r\times 1}$ is the i-th column of L^{-1} with the last entry deleted and $n \in \mathbb{R}^{1\times r}$ is the last row of L^{-1} with the i-th entry deleted and $z := (L^{-1})_{r+1,r+1}$.

Proof. When deleting the last column, we are left with a matrix with $r+1$ rows that is of rank r. This means that at least one row can be expressed as a linear combination of the others. Leaving out one of those rows does not decrease the rank and thus the remaining r rows form a basis and the remaining matrix is regular. We can always find such a row using the Woodburry criterion as follows. We will provide a criterion for which the r-th leading principal submatrix W is regular such that we simply have to delete the last column and last row rather than an arbitrary row. However, by rearranging the rows we obtain the results for an arbitrary submatrix. The idea is to transform our matrix to

$$L := \begin{bmatrix} W & y \\ w & x \end{bmatrix} \to \begin{bmatrix} W & 0_{r\times 1} \\ 0_{1\times r} & x \end{bmatrix},$$

where $y \in \mathbb{R}^{r\times 1}$, $w \in \mathbb{R}^{1\times r}$ and x is a scalar, using a rank-2 modification. The regularity of the latter matrix is equivalent to the regularity of the r-th leading principal submatrix W. We just perform a rank-2 modification using

$$U = -\begin{bmatrix} y & 0_{r\times 1} \\ 0 & 1 \end{bmatrix}, \quad V = \begin{bmatrix} 0_{1\times r} & 1 \\ w & 0 \end{bmatrix}$$

to yield

$$(L - UV)^{-1} = L^{-1} + L^{-1}U(I_2 - VL^{-1}U)^{-1}VL^{-1}$$

iff $I_2 - VL^{-1}U$ is invertible by Woodbury's matrix identity. Denoting the inverse as

$$L^{-1} = \begin{bmatrix} Z & m \\ n & z \end{bmatrix},$$

we obtain the outer factors

$$L^{-1}U = -\begin{bmatrix} Z & m \\ n & z \end{bmatrix} \begin{bmatrix} y & 0_{r\times 1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Zy & m \\ ny & z \end{bmatrix} = \begin{bmatrix} -mx & m \\ ny & z \end{bmatrix}$$

$$VL^{-1} = \begin{bmatrix} 0_{1\times r} & 1 \\ w & 0 \end{bmatrix} \begin{bmatrix} Z & m \\ n & z \end{bmatrix} = \begin{bmatrix} n & z \\ wZ & wm \end{bmatrix} = \begin{bmatrix} n & z \\ -xn & wm \end{bmatrix}$$

because

$$I = L^{-1}L = \begin{bmatrix} Z & m \\ n & z \end{bmatrix} \begin{bmatrix} W & y \\ w & x \end{bmatrix} = \begin{bmatrix} ZW + mw & Zy + mx \\ nW + zw & ny + zx \end{bmatrix} \Rightarrow Zy = -mx,$$
$$I = LL^{-1} = \begin{bmatrix} W & y \\ w & x \end{bmatrix} \begin{bmatrix} Z & m \\ n & z \end{bmatrix} = \begin{bmatrix} WZ + yn & Wm + yx \\ wZ + xn & wm + xz \end{bmatrix} \Rightarrow wZ = -xn,$$

and thereby

$$I_2 - VL^{-1}U = I_2 - \begin{bmatrix} 0_{1\times r} & 1 \\ w & 0 \end{bmatrix} \begin{bmatrix} -mx & m \\ ny & z \end{bmatrix} = \begin{bmatrix} 1 - ny & -z \\ wmx & 1 - wm \end{bmatrix} = \begin{bmatrix} xz & -z \\ x(xz - 1) & xz \end{bmatrix}$$

as $ny + xz = 1 = wm + xz$. The matrix is invertible iff

$$0 \neq \det(I_2 - VL^{-1}U) = x^2z^2 + xz(1 - xz) = xz \quad \Leftrightarrow \quad x \neq 0 \wedge z \neq 0.$$

In the regular case, the inverse is given as

$$(I_2 - VL^{-1}U)^{-1} = \frac{1}{xz} \begin{bmatrix} xz & z \\ x(1 - xz) & xz \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{x} \\ \frac{xz-1}{z} & 1 \end{bmatrix}$$

and hence the inverse of W is given by

$$Z + \begin{bmatrix} -mx & m \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{x} \\ \frac{xz-1}{z} & 1 \end{bmatrix} \begin{bmatrix} n \\ -xn \end{bmatrix} = Z + \begin{bmatrix} -mx + m\frac{xz-1}{z} & 0 \end{bmatrix} \begin{bmatrix} n \\ -xn \end{bmatrix} = Z - \frac{mn}{z}.$$

□

Lemma 5.1.1. *The manifold of rank-r matrices $M_r = \{A \in \mathbb{R}^{m\times n} | rk(A) = r\}$ is a submanifold of $\mathbb{R}^{m\times n}$ with dimension $(m + n)r - r^2$.*

Proof. First, we show the claim for matrices of the form

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix} \in \mathbb{R}^{m\times n}$$

where $B \in \mathbb{R}^{r\times r}$ is regular. Since the set of invertible matrices is an open set, the set of matrices of that form, denoted by Z, is a submanifold of $\mathbb{R}^{m\times n}$. Postmultiplying such a matrix by the regular matrix

$$\begin{bmatrix} I & -B^{-1}C \\ 0 & I \end{bmatrix}$$

yields

$$\begin{bmatrix} B & 0 \\ D & -DB^{-1}C + E \end{bmatrix}.$$

A is of rank r if and only if this matrix is which is only the case if $-DB^{-1}C + E = 0$ as B is of full rank. Now, we can define a map $f : Z \to \mathbb{R}^{m-r\times n-r}$ with

$$f(A) = f\left(\begin{bmatrix} B & C \\ D & E \end{bmatrix}\right) = -DB^{-1}C + E$$

which is a smooth map. We need to check whether this map is a submersion. As the image $\mathbb{R}^{m-r\times n-r}$ is a vector space the tangent space coincides with the space. Let $X \in \mathbb{R}^{m-r\times n-r}$ and consider the curve

$$\gamma(t) = \begin{bmatrix} B & C \\ D & E + tX \end{bmatrix}$$

which passes through any matrix $A \in Z$ (for $t = 0$). The derivative of $f \circ \gamma$ at 0 is X which is also equal to

$$df_A\left(\begin{bmatrix} 0 & 0 \\ 0 & X \end{bmatrix}\right)$$

such that for every point A we have shown that there exists a preimage tangent vector at A that is mapped by df to X which means that f is a submersion. Therefore, $f^{-1}(0)$ is a smooth submanifold of $\mathbb{R}^{m\times n}$. The dimension of $f^{-1}(0)$ is $mn - (m-r)(n-r)$. We have yet only shown that matrices of rank r contained in Z form a smooth submanifold. However, we can always transform any rank-r matrix to the form above by linear isomorphisms and therefore the statement also holds for general rank-r matrices if we consider the collection of all such composed charts.

$\square$

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Masterarbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht.
Außerdem erkläre ich, dass diese Arbeit weder komplett noch zu Teilen Bestandteil einer anderen Arbeit war.

....

Ort, Datum Unterschrift des Verfassers