A Quadratically Convergent Algorithm for Structured Low-Rank Approximation

Éric Schost[1] and Pierre-Jean Spaenlehauer[1,2,3]

[1]ORCCA and CS Department, Western University, London, ON Canada
[2]Max Planck Institute for Mathematics, Bonn, Germany
[3]Inria, CNRS, Université de Lorraine, France

Abstract

Structured Low-Rank Approximation is a problem arising in a wide range of applications in Numerical Analysis and Engineering Sciences. Given an input matrix M, the goal is to compute a matrix M' of given rank r in a linear or affine subspace E of matrices (usually encoding a specific structure) such that the Frobenius distance $\|M - M'\|$ is small. We propose a Newton-like iteration for solving this problem, whose main feature is that it converges locally quadratically to such a matrix under mild transversality assumptions between the manifold of matrices of rank r and the linear/affine subspace E. We also show that the distance between the limit of the iteration and the optimal solution of the problem is quadratic in the distance between the input matrix and the manifold of rank r matrices in E. To illustrate the applicability of this algorithm, we propose a `Maple` implementation and give experimental results for several applicative problems that can be modeled by Structured Low-Rank Approximation: univariate approximate GCDs (Sylvester matrices), low-rank Matrix completion (coordinate spaces) and denoising procedures (Hankel matrices). Experimental results give evidence that this all-purpose algorithm is competitive with state-of-the-art numerical methods dedicated to these problems.

Keywords: Structured low-rank approximation, Newton iteration, quadratic convergence, Approximate GCD, Matrix completion.

AMS classification: 65B99 (Acceleration of Convergence), 65Y20 (Complexity and performance of numerical algorithms), 15A83 (Matrix completion problems).

1 Introduction

1.1 Motivation and problem statement

In a wide range of applications (data fitting, symbolic-numeric computations, signal processing, system and control theory,...), the problem arises of computing low rank approximations of matrices under linear constraints; this central question is known as *Structured Low-Rank Approximation* (abbreviated SLRA). Quoting Markovsky [33]: *behind every linear data modeling problem there is a (hidden) low-rank approximation problem: the model imposes relations on the data which render a matrix constructed from exact data rank deficient.* We refer the reader to [33] for an overview of the vast extent of fields where SLRA arises in a natural way.

Let $\mathcal{M}_{p,q}(\mathbb{R})$ denote the space of $p \times q$ matrices with real entries, endowed with the inner product
$$\langle M_1, M_2 \rangle = \mathrm{trace}(M_1 \cdot M_2^{\mathsf{T}});$$
this vector space inherits the Frobenius norm $\|M\| = \sqrt{\langle M, M \rangle}$ deduced from this inner product. For $r \in \mathbb{N}$, let further $\mathcal{D}_r \subset \mathcal{M}_{p,q}(\mathbb{R})$ denote the set of matrices of size $p \times q$ and of rank equal to r; this is both a semi-algebraic set and an analytic manifold in $\mathcal{M}_{p,q}(\mathbb{R})$ of codimension $(p-r)(q-r)$ [6, Proposition 1.1]. The Structured Low-Rank Approximation Problem can be stated as follows:

Problem 1 - Structured Low-Rank Approximation (SLRA). Let $E \subset \mathcal{M}_{p,q}(\mathbb{R})$ be an affine subspace of $\mathcal{M}_{p,q}(\mathbb{R})$, let $M \in E$ be a matrix and let $r \in \mathbb{N}$ be a integer. Find a matrix $M^\star \in E \cap \mathcal{D}_r$ such that $\|M - M^\star\|$ is "small".

The problem is not entirely specified yet, since we have to state what "small" means. Actually, several variants of this problem can be found in the litterature (for instance this problem can be stated similarly for other norms). One way to approach SLRA is as an optimization problem, by looking for the matrix $M^\star$ in $E \cap \mathcal{D}_r$ which minimizes $\|M - M^\star\|$, *i.e.* such that $\|M - M^\star\| = \mathrm{dist}(M, E \cap \mathcal{D}_r)$, where $\mathrm{dist}(M, S)$ denotes the distance of M to the set S. Let us denote by $\Pi_{E \cap \mathcal{D}_r}$ the orthogonal projection $\Pi_{E \cap \mathcal{D}_r}(M) = \mathrm{argmin}_{M^\star \in E \cap \mathcal{D}}(\|M - M^\star\|)$, which is well-defined and continuous in a neighborhood of $E \cap \mathcal{D}_r$. Then, the optimization form of the SLRA problem precisely amounts to computing $\Pi_{E \cap \mathcal{D}_r}(M)$.

On another hand, it may also be sufficient to compute a matrix M' whose distance to the optimal solution is small with respect to $\mathrm{dist}(M, E \cap \mathcal{D}_r)$. This is a mild relaxation of the optimization form of the problem, and it seems to be sufficient for many applications. Indeed, the SLRA problem often arises in situations where an exact structured matrix has been perturbed by some noise, and SLRA can be viewed as a denoising procedure; in this context, the original matrix may not be the optimal solution of the underlying SLRA problem and therefore computing $M' \in E \cap \mathcal{D}_r$ such that $\|M - M'\|$ is small may be sufficient if the error is controlled.

1.2 Main results

We propose an iterative algorithm, called `NewtonSLRA`, solving the second form of the SLRA problem with proven quadratic convergence, under mild transversality conditions on E and $\mathcal{D}_r$. Given an input matrix M in E, the output of the algorithm is a matrix M' in $E \cap \mathcal{D}_r$ which is a good approximation of the optimal $\Pi_{E \cap \mathcal{D}_r}(M)$, in the sense that the distance $\|\Pi_{E \cap \mathcal{D}_r}(M) - M'\|$ is quadratic in $\text{dist}(M, E \cap \mathcal{D}_r) = \|\Pi_{E \cap \mathcal{D}_r}(M) - M\|$.

An iteration of the algorithm relies mainly on a Singular Value Decomposition, plus a few further linear algebra operations. It is not our goal in this paper to analyze the numerical accuracy of our algorithm in floating-point arithmetic. For this reason, we would like to state the complexity analysis in terms of arithmetic operations $+, -, \times, \div$ on real numbers. We can achieve this for all operations except the Singular Value Decomposition, which is an iterative process in itself (see [24, Ch. 45-46]). As a result, in our cost analysis, we isolate the cost induced by the Singular Value Decomposition, and count all other arithmetic operations at unit cost.

In all that follows, if M is a matrix in $\mathcal{M}_{p,q}(\mathbb{R})$, we let $B_\rho(M) \subset \mathcal{M}_{p,q}(\mathbb{R})$ denote the open ball centered at M and of radius ρ.

Theorem. *The algorithm* `NewtonSLRA` *computes a function φ defined on an open neighborhood $U \supset \mathcal{D}_r$, and with codomain E, verifying the following property:*

Let ζ be in $E \cap \mathcal{D}_r$ such that E and $\mathcal{D}_r$ intersect transversally at ζ. There exist $\nu, \gamma, \gamma' > 0$ such that, for all M_0 in $E \cap B_\nu(\zeta)$, the sequence (M_i) given by $M_{i+1} = \varphi(M_i)$ is well-defined and converges towards a matrix $M_\infty \in E \cap \mathcal{D}_r$ and

- $\|M_{i+1} - M_\infty\| \leq \gamma \|M_i - M_\infty\|^2$ *for all $i \geq 0$;*

- $\|\Pi_{E \cap \mathcal{D}_r}(M_0) - M_\infty\| \leq \gamma' \, \text{dist}(M_0, E \cap \mathcal{D}_r)^2$.

Moreover, the function φ can be computed by means of a Singular Value Decomposition of the input matrix M, plus $O(\min(pqd(p-r)(q-r) + pqr, pqr(pq-d)(p+q-r)))$ arithmetic operations, with $d = \dim(E)$.

To the best of our knowledge, this is the first algorithm for SLRA with proven local quadratic convergence. We can actually give explicit estimates of the constants γ and γ', which depend on the incidence angle between $\mathcal{D}_r$ and E around ζ and on the second derivatives of the projection operators $\Pi_{\mathcal{D}_r}$ and $\Pi_{E \cap \mathcal{D}_r}$.

Algorithm `NewtonSLRA` is a variant of a lift-and-project technique which was introduced by Cadzow [7]. However, instead of projecting orthogonally from $\mathcal{D}_r$ back to E, we choose a direction of projection which is tangent to the determinantal variety $\mathcal{D}_r$, in the spirit of Newton's method. The algorithm relies on the Singular Value Decomposition in order to achieve the "lifting" step.

Let us denote by Φ the limit mapping, given by $\Phi(M) = M_\infty$, for M_∞ as in the above theorem. The following theorem states that Φ behaves to the first order as the operator $\Pi_{E \cap \mathcal{D}_r}$, which returns the optimal solution of the SLRA problem. In what follows, for ζ in $E \cap \mathcal{D}_r$, we denote by $T_\zeta(E \cap \mathcal{D}_r)^0$ the tangent vector space to $E \cap \mathcal{D}_r$ at ζ (which is well-defined as soon as $\mathcal{D}_r$ and E intersect transversally at ζ).

3

Theorem. *The limit operator Φ is well-defined and continuous around any point $\zeta \in E \cap \mathcal{D}_r$ such that $\mathcal{D}_r$ and E intersect transversally at ζ, and Φ satisfies $\Phi(\zeta) = \Pi_{E \cap \mathcal{D}_r}(\zeta) = \zeta$. Moreover, Φ is differentiable at ζ and*

$$D\Phi(\zeta) = D\Pi_{E \cap \mathcal{D}_r}(\zeta) = \Pi_{T_\zeta(E \cap \mathcal{D}_r)^0}.$$

These results actually hold more generally than in the SLRA context: the manifold $\mathcal{D}_r$ of rank r matrices could be replaced by any manifold $\mathcal{V}$ such that the projection $\Pi_\mathcal{V}$ is of class C^2 and can be computed efficiently, and such that for any point $v \in \mathcal{V}$ a basis of the normal space $N_v\mathcal{V}$ can be obtained. In the context of SLRA where $\mathcal{V} = \mathcal{D}_r$, the projection on $\mathcal{D}_r$ and a description of the normal space can be obtained from the Singular Value Decomposition.

Our algorithm `NewtonSLRA` is suitable for practical computations: to illustrate its efficiency, we have implementated it in `Maple` and applied it in different contexts:

- univariate approximate GCDs;

- low-rank matrix completion;

- low-rank approximation of Hankel matrices.

For all of these contexts, we provide experimental results and compare it with state-of-the-art techniques.

1.3 Related works

Structured low-rank approximation and its applications have led to huge amounts of work during the last decades, from different perspectives. One of the first iterative methods for computing SLRA is due to Cadzow and is based on alternating projections [7]; it has a linear rate of convergence [31].

A different approach is based on optimization techniques to approximate the nearest low-rank matrix. The difficulty in this setting lies in the implicit description of the problem and of the feasible set. It has led to a large family of algorithms, see *e.g.* [12] and references therein.

Several particular cases of SLRA problems have also been deeply investigated, and specific algorithms have been proposed for these special cases. For instance, the *matrix completion* problem asks for unknown values of a matrix in order to satisfy a rank condition [43]. In particular, this computational question appears in machine learning or in compressed sensing problems, and convex optimization techniques have been developped in this context, see *e.g.* [10, 8, 37]. Techniques of alternating minimizations for SLRA, leading to linear (also called *geometric*) convergence have been introduced in [25].

Structured Low-Rank Approximation is also underlying several problems in hybrid symbolic-numerical computations. The notion of *quasi-GCD* introduced in [41] shows how to compute GCDs by using floating-point computations and has led to developments in the last decades of different notions of approximate GCD. In particular, degree conditions on the approximate univariate or multivariate GCD can be expressed by a rank condition in a linear

space of matrices (see *e.g* [29, 30, 27, 46, 32, 44]). Certified techniques [20] and geometric approaches [35] (by perturbing the roots instead of perturbing the coefficients) have also been developed.

Approximate multivariate factorization also involves a linear space of matrices (Ruppert matrices) and can be modeled by SLRA [22, 26]. The relation between the ranks of Ruppert matrices and the reducibility of multivariate polynomials follows from a criterion introduced in [40].

Denoising procedures in Signal Processing often involve low rank approximation in the linear space of Hankel matrices. Dedicated techniques for this task have been designed and analyzed in [36].

Another line of work motivated by the matrix completion problem has been initiated in [1] by designing a Newton-like method for computing the optima of functions defined over Riemannian manifolds. Other optimization techniques such as the *Structured Total Least Squares* approach have also been applied to the SLRA problem and can be applied to different matrix norms [39, 36, 28].

In [18], the authors show several algebraic and geometric properties of the critical points of the Euclidean distance function on an algebraic variety. For instance, a connection is exhibited between the number of complex critical points and the degrees of the Chern classes of the variety. Algebraic methods for solving the SLRA optimization problem from this viewpoint have been investigated in [34], with a special focus on generic linear spaces E and on SLRA problems occurring in approximate GCD and symmetric tensor decompositions.

1.4   Organization of the paper

Section 2 introduces the main tools that will be used throughout this paper. In Section 3, we describe the algorithm `NewtonSLRA`, we prove its correctness and derive the complexity of each of its iteration. The main result is the proof of the local quadratic rate of convergence of `NewtonSLRA` in Section 4. Finally, we show the experimental behavior of `NewtonSLRA` in Section 5 and apply it to three different applicative contexts: univariate approximate GCD, low-rank matrix completion, and low-rank approximation of Hankel matrices.

2   Preliminaries

Our algorithm combines features of the alternating projections algorithm and of Newton's method for solving underdetermined systems. In this section, we introduce basic ingredients used in those previous algorithms that will be reused here, and present the basics of alternating projections techniques and Newton iteration for comparison purposes.

2.1   Notations and basic facts

Throughout this paper, if E is an affine space, E^0 denotes the underlying vector space, so that $E = x + E^0$, for any x in E. In particular, if $\mathcal{V}$ is a manifold or an algebraic set lying

in a Euclidean space, and x is in $\mathcal{V}$, then $T_x\mathcal{V}$ denotes the affine space that is tangent to $\mathcal{V}$ at x and the underlying vector space is denoted by $T_x\mathcal{V}^0$; thus $T_x\mathcal{V} = x + T_x\mathcal{V}^0$. Similarly, the normal space $N_x\mathcal{V}$ to $\mathcal{V}$ at x is given by $N_x\mathcal{V} = x + N_x\mathcal{V}^0$, where $N_x\mathcal{V}^0$ is the orthogonal complement of $T_x\mathcal{V}^0$.

Recall next our definition of the projection operator $\Pi_\mathcal{V}$ on the manifold $\mathcal{V}$. For a proof of the following properties, see [31, Lemma 4].

Lemma 2.1. *Let $\mathbb{E}$ be a Euclidean space and let $\mathcal{V} \subset \mathbb{E}$ be a manifold of class C^k with $k \geq 2$. There exists an open neighborhood U of $\mathcal{V}$ such that the projection*

$$\Pi_\mathcal{V}(x) = \mathrm{argmin}\{\|y - x\| : y \in \mathcal{V}\}$$

is well-defined on U. Moreover, $\Pi_\mathcal{V}$ is of class C^{k-1} around any point $\zeta \in \mathcal{V}$ and

$$\forall \zeta \in \mathcal{V}, D\Pi_\mathcal{V}(\zeta) = \Pi_{T_{\Pi_\mathcal{V}(\zeta)}\mathcal{V}^0}.$$

We will need further results regarding the projection $\Pi_\mathcal{V}$; they will be obtained under suitable transversality assumptions. For definiteness, let us recall the definition of transversality.

Definition 2.2. *Let $\mathbb{E}$ be a Euclidean space, let $\mathcal{V} \subset \mathbb{E}$ be a manifold of class C^1, and let E be an affine subspace of $\mathbb{E}$. We say that E and $\mathcal{V}$ intersect transversally at $\zeta \in E \cap \mathcal{V}$ if*

$$\mathrm{codim}(E^0 \cap T_\zeta\mathcal{V}^0) = \mathrm{codim}(E^0) + \mathrm{codim}(T_\zeta\mathcal{V}^0).$$

In particular, suppose that $\mathbb{E}$ has dimension n, $\mathcal{V}$ has dimension s, and E has dimension d; then, a necessary condition for them to intersect transversally is that $s + d \geq n$. In that case, remark that $E^0 \cap T_\zeta\mathcal{V}^0$ has dimension $t = s + d - n$.

Under such a transversality assumption, we obtain the following results on the existence of smooth bases of several vector spaces.

Lemma 2.3. *Let $\mathbb{E}$ be a Euclidean space of dimension n, let E be an affine subspace of $\mathbb{E}$ of dimension $d < n$, and let $\mathcal{V} \subset \mathbb{E}$ be a manifold of dimension s and of class C^k with $k \geq 1$. Suppose that E and $\mathcal{V}$ intersect transversally at a point $\zeta \in E \cap \mathcal{V}$; let further $t = s + d - n$ be the dimension of $E^0 \cap T_\zeta\mathcal{V}^0$.*

Then, there exists an open neighborhood U of ζ and functions $e_1, \ldots, e_t, e'_{t+1}, \ldots, e'_d$ and $e''_{t+1}, \ldots, e''_s$, all of class $C^{k-1} : U \to \mathbb{E}$, such that the following holds:

- *for x in U, the families $(e_1(x), \ldots, e_t(x))$, $(e_1(x), \ldots, e_t(x), e'_{t+1}(x), \ldots, e'_d(x))$ and $(e_1(x), \ldots, e_t(x), e''_{t+1}(x), \ldots, e''_s(x))$ are orthonormal*

and, for x in $\mathcal{V} \cap U$:

- *the intersection $E \cap T_x\mathcal{V}$ is not empty;*

- *$(e_1(x), \ldots, e_t(x))$ is a basis $E^0 \cap T_x\mathcal{V}^0$;*

6

- $(e_1(x), \ldots, e_t(x), e'_{t+1}(x), \ldots, e'_d(x))$ *is a basis of* E^0*;*

- $(e_1(x), \ldots, e_t(x), e''_{t+1}(x), \ldots, e''_s(x))$ *is a basis of* $T_x \mathcal{V}^0$*.*

Proof. There exist linear forms $\ell_1, \ldots, \ell_{n-d}$ and constants $b_1, \ldots, b_{n-d}$ such that for all u in $\mathbb{E}$, u is in E if and only if $\ell_i(u) = b_i$ for all i in $\{1, \ldots, n-d\}$. Similarly, taking the gradients of implicit equations $\varphi_1 = \cdots = \varphi_{n-s} = 0$ that define $\mathcal{V}$ around ζ, we see that there exists a neighborhood U of ζ, functions $\ell'_1, \ldots, \ell'_{n-s} : U \times \mathbb{E} \to \mathbb{R}$ of class C^{k-1} in $x \in U$ and linear in $u \in \mathbb{E}$, and functions $b'_1, \ldots, b'_{n-s} : U \to \mathbb{R}$ of class C^{k-1} such that for x in $\mathcal{V} \cap U$, $u \in \mathbb{E}$ belongs to $T_x \mathcal{V}$ if and only if $\ell'_j(x, u) = b'_j(x)$ for all j in $\{1, \ldots, n-s\}$.

Thus, for a given x in $\mathcal{V} \cap U$, u belongs to $E \cap T_x \mathcal{V}$ if and only if the linear equations $\ell_i(u) = b_i$ and $\ell'_j(x, u) = b'_j(x)$ are satisfied for all i in $\{1, \ldots, n-d\}$ and j in $\{1, \ldots, n-s\}$. Call $\eta_1, \ldots, \eta_{2n-s-d}$ the linear forms defining the homogeneous part of these equations; the corresponding homogeneous linear system $\eta_i = 0$ defines $E^0 \cap T_x \mathcal{V}^0$. The transversality assumption shows that for $x = \zeta$, the $(2n - s - d) \times n$ matrix of this system has full rank $2n - s - d$. By continuity, this remains true for x in a neighborhood of ζ, and for such x, $E \cap T_x \mathcal{V}$ is not empty. Up to restricting U, this proves the second item.

Applying Cramer's formulas, we can deduce from $(\varphi_1, \ldots, \varphi_{n-s})$ and $(\ell_1, \ldots, \ell_{n-d})$ functions $(\varepsilon_1, \ldots, \varepsilon_t)$, $(\varepsilon'_{t+1}, \ldots, \varepsilon'_d)$, $(\varepsilon''_{t+1}, \ldots, \varepsilon''_s)$, with all $\varepsilon_i, \varepsilon'_j, \varepsilon''_k$ of class $C^{k-1} : U \to \mathbb{E}$, such that for x in U, the vector families $(\varepsilon_1(x), \ldots, \varepsilon_t(x))$, $(\varepsilon_1(x), \ldots, \varepsilon_t(x), \varepsilon'_{t+1}(x), \ldots, \varepsilon'_d(x))$ and $(\varepsilon_1(x), \ldots, \varepsilon_t(x), \varepsilon''_{t+1}(x), \ldots, \varepsilon''_s(x))$ are nullspace bases for respectively

$$
\begin{aligned}
\ell_1(u) = \cdots = \ell_{n-d}(u) = D\varphi_1(x)(u) = \cdots = D\varphi_{n-s}(x)(u) &= 0 \\
\ell_1(u) = \cdots = \ell_{n-d}(u) &= 0 \\
D\varphi_1(x)(u) = \cdots = D\varphi_{n-s}(x)(u) &= 0.
\end{aligned}
$$

In particular, if x is actually in $\mathcal{V} \cap U$, those are bases for respectively $E^0 \cap T_x \mathcal{V}^0$, E^0 and $T_x \mathcal{V}^0$. Applying Gram-Schmidt orthogonalization to these families of functions, we obtain the functions in the statement of the lemma. $\qquad\square$

The following result is a direct corollary of the previous lemma.

Lemma 2.4. *Let $\mathbb{E}$ be a Euclidean space, let $\mathcal{V} \subset \mathbb{E}$ be a manifold of class C^1 and let E be an affine subspace of $\mathbb{E}$. Suppose that E and $\mathcal{V}$ intersect transversally at a point $\zeta \in E \cap \mathcal{V}$. Then, there exists a neighborhood U of ζ such that for x in U, $\Pi_{\mathcal{V}}(x)$ is well-defined and the intersection $E \cap T_{\Pi_{\mathcal{V}}(x)} \mathcal{V}$ is not empty.*

Proof. Let U_0 be a neighborhood of ζ such that $\Pi_{\mathcal{V}}$ is well-defined and continuous in U_0 and such that the intersection $E \cap T_x \mathcal{V}$ is not empty for x in $\mathcal{V} \cap U_0$ (such an U_0 exists by Lemmas 2.1 and 2.3). Then, take $U = \Pi_{\mathcal{V}}^{-1}(\mathcal{V} \cap U_0) \cap U_0$. $\qquad\square$

In the particular case where $\mathbb{E} = \mathcal{M}_{p,q}(\mathbb{R})$ and $\mathcal{V} = \mathcal{D}_r \subset \mathcal{M}_{p,q}(\mathbb{R})$, the projection $\Pi_{\mathcal{D}_r}$ can be made explicit using the *Eckart-Young Theorem*, which shows that $\Pi_{\mathcal{D}_r}(M)$ can be computed from the *singular value decomposition* of M:

Theorem 2.5. *Let $M \in \mathcal{M}_{p,q}(\mathbb{R})$ be a matrix, $M = U \cdot S \cdot V^{\mathsf{T}}$ be its singular value decomposition and $\sigma_1 \geq \cdots \geq \sigma_{\min(p,q)}$ be its singular values. Assume that $\sigma_r \neq \sigma_{r+1}$ and let $\widetilde{S}$ be the diagonal matrix defined by*

$$\widetilde{S}_{i,i} = \begin{cases} S_{i,i} \ \text{if} \ S_{i,i} \geq \sigma_r \\ 0 \ \text{otherwise} \end{cases}$$

Then there exists a unique matrix $\Pi_{\mathcal{D}_r}(M)$ of rank r minimizing the distance to M and this matrix is given by $\Pi_{\mathcal{D}_r}(M) = U \cdot \widetilde{S} \cdot V^{\mathsf{T}}$.

The last notion we will need is the *Moore-Penrose pseudoinverse* of either a matrix or a linear mapping A; in both cases, we will denote it by $A^{\dagger}$. Its main feature is that the solution of a consistent linear system $Ax = y$ with minimal 2-norm is given by $A^{\dagger}y$ (in the non-consistent case, this outputs the minimizer for the residual $Ax - y$).

2.2 Cadzow's algorithm: alternating projections

The first occurrence of the general problem of structured low rank approximation that we are aware of is described in [7]. In this paper, Cadzow introduces an algorithm based on alternating projections to solve SLRA problems. A solution M' of an SLRA problem should verify two properties:

- **(P1)** $M' \in E$;

- **(P2)** $\mathrm{rank}(M') \leq r$.

Cadzow's algorithm, as illustrated in Figure 1, proceeds by looking successively for the nearest matrices which satisfy alternatively **(P1)** and **(P2)**. The nearest matrix verifying **(P1)** is obtained by the orthogonal projection on E, and, as prescribed by the Eckart-Young theorem, the closest matrix verifying **(P2)** is obtained by truncating its Singular Value Decomposition.

We would like to emphasize that in the general case (and in most applications), the intersection $E \cap \mathcal{D}_r$ has positive dimension, whereas in Figure 1 (and all further ones), this intersection appears to have dimension zero.

Details of Cazdow's algorithm are given in Algorithm 1 below. In this algorithm, for $M \in \mathcal{M}_{p,q}(\mathbb{R})$, the subroutine SVD($M$) returns three matrices $U \in \mathcal{M}_{p,p}(\mathbb{R}), S \in \mathcal{M}_{p,q}(\mathbb{R}), V \in \mathcal{M}_{q,q}(\mathbb{R})$, such that $M = U \cdot S \cdot V^{\mathsf{T}}$, U and V are unitary matrices, and S is diagonal. The diagonal entries of S are the singular values of M, sorted in decreasing order.

Algorithm 1 (which is sometimes called *lift-and-project* or *alternating projections* in the literature) converges linearly towards a matrix $\hat{M}$ which verifies both conditions **(P1)** and **(P2)**, as proved in [31]. In this context, the linear convergence means that if $(M_i)_{i \geq 0}$ is the sequence of iterates of Cadzow's algorithm converging towards $\lim_{i \to \infty} M_i = M_\infty$, then there exists a positive constant c such that

$$\|M_{i+1} - M_\infty\| \leq c \|M_i - M_\infty\|.$$

As pointed out in [13], iterating Algorithm 1 does not converge in general towards $\Pi_{E \cap \mathcal{D}_r}(M_0)$.
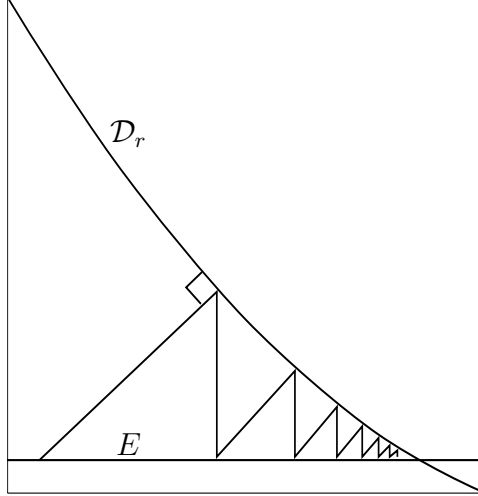
Figure 1: Cadzow's algorithm

Algorithm 1 one iteration of Cadzow's algorithm

1: **procedure** Cadzow(M, $(E_1, \ldots, E_d)$ an orthonormal basis of E^0, r)
2:     $U, S, V \leftarrow \mathrm{SVD}(M)$
3:     $U_r \leftarrow$ first r columns of U
4:     $V_r \leftarrow$ first r columns of V
5:     $S_r \leftarrow r \times r$ top-left sub-matrix of S
6:     $\widetilde{M} \leftarrow U_r \cdot S_r \cdot V_r^{\mathsf{T}}$
7:     **return** $M + \sum_{i=1}^{d} \langle \widetilde{M} - M, E_i \rangle E_i$
8: **end procedure**

2.3   Newton's method

Newton's method is an iterative technique to find zeros of real (or complex) functions. One of its main features is its quadratic rate of convergence: each iteration multiplies the number of significant digits of the solution by two. This iteration was first designed for systems with as many equations as variables, and was then successfully extended to non-square systems by using the Moore-Penrose pseudo-inverse. Let thus $f : \mathbb{E} \to \mathbb{F}$ be a differentiable mapping of Euclidean spaces. Then the Newton iteration is given by

$$\mathsf{Newton}_f(x) = x - Df(x)^\dagger f(x),$$

where, as said above, $Df(x)^\dagger$ denotes the Moore-Penrose pseudo-inverse of the linear application $Df(x)$.

In the underdetermined case (when $\dim(\mathbb{E}) > \dim(\mathbb{F})$), this iteration converges locally quadratically towards a point in $f^{-1}(\mathbf{0})$ if $Df(x)$ is locally surjective. The properties of this iteration have been deeply investigated during the last decades [4, 2, 16, 15].

Newton's method does not apply directly in our context. However, Figure 2 below suggests that in some cases (when for instance $\dim(\mathcal{D}_r) = \dim(E)$ and $\mathcal{D}_r$ is given as the graph of a mapping defined on E), using Newton iteration could lead to a fast iterative algorithm. Our algorithm is motivated by this remark.
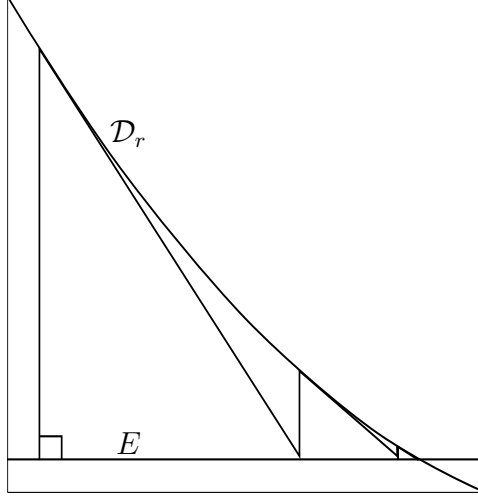

Figure 2: Newton's method

3   Algorithm `NewtonSLRA`

We propose an iterative algorithm `NewtonSLRA` which combines the applicability of Cadzow's algorithm and the quadratic convergence of Newton's iteration. Each of its iterations proceeds in the following three main steps.

- First, compute the projection $\widetilde{M} = \Pi_{\mathcal{D}_r}(M)$ onto the determinantal variety $\mathcal{D}_r$ (lines 2-6 in Algorithm 2);

- next, compute a set of generators of the normal space $N_{\widetilde{M}}\mathcal{D}_r$ (lines 7-11);

- finally, compute the point in $E \cap T_{\widetilde{M}}\mathcal{D}_r$ which minimizes the distance to M (lines 12-14).

We propose two dual methods for computing the last step, leading to the two variants `NewtonSLRA/1` and `NewtonSLRA/2` whose pseudo-codes are given in Algorithm 2 and Algorithm 3. Their main difference is the size of an intermediate matrix leading to the differences in their domains of efficiency: `NewtonSLRA/1` is well-suited when r is large and d is small, whereas `NewtonSLRA/1` performs better when r is small and d is large.

In Figure 3, we show one iteration of Algorithm `NewtonSLRA`; remark that the first step is similar to what happens in Cadzow's algorithm, but that we then use a linearization inspired by Newton's iteration. Note as well that in this very particular example, $E \cap T_{\widetilde{M}}\mathcal{D}_r$
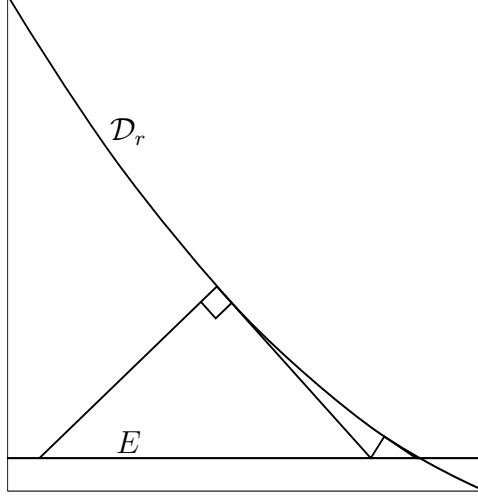
Figure 3: `NewtonSLRA`

has dimension zero, whereas this may not be the case in general. Nevertheless, this figure suggests that our algorithm may converge quadratically (we prove this rate of convergence in Section 4).

Notes on the pseudo-code of `NewtonSLRA`. While it is convenient to introduce the matrices $N_{(i-1)(q-r)+j} = \widetilde{u}_i \cdot \widetilde{v}_j^\intercal$ (and T_ℓ for the variant `NewtonSLRA/2`) to prove the correctness of Algorithms 2 and 3, they do not need to be explicitly computed. All that the algorithm needs are inner products of the form $\langle N_\ell, X \rangle$ (or $\langle T_\ell, X \rangle$ in `NewtonSLRA/2`) for various matrices X. Such an inner product can be computed efficiently by the formula

$$\langle \widetilde{u}_i \cdot \widetilde{v}_j^\intercal, X \rangle = \widetilde{u}_i^\intercal \cdot X \cdot \widetilde{v}_j.$$

Also, the Moore-Penrose pseudo-inverses $A^\dagger$ and $A'^\dagger$ do not need to be computed: what is actually needed is the solution of the linear least square problem $\mathrm{argmin}_x \|x\|$ subject to $A \cdot x = b$ (resp. $A' \cdot x = b'$). To our knowledge, using this trick does not change the asymptotic complexity, but it can make a notable efficiency improvement in practice.

Proposition 3.1 (Correctness of `NewtonSLRA`). *Suppose that $\mathcal{D}_r$ and E intersect transversally at a point $\zeta \in \mathcal{D}_r \cap E$. There exists an open neighborhood U of ζ such that if $M \in U \cap E$ and $(E_1, \ldots, E_d)$ is an orthonormal basis of E^0, then $\Pi_{E \cap T_{\widetilde{M}} \mathcal{D}_r}(M)$ is well-defined, for $\widetilde{M} = \Pi_{\mathcal{D}_r}(M)$, and Algorithms 2 and 3 with input $(M, (E_1, \cdots, E_d), r)$ return $\Pi_{E \cap T_{\widetilde{M}} \mathcal{D}_r}(M)$.*

The proof of this proposition, together with the cost analysis of the algorithm, occupy the end of this section. Let U be the neighborhood of ζ as defined in Lemma 2.4. In view of that lemma, $\Pi_{\mathcal{D}_r}$ is well-defined on U, and so is the mapping $M \mapsto \Pi_{E \cap T_{\Pi_{\mathcal{D}_r}(M)} \mathcal{D}_r}(M)$. In what follows, we let $\varphi : U \cap E \to E$ denote the latter function; thus, our claim is that Algorithm `NewtonSLRA` computes the mapping φ.

11

Algorithm 2 one iteration of `NewtonSLRA/1` algorithm

1: **procedure** `NewtonSLRA/1`$(M \in E, (E_1, \ldots, E_d)$ an orthonormal basis of $E^0, r \in \mathbb{N})$
2:     $(U, S, V) \leftarrow \text{SVD}(M)$
3:     $S_r \leftarrow r \times r$ top-left sub-matrix of S
4:     $U_r \leftarrow$ first r columns of U
5:     $V_r \leftarrow$ first r columns of V
6:     $\widetilde{M} \leftarrow U_r \cdot S_r \cdot V_r^\intercal$
7:     $\widetilde{u_1}, \ldots, \widetilde{u_{p-r}} \leftarrow$ last $p - r$ columns of U
8:     $\widetilde{v_1}, \ldots, \widetilde{v_{q-r}} \leftarrow$ last $q - r$ columns of V
9:     **for** $i \in \{1, \ldots, p - r\}, j \in \{1, \ldots, q - r\}$ **do**
10:         $N_{(i-1)(q-r)+j} \leftarrow \widetilde{u_i} \cdot \widetilde{v_j}^\intercal$
11:     **end for**
12:     $A \leftarrow (\langle N_k, E_\ell \rangle)_{k,\ell} \in \mathcal{M}_{(p-r)(q-r),d}(\mathbb{R})$
13:     $b \leftarrow (\langle N_k, \widetilde{M} - M \rangle)_k \in \mathcal{M}_{(p-r)(q-r),1}(\mathbb{R})$
14:     **return** $M + \sum_{\ell=1}^{d} \left(A^\dagger \cdot b \right)_\ell E_\ell$
15: **end procedure**

Algorithm 3 one iteration of `NewtonSLRA/2` algorithm

1: **procedure** `NewtonSLRA/2`$(M \in E, (E'_1, \ldots, E'_{pq-d})$ an orthonormal basis of $(E^0)^\perp, r \in \mathbb{N})$
2:     $(U, S, V) \leftarrow \text{SVD}(M)$
3:     $S_r \leftarrow r \times r$ top-left sub-matrix of S
4:     $U_r \leftarrow$ first r columns of U
5:     $V_r \leftarrow$ first r columns of V
6:     $\widetilde{M} \leftarrow U_r \cdot S_r \cdot V_r^\intercal$
7:     $u_1, \ldots, u_p \leftarrow$ columns of U
8:     $v_1, \ldots, v_q \leftarrow$ columns of V
9:     $(T_\ell)_{1 \leq \ell \leq (p+q-r)r} \leftarrow$ list of all matrices of the form $u_i \cdot v_j^\intercal$, where $i \leq r$ or $j \leq r$
10:     $A' \leftarrow (\langle E'_k, T_\ell \rangle)_{k,\ell} \in \mathcal{M}_{pq-d,(p+q-r)r}(\mathbb{R})$
11:     $b' \leftarrow (\langle E'_k, M - \widetilde{M} \rangle)_k \in \mathcal{M}_{pq-d,1}(\mathbb{R})$
12:     **return** $\widetilde{M} + \sum_{\ell=1}^{(p+q-r)r} \left(A'^\dagger \cdot b' \right)_\ell T_\ell$
13: **end procedure**

The following classical result yields an explicit description of the tangent and normal spaces of determinantal varieties. The notation $\mathrm{Hom}(\mathbb{R}^q, \mathbb{R}^p)$ stands for the set of $\mathbb{R}$-linear maps from $\mathbb{R}^q$ to $\mathbb{R}^p$.

Lemma 3.2. *Let $M \in \mathcal{M}_{p,q}(\mathbb{R})$ be such that $\mathrm{rank}(M) = r$. Let ℓ be the linear application*

$$\ell: \begin{array}{ccc} \mathbb{R}^q & \longrightarrow & \mathbb{R}^p \\ v & \longmapsto & M \cdot v \end{array}$$

Then the tangent space of $\mathcal{D}_r$ at M satisfies

$$\begin{aligned} T_M \mathcal{D}_r^0 &= \mathrm{Im}(\ell) \otimes \mathbb{R}^q + \mathbb{R}^p \otimes \mathrm{Ker}(\ell)^\perp \\ &= \{\ell' \in \mathrm{Hom}(\mathbb{R}^q, \mathbb{R}^p) \mid \ell'(\mathrm{Ker}(\ell)) \subset \mathrm{Im}(\ell)\} \end{aligned}$$

and the normal space to $\mathcal{D}_r$ at M satisfies

$$N_M \mathcal{D}_r^0 = \mathrm{Ker}(M^\intercal) \otimes \mathrm{Ker}(M).$$

Proof. Classical references for the proof of these claims are [3], [19, Section 3] and [23, Ch. 6,§1]. We recall the proof of the last claim with the notation used in this paper. Let $\{a_1, \ldots, a_{p-r}\}$ be a basis of $\mathrm{Ker}(M^\intercal)$, and $\{b_1, \ldots, b_{q-r}\}$ be a basis of $\mathrm{Ker}(M)$. Then the set $\{a_i \otimes b_j\}_{i,j}$ is a basis of $\mathrm{Ker}(M^\intercal) \otimes \mathrm{Ker}(M)$. Now let $v \in T_M \mathcal{D}_r^0$ be a tangent vector. In view of the first claim, it can be rewritten as a finite sum $\sum_k c_k \otimes d_k \in T_M \mathcal{D}_r^0$ where $c_k \in \mathrm{Im}(M)$ or $d_k \in \mathrm{Ker}(M)^\perp$. Consequently, $\langle a_i \otimes b_j, v \rangle = \sum_k \langle a_i, c_k \rangle \langle b_j, d_k \rangle = 0$ and thus $\mathrm{Ker}(M^\intercal) \otimes \mathrm{Ker}(M) \subset N_M \mathcal{D}_r^0$. Finally, since $\dim(N_M \mathcal{D}_r^0) = (p-r)(q-r) = \dim(\mathrm{Ker}(M^\intercal) \otimes \mathrm{Ker}(M))$, we obtain $\mathrm{Ker}(M^\intercal) \otimes \mathrm{Ker}(M) = N_M \mathcal{D}_r^0$. $\square$

Proof of Proposition 3.1. We are now able to prove the correctness of the two variants of `NewtonSLRA`. As in the algorithm, let us define $\widetilde{M} = U_r \cdot S_r \cdot V_r^\intercal$, where S_r is the $r \times r$ top-left sub-matrix of S, and U_r and V_r are made of the first r columns of respectively U and V. Then, by the Eckart-Young Theorem, for $M \in U$, the matrix $\widetilde{M}$ is equal to $\Pi_{\mathcal{D}_r}(M)$. Besides, by construction, the vectors $\widetilde{u_1}, \ldots, \widetilde{u_{p-r}}$ (resp. $\widetilde{v_1}, \ldots, \widetilde{v_{q-r}}$) are a basis of $\mathrm{Ker}(\widetilde{M^\intercal})$ (resp. $\mathrm{Ker}(\widetilde{M})$). Then, the previous lemma implies that the matrices N_ℓ in `NewtonSLRA/1` (resp. T_ℓ in `NewtonSLRA/2`) are a basis of the normal space $N_{\widetilde{M}} \mathcal{D}_r$ (resp. a basis of the tangent space $T_{\widetilde{M}} \mathcal{D}_r$).

Let $\varphi(M)$ denote $\Pi_{E \cap T_{\widetilde{M}} \mathcal{D}_r}(M)$. In order to conclude, we have to prove that the matrix computed at line 14 is $\varphi(M)$, that is, that (with the notation of the algorithms)

$$M + \sum_{\ell=1}^d \left(A^\dagger \cdot b\right)_\ell E_\ell = \widetilde{M} + \sum_{\ell=1}^{(p+q-r)r} \left(A'^\dagger \cdot b'\right)_\ell T_\ell = \Pi_{E \cap T_{\widetilde{M}} \mathcal{D}_r}(M).$$

An element F of $\mathcal{M}_{p,q}(\mathbb{R})$ belongs to $E \cap T_{\widetilde{M}} \mathcal{D}_r$ if and only if $F - M$ is in E^0 and $F - \widetilde{M}$ is in $T_{\widetilde{M}} \mathcal{D}_r^0$. The first condition is equivalent to the existence of $a_1, \ldots, a_d$ such that $F - M = \sum_{i=j}^d a_j E_j$ and the second one holds when

$$\forall i \in \{1, \ldots, (p-r)(q-r)\}, \quad \langle N_i, F - \widetilde{M} \rangle = 0;$$

taking into account the first constraint, the latter ones become, for all $i \in \{1, \ldots, (p-r)(q-r)\}$,

$$
\begin{aligned}
\langle N_i, M - \widetilde{M} \rangle + \sum_{j=1}^{d} a_j \langle N_i, E_j \rangle &= \langle N_i, M - \widetilde{M} \rangle + \langle N_i, F - M \rangle \\
&= 0.
\end{aligned}
$$

As in the algorithm, set

$$
A = \begin{bmatrix} \langle N_1, E_1 \rangle & \cdots & \langle N_1, E_d \rangle \\ \vdots & \vdots & \vdots \\ \langle N_{(p-r)(q-r)}, E_1 \rangle & \cdots & \langle N_{(p-r)(q-r)}, E_d \rangle \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} \langle N_1, \widetilde{M} - M \rangle \\ \vdots \\ \langle N_{(p-r)(q-r)}, \widetilde{M} - M \rangle \end{bmatrix}.
$$

Then, the previous discussion shows that F belongs to $E \cap T_{\widetilde{M}} \mathcal{D}_r$ if and only if

$$
F = M + \sum_{i=j}^{d} a_j E_j,
$$

where $a_1, \ldots, a_d$ satisfy the linear system

$$
A \cdot \begin{bmatrix} a_1 \\ \vdots \\ a_d \end{bmatrix} = b.
$$

By construction, $\varphi(M)$ is the matrix satisfying these constraints that minimizes $\|\varphi(M) - M\|$. Since $(E_1, \ldots, E_d)$ is an orthonormal basis, $\|\varphi(M) - M\|^2 = \sum_{i=1}^{d} a_i^2$ and hence the least square condition on $\varphi(M) - M$ amounts to finding the solution $a_1, \ldots, a_d$ of the former linear system that minimizes the 2-norm (we know that this linear system is consistent, since $E \cap T_{\widetilde{M}} \mathcal{D}_r$ is not empty). The least-square solution can be obtained with the Moore-Penrose pseudo-inverse of A, so we finally deduce that

$$
\begin{bmatrix} a_1 \\ \vdots \\ a_d \end{bmatrix} = A^\dagger \cdot b,
$$

and hence $\varphi(M) = M + \sum_{i=1}^{d} \left(A^\dagger \cdot b \right)_i E_i$. This proves the correctness of `NewtoNSLRA/1`.

The correctness of `NewtonSLRA/2` is proved similarly, by writing $F - \widetilde{M} = \sum_{\ell=1}^{(p+q-r)r} a'_\ell T_\ell$ for unknown values $a'_\ell \in \mathbb{R}$. The condition $F - M = \widetilde{M} - M + \sum_{\ell=1}^{(p+q-r)r} a'_\ell T_\ell \in E$ becomes

$$
\langle \widetilde{M} - M, E'_i \rangle + \sum_{\ell=1}^{(p+q-r)r} a'_\ell \langle T_\ell, E'_i \rangle = 0
$$

and the rest of the proof is similar to the one above.

$\square$

Complexity. All subroutines that appear in `NewtonSLRA` are linear algebra algorithms. In particular, one iteration needs to compute:

- the Singular Value Decomposition of the $p \times q$ matrix M;

- the matrix $\widetilde{M}$;

- $O(d(p-r)(q-r))$ inner products between matrices of size $p \times q$ (with $d = \dim(E)$), of the form $\langle N_{(i-1)(q-r)+j}, E_\ell \rangle$ or $\langle N_{(i-1)(q-r)+j}, \widetilde{M} - M \rangle$, with $N_{(i-1)(q-r)+j} = \widetilde{u}_i \cdot \widetilde{v}_j{}^\mathsf{T}$;

- the Moore-Penrose pseudoinverse of the $(p-r)(q-r) \times d$ matrix A (`NewtonSLRA/1`) or the $(p+q-r)r \times (pq-d)$ matrix A' (`NewtonSLRA/2`);

- the output $\varphi(M) = M + \sum_{i=1}^{d} \left(A^\dagger \cdot b \right)_i E_i$.

As explained in the introduction, we would want to give simple complexity statements, counting arithmetic operations $+, -, \times, \div$ over the reals at unit cost, avoiding the discussion of accuracy inherent to floating-point arithmetic. This is not possible for the Singular Value Decomposition, so we will simply take this computation as a black-box.

Computing $\widetilde{M}$ can be done in $O(pqr)$ arithmetic operations. For $N_\ell = \widetilde{u}_i \cdot \widetilde{u}_j{}^\mathsf{T}$, the inner products of the form $\langle N_\ell, X \rangle$ can be computed by the formula

$$\langle \widetilde{u}_i \cdot \widetilde{v}_j{}^\mathsf{T}, X \rangle = \widetilde{u}_i{}^\mathsf{T} \cdot X \cdot \widetilde{v}_j$$

in $O(pq)$ arithmetic operations, for a total of $O(pqd(p-r)(q-r))$ for the construction of the matrix A. Similarly, the matrix A' in `NewtonSLRA/2` can be constructed within $O(pqr(pq - d)(p+q-r))$ operations. The Moore-Penrose pseudoinverse of A (resp. A') can then be computed in $O(d(p-r)^2(q-r)^2)$ arithmetic operations (resp. $O((pq-d)^2(p+q-r)r))$), and deducing $\varphi(M)$ can be done in $O(dpq)$ operations in `NewtonSLRA/1` (resp. $O(pqr(p+q-r))$ in `NewtonSLRA/2`).

Altogether, up to the SVD computation, all operations can be achieved within

- $O(pqd(p-r)(q-r) + pqr)$ arithmetic operations for `NewtonSLRA/1`;

- $O(pqr(pq-d)(p+q-r))$ arithmetic operations for `NewtonSLRA/2`.

In particular, the cost of `NewtonSLRA/1` is at most quadratic in the size of the input (specifying the basis $E_1, \ldots, E_d$ of E^0 requires $O(dpq)$ entries).

4 Rate of convergence

The aim of this section is to prove the local quadratic convergence of `NewtonSLRA` and to control the distance between its output and the optimal solution of the SLRA problem. The results given in this part of the paper are more general than the SLRA context: as in [31], we will perform our analysis for a manifold $\mathcal{V}$ in a Euclidean space $\mathbb{E}$ of class C^3, instead of $\mathcal{D}_r$;

as before, we let E be a proper affine subspace of $\mathbb{E}$. We assume without loss of generality that $\mathcal{V} \neq \mathbb{E}$.

Let $\zeta \in E \cap \mathcal{V}$ be such that the intersection of E and $\mathcal{V}$ is transverse at ζ. By Lemmas 2.1 and 2.4, we know that in a neighborhood U of ζ, the mapping $x \mapsto \Pi_{\mathcal{V}}(x)$ is well-defined and of class C^2, and the intersection $E \cap T_{\Pi_{\mathcal{V}}(x)}\mathcal{V}$ is not empty. As a result, the projection $\varphi : x \mapsto \Pi_{E \cap T_{\Pi_{\mathcal{V}}(x)}\mathcal{V}}(x)$ is itself well-defined over U. We saw in the previous section that in the case $\mathcal{V} = \mathcal{D}_r$, algorithm `NewtonSLRA` precisely computes the mapping φ. In the more general context of this section, we study the iterates $\varphi^n = \varphi \circ \cdots \circ \varphi$ (which will turn out to be well-defined, up to restricting the domain of φ).

The transversality assumption implies that, up to restricting U, the intersection $\mathcal{W} = E \cap \mathcal{V} \cap U$ is a manifold of class C^3. Up to restricting U further, we can assume (by means of Lemma 2.1) that the projection operator $\Pi_{\mathcal{W}}$ is well-defined and of class C^2 in U. In the context of Structured Low Rank Approximation, $\mathcal{W} = E \cap \mathcal{D}_r \cap U$, and the projection $\Pi_{\mathcal{W}}$ represents the optimal solution to our approximation problem.

The following theorems are the main results of this section; taken in the context of SLRA, they finish proving the theorems stated in the introduction.

The first part of the following theorem ensures the local quadratic convergence of the iterates of φ; the second part bounds the distance between the limit point of the iteration and the optimal solution $\Pi_{\mathcal{W}}(x_0)$. Roughly speaking, this shows that locally the limit of the iteration looks like the orthogonal projection on $\mathcal{W}$. This will be formalized in Theorem 4.2.

Theorem 4.1. *Let ζ be in $E \cap \mathcal{V}$ such that $\Pi_{\mathcal{V}}$ is C^2 around ζ and $\mathcal{V}$ and E intersect transversally at ζ. There exists $\nu, \gamma, \gamma' > 0$ such that, for all $x_0 \in B_\nu(\zeta)$, the sequence (x_i) given by $x_{i+1} = \varphi(x_i)$ is well-defined and converges towards a point $x_\infty \in \mathcal{W}$, with*

- $\|x_{i+1} - x_\infty\| \leq \gamma \|x_i - x_\infty\|^2$ *for $i \geq 0$;*

- $\|\Pi_{\mathcal{W}}(x_0) - x_\infty\| \leq \gamma' \|\Pi_{\mathcal{W}}(x_0) - x_0\|^2$.

In general, $x_\infty \neq \Pi_{\mathcal{W}}(x_0)$; in particular, `NewtonSLRA` will usually not converge to the optimal solution of an SLRA problem. Nevertheless, the following theorem shows that Φ is a good local approximation of the function $\Pi_{\mathcal{W}}$ around $\mathcal{W}$.

Theorem 4.2. *Let ζ be in $E \cap \mathcal{V}$ such that $\Pi_{\mathcal{V}}$ is C^2 around ζ and $\mathcal{V}$ and E intersect transversally at ζ, and let $\Phi : B_\nu(\zeta) \to \mathbb{E}$ denote the limit operator $\Phi(x) = x_\infty$, for x_∞ as in Theorem 4.1. Then, Φ is differentiable at ζ and $D\Phi(\zeta) = \Pi_{T_\zeta \mathcal{W}^0}$.*

Note that in the context of SLRA, $\mathcal{D}_r$ and $E \cap \mathcal{D}_r$ are of class C^∞ in the neighborhood of points $\zeta \in E \cap \mathcal{D}_r$ where the intersection is transverse.

4.1 Angle between linear subspaces

Our analysis will rely on the notion of *angle* between two linear subspaces (see *e.g.* [21], [17, Ch. 9], [31, Section 3]). In what follows, $\mathbb{S} = \{x \in \mathbb{E} : \|x\| = 1\}$ denotes the unit sphere and $M^\perp$ denotes the orthogonal complement of a linear subspace M of $\mathbb{E}$.

Definition 4.3 (angle between linear subspaces). *Let $M, N \subset \mathbb{E}$ be two linear subspaces. If $N \subset M$ or $M \subset N$, we set $\alpha(M, N) = 0$. Otherwise, their angle $\alpha(M, N)$ is the value in $[0, \pi/2]$ defined by*

$$\alpha(M, N) := \arccos\left(\max\{\langle x, y\rangle : x \in \mathbb{S} \cap M \cap (M \cap N)^\perp, y \in \mathbb{S} \cap N \cap (M \cap N)^\perp\}\right).$$

The following lemma (see [17, Lemma 9.5] for a proof) shows that when we consider the maximum of the scalar products, we only need one vector to be orthogonal to $M \cap N$.

Lemma 4.4. *If x is in $\mathbb{S} \cap M \cap (M \cap N)^\perp$ and y is in $\mathbb{S} \cap N$, then*

$$\langle x, y\rangle \leq \cos(\alpha(M, N)).$$

We can now describe a few consequences of our transversality assumptions for angles between various subspaces.

Lemma 4.5. *There exists an open neighborhood U of ζ such that $\inf_{x \in \mathcal{V} \cap U} \alpha(T_x \mathcal{V}^0, E^0) > 0$.*

Proof. First, notice that the angle $\alpha(M, N)$ between two linear subspaces M and N cannot be 0 if $M \not\subset N$ and $N \not\subset M$. Since by assumption $\mathcal{V} \neq \mathbb{E}$ and $E \neq \mathbb{E}$, and $\mathcal{V}$ and E intersect transversely at ζ, we have neither $T_\zeta \mathcal{V}^0 \subset E^0$ nor $E^0 \subset T_\zeta \mathcal{V}^0$. We deduce that $\alpha(T_\zeta^0 \mathcal{V}, E^0) \neq 0$.

The rest of the proof is similar to that of [31, Lemma 10]. Recall from Lemma 2.3 that for x in a neighborhood U_0 of ζ, we know orthonormal families $(e_1(x), \ldots, e_t(x))$, $(e_1(x), \ldots, e_t(x), e'_{t+1}(x), \ldots, e'_d(x))$ and $(e_1(x), \ldots, e_t(x), e''_{t+1}(x), \ldots, e''_s(x))$, that vary continuously with x, and that are bases of respectively $E^0 \cap T_x \mathcal{V}^0$, E^0 and $T_x \mathcal{V}^0$ whenever x is in $\mathcal{V} \cap U_0$.

For x in U_0, consider the linear mapping $\pi_x = \Pi_{S'(x)}\Pi_{S''(x)} - \Pi_{S(x)}$, where $S(x), S'(x), S''(x)$ are the vector spaces spanned by the three families above. The matrix of this linear mapping, and thus its operator norm, vary continuously with x.

Now, when x is in $\mathcal{V} \cap U_0$, π_x is the linear mapping $\Pi_{E^0}\Pi_{T_x \mathcal{V}^0} - \Pi_{E^0 \cap T_x \mathcal{V}^0}$. From [17, Ch. 9], we know that the norm of this operator is the cosine of $\alpha(T_x \mathcal{V}^0, E^0)$. This shows that at $x = \zeta$, the norm of π_x is nonzero; by continuity, this remains true in a neighborhood $U \subset U_0$ of ζ. $\qquad\square$

Lemma 4.6. *There exists an open neighborhood U of ζ such that for any x and y in $\mathcal{V} \cap U$, the intersection of the vector spaces $E^0 \cap T_x \mathcal{V}^0$ and $(E^0 \cap T_y \mathcal{V}^0)^\perp$ is trivial.*

Proof. Let $n = \dim(\mathbb{E})$, $d = \dim(E)$, $s = \dim(\mathcal{V})$ and $t = \dim(E^0 \cap T_\zeta \mathcal{V}^0)$; the transversality assumption shows that $t = s + d - n$.

Using again Lemma 2.3, we know that there exist a neighborhood U_0 of ζ and vectors $e_1(x), \ldots, e_t(x)$ depending continuously of $x \in U_0$, that form a orthonormal family, and whose span is $E^0 \cap T_x \mathcal{V}^0$ for x in $\mathcal{V} \cap U_0$. Then, up to restricting further U_0, we consider a local submersion $\psi : \mathbb{E} \to \mathbb{R}^{n-t}$ such that $\psi^{-1}(0) \cap U_0 = E \cap \mathcal{V} \cap U_0$. Applying Gram-Schmidt orthogonalisation to the gradient of ψ defines vectors $e_{t+1}(x), \ldots, e_n(x)$ that depend

continuously on x and such that $(e_1(x), \ldots, e_n(x))$ is an orthonormal basis of $\mathbb{E}$. In particular, when x is in $\mathcal{V} \cap U_0$, $(e_{t+1}(x), \ldots, e_n(x))$ is an orthonormal basis of $(E^0 \cap T_x \mathcal{V}^0)^\perp$.

For x and y in $\mathcal{V} \cap U_0$, the intersection of $E^0 \cap T_x \mathcal{V}^0$ and $(E^0 \cap T_y \mathcal{V}^0)^\perp$ is reduced to $\{0\}$ whenever the determinant Δ of the family $(e_1(x), \ldots, e_t(x), e_{t+1}(y), \ldots, e_n(y))$ is nonzero. The determinant Δ is a continuous function $U_0 \times U_0 \to \mathbb{R}$, and $\Delta(\zeta, \zeta)$ is nonzero, so there exists a neighborhood $\Omega \subset U_0 \times U_0$ of (ζ, ζ) that does not intersect $\Delta^{-1}(0)$. It is then enough to take U such that $U \times U \subset \Omega$. $\square$

Lemma 4.7. *Consider the mapping*

$$\begin{aligned} \Lambda : \quad \mathcal{V} \times \mathcal{V} \quad &\to \quad [0, 1] \\ (x, y) \quad &\mapsto \quad \cos(\alpha(E^0 \cap T_x \mathcal{V}^0, (E^0 \cap T_y \mathcal{V}^0)^\perp)). \end{aligned}$$

There exists an open neighborhood U of ζ and a constant λ such that for x, y in $\mathcal{V} \cap U$, $\Lambda(x, y)$ is well-defined, and the inequality $\Lambda(x, y) \leq \lambda \|x - y\|$ holds.

Proof. As before, let $n = \dim(\mathbb{E})$, $d = \dim(E)$, $s = \dim(\mathcal{V})$ and $t = \dim(E^0 \cap T_\zeta \mathcal{V}^0)$. Using Lemma 2.3, we know that there exist C^2 functions $e_1, \ldots, e_t : U \to \mathbb{E}$, defined in a neighborhood U_0 of ζ, such that for x in $\mathcal{V} \cap U$, $e_1(x), \ldots, e_t(x)$ is a orthonormal basis of $E^0 \cap T_x \mathcal{V}^0$. As in the previous lemma, this basis can be completed to an orthonormal basis $(e_1(x), \ldots, e_n(x))$ of $\mathbb{E}$, with functions $e_{t+1}, \ldots, e_n$ that are still C^2 around ζ.

Consider the function $\Gamma : U_0 \times U_0 \to \mathbb{R}$, such that $\Gamma(x, y)$ is the 2-norm of the linear mapping $\Pi_{e_1(x), \ldots, e_t(x)} \Pi_{e_{t+1}(y), \ldots, e_n(y)}$. Using the previous lemma, up to restricting U_0, we may also assume that for x and y both in $\mathcal{V} \cap U_0$, the intersection of the vector spaces $E^0 \cap T_x \mathcal{V}^0$ and $(E^0 \cap T_y \mathcal{V}^0)^\perp$ is trivial. Using [17, Ch. 9] as in Lemma 4.5, this implies in particular that for such x and y, $\Lambda(x, y) = \Gamma(x, y)$. Thus, we are going to prove that an inequality of the form $\Gamma(x, y) \leq C \|x - y\|$ holds for x and y in $\mathcal{V} \cap U$, for suitable $U \subset U_0$ and C.

Let U be an open ball centered at ζ, such that $\overline{U}$ is contained in U_0. Because $e_{t+1}, \ldots, e_n$ are C^1, there exists a constant $c \geq 0$ such that $\|e_i(x) - e_i(y)\| \leq c/n \|x - y\|$ holds for all x, y in U and i in $\{t+1, \ldots, n\}$.

The matrix P_y of the orthogonal projection $\Pi_{e_{t+1}(y), \ldots, e_n(y)}$ can be written as $P_y = R_y R_y^\intercal$, where R_y is the matrix with columns $e_{t+1}(y), \ldots, e_n(y)$. In particular, R_y can be rewritten as $R_y = R_x + \delta_{x,y}$, with R_x being the matrix with columns $e_{t+1}(x), \ldots, e_n(x)$ and where the operator norm of $\delta_{x,y}$ is bounded by $c \|x - y\|$. As a result, P_y can be rewritten as

$$\begin{aligned} P_y &= R_y R_y^\intercal \\ &= R_x R_x^\intercal + R_x \delta_{x,y}^\intercal + \delta_{x,y} R_x^\intercal + \delta_{x,y} \delta_{x,y}^\intercal \\ &= P_x + \Delta_{x,y}, \end{aligned}$$

with $\Delta_{x,y} = R_x \delta_{x,y}^\intercal + \delta_{x,y} R_x^\intercal + \delta_{x,y} \delta_{x,y}^\intercal$. By construction, the norm of $\delta_{x,y}$ is bounded by $c \|x - y\|$, and the norm of R_x is equal to 1. Consequently, the norm of $\Delta_{x,y}$ is bounded by $\lambda \|x - y\|$ on U, with $\lambda = 2c + c^2 \sup_{x,y \in U} \|x - y\|$ (up to restricting U, $\sup_{x,y \in U} \|x - y\|$ can be made arbitrarily small).

18

Let further S_x be the matrix of the orthogonal projection $\Pi_{e_1(x),\ldots,e_t(x)}$, and remark that $S_x P_x = 0$. In view of the above paragraphs, the matrix $Q_{x,y}$ of the linear mapping $\Pi_{e_1(x),\ldots,e_t(x)}\Pi_{e_{t+1}(y),\ldots,e_n(y)}$ can be rewritten as

$$
\begin{aligned}
Q_{x,y} &= S_x P_y \\
&= S_x P_x + S_x \Delta_{x,y} \\
&= S_x \Delta_{x,y}.
\end{aligned}
$$

Because the norm of $\Delta_{x,y}$ is bounded by $\lambda \|x - y\|$, and the norm of an orthogonal projection is at most 1, the norm of $Q_{x,y}$ is also bounded by $\lambda \|x - y\|$. This implies that $\Gamma(x, y)$, which is the norm of $Q_{x,y}$ is a most $\lambda \|x - y\|$. $\qquad\square$

4.2 Analysis of one iteration

In what follows, we work over an open neighborhood U of ζ that has the form $U = B_\rho(\zeta)$, for some $\rho > 0$ chosen such that

- φ, $\Pi_\mathcal{V}$ and $\Pi_\mathcal{W}$ are well-defined in the *closed* ball $\overline{B_\rho(\zeta)}$, with $\Pi_\mathcal{V}$ and $\Pi_\mathcal{W}$ of class C^2;

- the inequality $\alpha(T_v^0 \mathcal{V}, E^0) > 0$ (as in Lemma 4.5) and the conclusions of Lemmas 2.3, 4.6 and 4.7 hold in the closed ball $\overline{B_\rho(\zeta)}$;

- $\mathcal{W} \cap \overline{B_\rho(\zeta)}$ is closed (for the Euclidean topology).

Define the following:

- $\alpha_0 = \inf_{v \in \mathcal{V} \cap \overline{B_\rho(\zeta)}} \alpha(T_v \mathcal{V}^0, E^0)$, so that $\alpha_0 > 0$;

- $C_\mathcal{V} = \sup_{v \in \overline{B_\rho(\zeta)}} \|D^2 \Pi_\mathcal{V}(v)\|$;

- $C_\mathcal{W} = \sup_{z \in B_\rho(\zeta)} \|D\Pi_\mathcal{W}(z)\|$;

- λ is the constant introduced in Lemma 4.7;

- $K = \left(\frac{C_\mathcal{V}}{\sin(\alpha_0)} + \sqrt{2}\lambda\right)$

- $K' = C_\mathcal{W} K$

- $\delta > 0$ is such that $C_\mathcal{V}^2 \delta^2 \leq 1/2$ and $2\delta + K\delta^2 \leq \rho$ hold.

Proposition 4.8. *For x in $B_\delta(\zeta)$, the following properties hold:*

- *$\varphi(x)$ is in $B_\rho(\zeta)$, so $\Pi_\mathcal{W}(\varphi(x))$ is well-defined;*

- *$\|\varphi(x) - \Pi_\mathcal{W}(x)\| \leq K \|x - \Pi_\mathcal{W}(x)\|^2$;*

- *$\|\Pi_\mathcal{W}(\varphi(x)) - \Pi_\mathcal{W}(x)\| \leq K' \|x - \Pi_\mathcal{W}(x)\|^2$.*

The rest of this subsection is devoted to the proof of this proposition. Thus, we fix x in $B_\delta(\zeta)$ in all that follows; we also use the following shorthand: $y = \Pi_\mathcal{V}(x)$, $w = \Pi_\mathcal{W}(x)$ and $z = \Pi_{T_y \mathcal{V}}(w)$. Another pair of points w' and z' will be used: w' is the orthogonal projection of x on the affine space parallel to $E \cap T_y \mathcal{V}$ containing w, and $z' = \Pi_{T_y \mathcal{V}}(w')$.

19

Step 1: Some basic inequalities. First, notice that if x is in $B_\delta(\zeta)$, then we have

$$\|x - w\| \le \|x - \zeta\| < \delta,$$

because ζ is in $\mathcal{W}$ and $w = \Pi_{\mathcal{W}}(x)$, and

$$\|x - y\| \le \|x - \zeta\| < \delta,$$

because ζ is in $\mathcal{V}$ and $y = \Pi_{\mathcal{V}}(x)$. This implies that w and y belong to $B_{2\delta}(\zeta)$ and thus to $B_\rho(\zeta)$ since

$$\|w - \zeta\| \le \|w - x\| + \|x - \zeta\| < 2\delta \le \rho,$$
$$\|y - \zeta\| \le \|y - x\| + \|x - \zeta\| < 2\delta \le \rho.$$

Note also for further use that since $\|x - w\| < \delta$ and $\|x - y\| < \delta$, we also have $\|y - w\| < 2\delta$.

Step 2: Proof of inequality $\|z - w\| < C_{\mathcal{V}} \|x - w\|^2$. We continue by doing a Taylor approximation of $\Pi_{\mathcal{V}}$ between y and w. Since $\Pi_{\mathcal{V}}(w) = w$ and $\Pi_{\mathcal{V}}(y) = y$, and since all points of the line segment between y and w are in $B_\rho(\zeta)$, we obtain

$$
\begin{aligned}
w - y &= \Pi_{\mathcal{V}}(w) - \Pi_{\mathcal{V}}(y) \\
&= \Pi_{T_y \mathcal{V}^0}(w - y) + r,
\end{aligned}
$$

with $\|r\| \le \dfrac{C_{\mathcal{V}} \|w - y\|^2}{2}$. Because $y + \Pi_{T_y \mathcal{V}^0}(w - y) = \Pi_{T_y \mathcal{V}}(w) = z$, this implies

$$\|z - w\| \le \frac{C_{\mathcal{V}}}{2} \|y - w\|^2. \tag{1}$$

Since we saw previously that $\|y - w\| \le 2\delta$, we deduce in particular that

$$\|z - w\| \le C_{\mathcal{V}} \delta \|y - w\| \le 2 C_{\mathcal{V}} \delta^2.$$

Because $x - y$ is orthogonal to $T_y \mathcal{V}^0$, it is orthogonal to $y - z$, and similarly for $w - z$; these relations imply that $\|y - z\| \le \|x - w\|$. On the other hand, since $w - z$ is orthogonal to $y - z$, we also have by the Pythagorean theorem

$$\|y - w\|^2 = \|y - z\|^2 + \|z - w\|^2,$$

so that

$$\|y - w\|^2 \le \|x - w\|^2 + \|z - w\|^2.$$

From this inequality, using the upper bound $\|z - w\| \le 2 C_{\mathcal{V}} \delta^2$, we obtain

$$
\begin{aligned}
\|z - w\| &\le \frac{C_{\mathcal{V}}}{2} \|x - w\|^2 + \frac{C_{\mathcal{V}}}{2} \|z - w\|^2 \\
&\le \frac{C_{\mathcal{V}}}{2} \|x - w\|^2 + C_{\mathcal{V}}^2 \delta^2 \|z - w\| \\
&\le \frac{C_{\mathcal{V}}}{2} \|x - w\|^2 + \frac{1}{2} \|z - w\|,
\end{aligned}
$$

since δ is such that $C_{\mathcal{V}}^2 \delta^2 \le \frac{1}{2}$. We finally get, as claimed,

$$\|z - w\| < C_{\mathcal{V}} \|x - w\|^2. \tag{2}$$

Step 3: Proof of inequality $\|\varphi(x) - w'\| \le \|z' - w'\|/\sin(\alpha_0)$**.** To prove this inequality, let us introduce the angle ϑ between $w' - \varphi(x)$ and $z' - \varphi(x)$. First, we prove that $\cos(\vartheta) \le \cos(\alpha_0)$, by an application of Lemma 4.4.

- $w' - \varphi(x)$ is in E^0, because $w' - \varphi(x) = (w' - w) + (w - \varphi(x))$ and both summands are in E^0. By construction of w', $w' - w$ is in $(E \cap T_y \mathcal{V})^0$, which is in E^0, and w and $\varphi(x)$ are in E, so $w - \varphi(x)$ is indeed in E^0 as well.

- $z' - \varphi(x)$ is in $T_y \mathcal{V}^0$, because both z' and $\varphi(x)$ are in $T_y \mathcal{V}$.

- $z' - \varphi(x)$ is in the orthogonal complement of $(E \cap T_y \mathcal{V})^0$, because $z' - \varphi(x) = (z' - w') + (w' - x) + (x - \varphi(x))$, which are respectively orthogonal to $T_y \mathcal{V}^0$, $(E \cap T_y \mathcal{V})^0$ and $(E \cap T_y \mathcal{V})^0$. By Lemma 2.3, $E \cap T_y \mathcal{V}$ is not empty, and thus $(E \cap T_y \mathcal{V})^0 = E^0 \cap T_y \mathcal{V}^0$.

Thus, we can apply Lemma 4.4 to deduce $\cos(\vartheta) \le \cos(\alpha_0)$, as claimed. Alternatively, $1/\sin(\vartheta) \le 1/\sin(\alpha_0)$.

Second, remark that $w' - z'$ is orthogonal to $\varphi(x) - z'$. Indeed, the latter is in $T_y \mathcal{V}^0$, and by construction $w' - z'$ is orthogonal to $T_y \mathcal{V}^0$. This proves that $\|w' - z'\| = \sin(\vartheta) \|w' - \varphi(x)\|$, and thus the inequality

$$\|\varphi(x) - w'\| \le \frac{\|z' - w'\|}{\sin(\alpha_0)}. \tag{3}$$

Step 4: Proof of inequality $\|\varphi(x) - w\| \le \frac{C_\mathcal{V}}{\sin(\alpha_0)} \|x - w\|^2 + \|w' - w\|$**.** In order to establish this inequality, remark that the vectors $z - w$ and $z' - w'$ have the same norm. Indeed, $z - w$ and $z' - w'$ are orthogonal to $T_y \mathcal{V}^0$ by construction of z and z', and both $w - w'$ and $z - z'$ are in $T_y \mathcal{V}^0$. Using (2) and (3), we deduce

$$\|\varphi(x) - w'\| \le \frac{\|z - w\|}{\sin(\alpha_0)} \le \frac{C_\mathcal{V}}{\sin(\alpha_0)} \|x - w\|^2.$$

Using the triangle inequality $\|\varphi(x) - w\| \le \|\varphi(x) - w'\| + \|w' - w\|$, we finally deduce

$$\|\varphi(x) - w\| \le \frac{C_\mathcal{V}}{\sin(\alpha_0)} \|x - w\|^2 + \|w' - w\|. \tag{4}$$

Step 5: Proof of inequality $\|w - w'\| \le \lambda \|y - w\| \|x - w\|$**.** Let ϑ' be the angle between the vectors $w' - w$ and $x - w$; then, because $x - w'$ is orthogonal to $w' - w$, we have $\|w - w'\| = \cos(\vartheta') \|x - w\|$. We claim further that the inequality

$$\cos(\vartheta') \le \cos(\alpha(E^0 \cap T_y \mathcal{V}^0, (E^0 \cap T_w \mathcal{V}^0)^\perp))$$

holds. Indeed, this follows from applying Lemma 4.4 to the vectors $w' - w$ and $x - w$; let us briefly verify that its assumptions are satisfied:

- $w' - w$ is in $(E \cap T_y \mathcal{V})^0 = E^0 \cap T_y \mathcal{V}^0$, by construction.

21

- $x - w$ is orthogonal to $T_w \mathcal{W}^0$, and the transversality assumption (Lemma 2.3) shows that $T_w \mathcal{W} = E \cap T_w \mathcal{V}$.

- By Lemma 4.6, the vector spaces $E^0 \cap T_y \mathcal{V}^0$ and $(E^0 \cap T_w \mathcal{V}^0)^\perp$ have a trivial intersection.

Using Lemma 4.7, we deduce the inequality $\cos(\vartheta') \leq \lambda \, \|y - w\|$, and thus

$$\|w - w'\| \leq \lambda \, \|y - w\| \, \|x - w\| . \tag{5}$$

Step 6: Proof of inequality $\|y - w\| \leq \sqrt{2} \, \|x - w\|$. We established the following inequalities at Step 2:

$$\|z - w\| \leq C_\mathcal{V} \delta \, \|y - w\|$$

and

$$\|y - w\|^2 \leq \|x - w\|^2 + \|z - w\|^2 .$$

Combining these two inequalities gives

$$
\begin{aligned}
\|y - w\|^2 &\leq \|x - w\|^2 + \|z - w\|^2 \\
&\leq \|x - w\|^2 + C_\mathcal{V}^2 \delta^2 \, \|y - w\|^2 \\
&\leq \|x - w\|^2 + \frac{1}{2} \, \|y - w\|^2 ,
\end{aligned}
$$

since $\delta^2 C_\mathcal{V}^2 \leq 1/2$. Thus, we deduce

$$\|y - w\| \leq \sqrt{2} \, \|x - w\| . \tag{6}$$

Step 7: Proof of inequality $\|\varphi(x) - w\| \leq K \, \|x - w\|^2$. Combining the results of Steps 4, 5 and 6, we obtain the first inequality claimed in Proposition 4.8:

$$
\begin{aligned}
\|\varphi(x) - w\| &\leq \frac{C_\mathcal{V}}{\sin(\alpha_0)} \, \|x - w\|^2 + \|w' - w\| \\
&\leq \frac{C_\mathcal{V}}{\sin(\alpha_0)} \, \|x - w\|^2 + \lambda \, \|y - w\| \, \|x - w\| \\
&\leq \frac{C_\mathcal{V}}{\sin(\alpha_0)} \, \|x - w\|^2 + \sqrt{2} \lambda \, \|x - w\|^2 \\
&\leq K \, \|x - w\|^2 .
\end{aligned}
$$

Step 8: Proof that $\varphi(x)$ is in $B_\rho(\zeta)$. Next, we prove that $\|\zeta - \varphi(x)\| < \rho$. Indeed, recall that we saw in Step 1 that $\|\zeta - w\| < 2\delta$ and $\|x - w\| < \delta$. Using the inequality proved above, we deduce

$$
\begin{aligned}
\|\zeta - \varphi(x)\| &\leq \|\zeta - w\| + \|\varphi(x) - w\| \\
&< 2\delta + K \, \|x - w\|^2 \\
&< 2\delta + K\delta^2 ,
\end{aligned}
$$

which is less than ρ by construction of δ.

Step 9: Proof of inequality $\|\Pi_{\mathcal{W}}(\varphi(x)) - w\| \leq K' \|x - w\|^2$. This is last item required to conclude the proof of Proposition 4.8. A first order Taylor expansion of $\Pi_{\mathcal{W}}$ along the line segment joining $\varphi(x)$ to w (which are both in $B_\rho(\zeta)$) gives

$$
\begin{aligned}
\|\Pi_{\mathcal{W}}(\varphi(x)) - \Pi_{\mathcal{W}}(w)\| &\leq C_{\mathcal{W}} \|\varphi(x) - w\| \\
&\leq K C_{\mathcal{W}} \|x - w\|^2 \qquad \textit{c.f. Step 7} \\
&\leq K' \|x - w\|^2.
\end{aligned}
$$

Since $\Pi_{\mathcal{W}}(w) = w$, the proof is complete.

4.3 Convergence of `NewtonSLRA`

In this subsection, we study the behavior of the sequence defined by $x_{i+1} = \varphi(x_i)$: we prove that the sequence is well-defined for x_0 close enough to ζ and that it converges quadratically to a limit x_∞. In what follows, we write $\kappa = K + K'$ and we choose $\nu > 0$ such that $\kappa \nu < 1/2$ and $4\nu < \delta$.

Proposition 4.9. *Let x_0 be in $B_\nu(\zeta)$. One can define sequences $(x_i)_{i \geq 0}$ and $(w_i)_{i \geq 0}$ of elements of $\mathbb{E}$ such that $\|x_0 - w_0\| \leq \nu$ and, for $i \geq 0$:*

- *x_i is in $B_\delta(\zeta)$;*

- *$w_i = \Pi_{\mathcal{W}}(x_i)$;*

- *$x_i = \varphi(x_{i-1})$ if $i \geq 1$;*

- *$\|x_i - w_i\| \leq \kappa \|x_{i-1} - w_{i-1}\|^2$ if $i \geq 1$;*

- *$\|w_i - w_{i-1}\| \leq \kappa \|x_{i-1} - w_{i-1}\|^2$ if $i \geq 1$.*

Proof. We do a proof by induction; precisely, we prove that for all $i \geq 0$, one can construct $x_1, \ldots, x_i$ and $w_0, \ldots, w_i$ that satisfy the five items above.

For $i = 0$, the inequality $\|x_0 - \zeta\| \leq \delta$ follows from the fact that x_0 is in $B_\nu(\zeta)$ and that $\nu \leq \delta$. This implies that $w_0 = \Pi_{\mathcal{W}}(x_0)$ is well-defined; these are all the facts we need to prove for index 0. In what follows, we will also use the facts that $\|x_0 - w_0\| \leq \nu$, which holds since w_0 is the closest point to x_0 on $\mathcal{W}$, and hence $\|w_0 - \zeta\| \leq \|w_0 - x_0\| + \|x_0 - \zeta\| \leq 2\nu$.

Let us now assume that the claims hold up to index i, and prove that they still hold at index $i + 1$. Thus, $x_1, \ldots, x_i$ and $w_0, \ldots, w_i$ have been defined, and x_i is in $B_\delta(\zeta)$.

We set $x_{i+1} = \varphi(x_i)$; this is valid since x_i is in $B_\delta(\zeta)$. For the same reason, we can apply Proposition 4.8; we deduce that $\|x_{i+1} - w_i\| \leq K \|x_i - w_i\|^2$, that we can define $w_{i+1} = \Pi_{\mathcal{W}}(x_{i+1})$, and that $\|w_{i+1} - w_i\| \leq K' \|x_i - w_i\|^2$ holds. By the triangle inequality $\|x_{i+1} - w_{i+1}\| \leq \|x_{i+1} - w_i\| + \|w_i - w_{i+1}\|$, we get

$$
\|x_{i+1} - w_{i+1}\| \leq \kappa \|x_i - w_i\|^2
$$

and similarly

$$
\|w_{i+1} - w_i\| \leq K' \|x_i - w_i\|^2 \leq \kappa \|x_i - w_i\|^2.
$$

The only thing left to prove is that x_{i+1} is in $B_\delta(\zeta)$. To this effect, remark that we have (by an easy induction, and using the fact that $\kappa\nu < 1/2$)

$$\|x_j - w_j\| \leq \kappa^{2^j-1}\nu^{2^j} \leq \frac{\nu}{2^{2^j-1}}$$

for $0 \leq j \leq i+1$ and

$$\|w_{j+1} - w_j\| \leq \kappa^{2^{j+1}-1}\nu^{2^{j+1}} \leq \frac{\nu}{2^{2^{j+1}-1}}$$

for $0 \leq j \leq i$. We deduce

$$
\begin{aligned}
\|x_{i+1} - \zeta\| &\leq& \|x_{i+1} - w_{i+1}\| + \|w_{i+1} - w_i\| + \cdots + \|w_1 - w_0\| + \|w_0 - \zeta\| \\
&\leq& \frac{\nu}{2^{2^{i+1}-1}} + \sum_{j=0}^{i} \frac{\nu}{2^{2^{j+1}-1}} + 2\nu \\
&\leq& 2\nu\left(1 + \sum_{\ell\in\mathbb{N}} \frac{1}{2^\ell}\right) \\
&\leq& 4\nu \\
&<& \delta \quad \text{because } 4\nu < \delta.
\end{aligned}
$$

$\square$

Proof of Theorem 4.1. First, we prove that the sequence (w_i) is a Cauchy sequence. Assume that x_0 lies in the ball $B_\nu(\zeta)$. As a consequence of Proposition 4.9, we deduce by a simple induction (as we did during the proof of that proposition) that the following holds for all $i \geq 0$:

$$\|x_i - w_i\| \leq \frac{\nu}{2^{2^i-1}} \quad \text{and} \quad \|w_{i+1} - w_i\| \leq \frac{\nu}{2^{2^{i+1}-1}}. \tag{7}$$

We deduce in particular

$$\|x_i - w_i\| \leq \nu, \tag{8}$$

and this in turn allows us to prove (by induction on ℓ) that for all i, ℓ, the following holds:

$$\|x_{i+\ell} - w_{i+\ell}\| \leq \frac{\|x_i - w_i\|}{2^{2^\ell-1}}. \tag{9}$$

As a first consequence, we have, for all $k, \ell \in \mathbb{N}$, with $k \geq \ell$:

$$
\begin{aligned}
\|w_k - w_\ell\| &\leq& \sum_{i=0}^{k-\ell-1} \|w_{\ell+i+1} - w_{\ell+i}\| \\
&\leq& \sum_{i=0}^{\infty} \frac{\nu}{2^{2^{\ell+i+1}-1}} \quad \text{by Eq. (7)} \\
&\leq& \frac{\nu}{2^\ell}.
\end{aligned}
$$

Therefore, the sequence (w_i) is a Cauchy sequence; since $\lim_i \|x_i - w_i\| = 0$, both sequences (x_i) and (w_i) converge to a common limit x_∞. This proves the first claim in Theorem 4.1. Furthermore, we obtain the following estimates:

$$
\begin{aligned}
\|x_\infty - w_i\| &\leq \sum_{\ell \in \mathbb{N}} \|w_{i+\ell+1} - w_{i+\ell}\| \\
&\leq K' \sum_{\ell \in \mathbb{N}} \|x_{i+\ell} - w_{i+\ell}\|^2 \qquad \text{by Proposition 4.8} \\
&\leq K' \sum_{\ell \in \mathbb{N}} \frac{\|x_i - w_i\|^2}{2^{2^{\ell+1}-2}} \qquad \text{by Eq. (9)} \\
&\leq 2K' \|x_i - w_i\|^2 .
\end{aligned}
$$

In particular, for $i = 0$, we get the claim of the theorem that $\|x_\infty - \Pi_{\mathcal{W}}(x_0)\| \leq \gamma' \|x_0 - \Pi_{\mathcal{W}}(x_0)\|^2$, with $\gamma' = 2K'$. Besides, since $\|x_i - w_i\| \leq \nu$ and $2\kappa\nu < 1$, we also obtain, for any $i \geq 0$,

$$
\|x_\infty - w_i\| \leq 2K'\nu \|x_i - w_i\| \leq \|x_i - w_i\| . \tag{10}
$$

Finally, we prove that the convergence for the sequence (x_i) is quadratic. Note that since $\mathcal{W} \cap \overline{B_\rho(\zeta)}$ is closed, x_∞ is in $\mathcal{W}$ (as claimed in the theorem). In particular, $\|x_i - w_i\| \leq \|x_i - x_\infty\|$. We deduce, for $i \geq 0$,

$$
\begin{aligned}
\|x_{i+1} - x_\infty\| &\leq \|x_{i+1} - w_{i+1}\| + \|w_{i+1} - x_\infty\| \\
&\leq \|x_{i+1} - w_{i+1}\| + \|x_{i+1} - w_{i+1}\| \quad \text{by Eq. (10)} \\
&\leq 2\kappa \|x_i - w_i\|^2 \quad \text{using Proposition 4.9} \\
&\leq 2\kappa \|x_i - x_\infty\|^2 .
\end{aligned}
$$

This proves the last missing item from Theorem 4.1, with $\gamma = 2\kappa$.

Proof of Theorem 4.2. We prove that Φ is differentiable at ζ (which implies as a by-product its continuity around ζ) and that its derivative is $\Pi_{T_\zeta \mathcal{W}^0}$. Let $C'_{\mathcal{W}}$ denote the operator norm of the second derivative of $\Pi_{\mathcal{W}}$ at ζ, which is well defined since $\Pi_{\mathcal{W}}$ is of class C^2 in $B_\nu(\zeta)$.

Doing a first order expansion of $\Pi_{\mathcal{W}}$ between ζ and a point x in $B_\nu(\zeta)$, and using Theorem 4.1 and the facts that $\Pi_{\mathcal{W}}(\zeta) = \Phi(\zeta) = \zeta$ and $\|\Phi(x) - \Pi_{\mathcal{W}}(x)\| \leq \gamma' \|x - \Pi_{\mathcal{W}}(x)\|^2$ proved above, we get

$$
\begin{aligned}
\left\|\Phi(x) - \Phi(\zeta) - \Pi_{T_\zeta \mathcal{W}^0}(x - \zeta)\right\| &\leq \|\Phi(x) - \Pi_{\mathcal{W}}(x)\| + \left\|\Pi_{\mathcal{W}}(x) - \Pi_{\mathcal{W}}(\zeta) - \Pi_{T_\zeta \mathcal{W}^0}(x - \zeta)\right\| \\
&\leq \gamma' \|x - \Pi_{\mathcal{W}}(x)\|^2 + \frac{C'_{\mathcal{W}}}{2} \|x - \zeta\|^2 \\
&\leq (\gamma' + \frac{C'_{\mathcal{W}}}{2}) \|x - \zeta\|^2 \quad \text{since } \|x - \Pi_{\mathcal{W}}(x)\| \leq \|x - \zeta\|.
\end{aligned}
$$

Our claim, and thus Theorem 4.2, are proved.

5 Applications and experimental results

Our algorithm `NewtonSLRA` has been implemented in the `Maple` computer algebra system. In this section, we describe three applications of Structured Low-Rank Approximation (approximate GCD, matrix completion and approximate Hankel matrices) and compare our implementation to previous state-of-the-art. These experiments show that our all-purpose algorithm often performs as well as, or better than, existing solutions in a variety of settings.

All experiments have been conducted on a QUAD-core AMD Opteron 8384 2.7GHz.

5.1 Univariate approximate GCD

For $i \in \mathbb{N}$, let $\mathbb{R}[x]_i$ denote the vector space of polynomials with real coefficients of degree at most i. For $m, n, d \in \mathbb{N}$, let $G_{m,n,d} \subset \mathbb{R}[x]^2$ denote the set

$$G_{m,n,d} = \{(f, g) \in \mathbb{R}[x]_m \times \mathbb{R}[x]_n : \deg(\mathrm{GCD}(f, g)) = d\}.$$

We consider the Euclidean norm on $\mathbb{R}[x]_m$ and $\mathbb{R}[x]_m \times \mathbb{R}[x]_n$: if $f = \sum_{i=0}^m f_i x^i$ and $g = \sum_{i=0}^n g_i x^i$, then

$$\begin{aligned} \|f\| &= \sqrt{\sum_{i=0}^m f_i^2} \\ \|(f, g)\| &= \sqrt{\|f\|_m^2 + \|g\|_n^2} \end{aligned}$$

Problem 2 - Approximate GCD. Let $(f, g) \in \mathbb{R}[x]_m \times \mathbb{R}[x]_n$, $d \in \mathbb{N}$. Find $(f^*, g^*) \in \mathbb{R}[x]_m \times \mathbb{R}[x]_n$ such that $\deg(\mathrm{GCD}(f^*, g^*)) = d$ and $\|(f - f^*, g - g^*)\|$ is "small".

Similarly to SLRA, there are several variants of the approximate GCD problem. In some articles, the goal is to find a pair (f^*, g^*) which minimizes the distance $\|(f - f^*, g - g^*)\|$ (see e.g. [42] and references therein). In particular, [11] yields a certified quadratically convergent algorithm in the particular case $d = 1$ (i.e. the resultant of f^* and g^* vanishes). Sometimes, the goal is to find, if it exists, a ε-GCD, i.e. a pair (f^*, g^*) such that $\|(f - f^*, g - g^*)\| < \varepsilon$ and which have common roots for a given $\varepsilon > 0$, see e.g. [5, 14]. In some other contexts, the degree of the GCD is not known in advance and the goal is to maximize the degree $\deg(\mathrm{GCD}(f^*, g^*))$ provided that $\|(f - f^*, g - g^*)\| < \varepsilon$ for a given $\varepsilon > 0$ [20].

First, we recall the definition of the d-th Sylvester matrix of two univariate polynomials, which is rank-deficient if and only if $\deg(\mathrm{GCD}(f, g)) \geq d$.

Definition 5.1 (d-th Sylvester matrix)**.** *Let $(f, g) \in \mathbb{R}[x]_m \times \mathbb{R}[x]_n$ be univariate polynomials $f = \sum_{i=0}^m f_i x^i$, $g = \sum_{i=0}^n g_i x^i$. The d-th Sylvester matrix is the $(m+n-d+1) \times (m+n-2d+2)$*

matrix defined by

$$\mathrm{Syl}_d(f,g) = \left.\begin{bmatrix} f_m & 0 & \cdots & 0 & 0 & g_n & 0 & \cdots & 0 & 0 \\ f_{m-1} & f_m & \ddots & \vdots & \vdots & g_{n-1} & g_n & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & f_0 & f_1 & 0 & 0 & \cdots & g_0 & g_1 \\ 0 & 0 & \cdots & 0 & f_0 & 0 & 0 & \cdots & 0 & g_0 \end{bmatrix}\right\} n+m-d+1 .$$

$$\underbrace{\qquad\qquad}_{n-d+1} \quad \underbrace{\qquad\qquad}_{m-d+1}$$

It is well-known that $\deg(\mathrm{GCD}(f,g)) = d$ if and only if $\mathrm{rank}(\mathrm{Syl}_d(f,g)) = m+n-2d+1$ (see *e.g.* [28, Section 2]). Let then $\mathcal{D}_{m+n-2d+1}$ denote the determinantal variety of the $(m+n-d+1) \times (m+n-2d+2)$-matrices of rank $m+n-2d+1$. The following corollary is a direct consequence of this equivalence; it shows that Problem 2 is a particular case of SLRA.

Corollary 5.2. *By identifying* $\mathbb{R}[x]_m \times \mathbb{R}[x]_n$ *with the linear subspace* $\mathrm{Syl}_d(\mathbb{R}[x]_m \times \mathbb{R}[x]_n)$, *we have* $G_{m,n,d} = \mathrm{Syl}_d(\mathbb{R}[x]_m \times \mathbb{R}[x]_n) \cap \mathcal{D}_{m+n-2d+1}$.

Since approximate GCD is a particular case of SLRA, we now report experimental results which describe the behavior of our `Maple` implementation of `NewtonSLRA/1` in this context (for this application, we are searching for matrices of corank 1, so the first variant of `NewtonSLRA` has a better complexity). Given $m, n, d \in \mathbb{N}$ and $\varepsilon > 0$, we construct an instance of the approximate GCD problem as follows:

- we generate three polynomials $(\tilde{f}, \tilde{g}, \tilde{h}) \in \mathbb{R}[x]_{m-d} \times \mathbb{R}[x]_{n-d} \times \mathbb{R}[x]_d$, with all coefficients chosen uniformly at random in the interval $[-10, 10]$;

- we set $\mathfrak{f} = \tilde{f} \cdot \tilde{h} / \left\| (\tilde{f} \cdot \tilde{h}, \tilde{g} \cdot \tilde{h}) \right\|$ and $\mathfrak{g} = \tilde{g} \cdot \tilde{h} / \left\| (\tilde{f} \cdot \tilde{h}, \tilde{g} \cdot \tilde{h}) \right\|$, so that $\deg(\mathrm{GCD}(\mathfrak{f}, \mathfrak{g})) = d$ and $\|(\mathfrak{f}, \mathfrak{g})\| = 1$;

- we construct (f, g) by adding to each of the coefficients of $\mathfrak{f}$ and $\mathfrak{g}$ a noise sampled from a Gaussian distribution $\mathcal{N}(0, \varepsilon)$ of standard deviation ε.

In the sequel, we let $(f, g) \in \mathbb{R}[x]_m \times \mathbb{R}[x]_n$ denote the noisy data constructed as described above and $(f^*, g^*) \in \mathbb{R}[x]_m \times \mathbb{R}[x]_n$ denote the pair minimizing $\|(f - f^*, g - g^*)\|$ subject to $\deg(\mathrm{GCD}(f^*, g^*)) = d$.

In Table 1, we compare the steps' sizes of `NewtonSLRA` with those of `GPGCD`, a state-of-the art algorithm dedicated to the computation of approximate GCDs [42]. The experimental results give evidence of the practical quadratic convergence of `NewtonSLRA`, as predicted by Theorem 4.1. Experimental results for `GPGCD` seem to indicate linear convergence, but we would like to point out that `GPGCD` converges towards a solution of the optimization problem

	sizes of iteration steps	
iteration	`NewtonSLRA`	`GPGCD`
1	**0.42 10^{-3}**	0.20 10^{-2}
2	**0.19 10^{-5}**	0.30 10^{-3}
3	**0.11 10^{-9}**	0.15 10^{-4}
4	**0.43 10^{-18}**	0.68 10^{-6}
5	**0.10 10^{-34}**	0.17 10^{-8}

Table 1: Quadratic convergence of `NewtonSLRA`. The polynomials are randomly generated with $m = n = 25$, $d = 10$, and `Digits` $= 100$ in `Maple`.

(it finds the nearest pair of polynomials subject to the degree condition on the GCD) and hence returns a nearer approximation than `NewtonSLRA`.

Table 2 shows the experimental behavior of `NewtonSLRA` on a small example ($n = m = 10$, $r = 5$) with high-precision. Here the computation is stopped when the step size becomes smaller than 10^{-50} or after 50 iterations. The computations were performed with different values of ε with `Digits`$=120$ in `Maple` and each entry of the table is on average over 20 random instances. For $\varepsilon = 0.1$ or $\varepsilon = 1$, `GPGCD` did not converge within 50 iterations for most of the instances while `NewtonSLRA` converges within approximately 10 iterations. One iteration of `NewtonSLRA` is slightly slower than one iteration of `GPGCD` because of the cost of the singular value decomposition. Consequently, the range of problems where the quadratic convergence of `NewtonSLRA` yields efficiency improvements are SLRA problems where linearly convergent algorithms would require a lot of iterations.

The third column reports the distance between the output of `NewtonSLRA` and its input (the noisy pair of polynomials). Note that the squared distance between the initial exact data $(\mathfrak{f}, \mathfrak{g})$ and the noisy data (f, g) follow a χ^2 distribution with $n + m$ degrees of freedom. Therefore, the expected magnitude of the noise is $\mathbb{E}(\|(\mathfrak{f} - f, \mathfrak{g} - g)\|) = \sqrt{\varepsilon^2(n + m)} = \varepsilon\sqrt{n + m}$. In Table 2, $m = n = 10$ and hence the expected amplitude of the noise is $2\sqrt{5}\varepsilon$. All the entries in the third column are below this value, which indicates that on average, the output of `NewtonSLRA` is actually a better approximation of the noisy data than the initial exact data $(\mathfrak{f}, \mathfrak{g})$. Consequently, the quality of the solution returned by `NewtonSLRA` should be sufficient for many applications even though it does not solve the associated minimization problem. The last column of Table 2 indicates the distance between (f', g'), the output of `NewtonSLRA` and the nearest solution (f^*, g^*). As predicted by Theorem 4.1, the distance to the nearest solution appears to be quadratic in the magnitude ε of the noise.

In order to estimate (f^*, g^*), we use the linearly convergent certified Gauss-Newton iteration in [45], using as the starting point of the iteration the pair $(\mathfrak{f}, \mathfrak{g})$. Note that using directly the Gauss-Newton approach for the approximate GCD problem requires a good starting point: in applicative situations, the pair $(\mathfrak{f}, \mathfrak{g})$ is unknown and therefore finding such a good pair with a high degree gcd is a difficult problem.

We would like to point out that `NewtonSLRA` is also able to solve larger problems: for instance, it can compute approximate GCDs for $m = n = 2000$ and $d = 1000$ within a few min-

	Nb. iterations			
ε	NewtonSLRA	GPGCD	$\|(f' - f, g' - g)\|$	$\|(f' - f^*, g' - g^*)\|$
10^{-10}	4.0	6.0	$1.86\ 10^{-10}$	$3.12\ 10^{-19}$
10^{-9}	4.0	6.6	$1.93\ 10^{-9}$	$2.98\ 10^{-17}$
10^{-8}	4.0	7.2	$2.01\ 10^{-8}$	$3.16\ 10^{-15}$
10^{-7}	4.9	8.7	$2.06\ 10^{-7}$	$3.25\ 10^{-13}$
10^{-6}	5.0	10.0	$1.62\ 10^{-6}$	$5.45\ 10^{-11}$
10^{-5}	5.1	11.9	$1.53\ 10^{-5}$	$1.15\ 10^{-9}$
10^{-4}	5.6	15.4	$1.82\ 10^{-4}$	$1.99\ 10^{-7}$
10^{-3}	6.3	24.4	$1.76\ 10^{-3}$	$1.96\ 10^{-5}$
10^{-2}	7.1	37.1	$1.87\ 10^{-2}$	$3.26\ 10^{-3}$
10^{-1}	8.7	49.2	$1.43\ 10^{-1}$	$6.94\ 10^{-2}$
10^{0}	11.0	50	$2.42\ 10^{-1}$	$1.71\ 10^{-1}$

Table 2: Experimental convergence of `NewtonSLRA`. The pair (f, g) is the input polynomials, (f', g') is the output of `NewtonSLRA`, and (f^*, g^*) is the optimal solution (the polynomials minimizing $\|f - f^*, g - g^*\|$ under the constraint $\deg(\mathrm{GCD}(f^*, g^*)) = d$). The polynomials are randomly generated with $m = n = 10$, $d = 5$, and `Digits` $= 120$ in `Maple`. The iteration is stopped when the step size becomes smaller that 10^{-50} or after 50 iterations

utes (for the default numerical precision of Maple: `Digits=10`). In order to demonstrate the efficiency of our approach, we compare in Table 3 timings obtained with our implementation of `NewtonSLRA` and with the software `uvGCD` [46]. We observe in this table that `NewtonSLRA` runs faster than `uvGCD`; the quality of the output (*i.e.* the value $\|(f_{output} - f, g_{output} - g)\|$) of `NewtonSLRA` is comparable to that of `uvGCD`.

5.2 Low-rank matrix completion

Matrix completion is a problem arising in several applications in Engineering Sciences, and plays an important role in the recent development of compressed sensing. Knowing some properties of a matrix (*e.g.* its rank), the goal is to recover it by looking only at a subset of its entries. We focus here on low-rank matrix completion which can be modeled by structured low-rank approximation: let I be a subset of $\{1, \ldots, p\} \times \{1, \ldots, q\}$ and $A = (a_{i,j})_{a_{i,j} \in \mathbb{R}, (i,j) \in I}$. We consider the affine space $E \subset \mathcal{M}_{p,q}(\mathbb{R})$ of all matrices $(M_{i,j})$ such that, for $(i, j) \in I$, $M_{i,j} = a_{i,j}$. Low-rank matrix completion is a SLRA problem since it asks to find a matrix in $E \cap \mathcal{D}_r$.

One particular case of interest for applications is when there is a unique solution to the matrix completion problem. In that case, $(p - r)(q - r) > \dim(E)$. Consequently, the transversality condition required for the analysis performed in Section 4 does not hold. Therefore, the results in this section are mainly experimental observations.

Efficient techniques have been developped to tackle the matrix completion problem via a convex relaxation (see [8, 10, 9, 37] and references therein). In this section, we report

	time (in s)		$\|(f_{output} - f, g_{output} - g)\|$	
(m, n, d)	NewtonSLRA	uvGCD	NewtonSLRA	uvGCD
(20,20,10)	0.0264	0.1876	0.0003034353	0.0003034150
(40,40,20)	0.0552	0.6185	0.0005466551	0.0004171554
(60,60,30)	0.1925000	1.670900	0.0005305	0.0005248752
(80,80,40)	0.3870000	3.277600	0.0006652485	0.0006573120
(100,100,50)	0.4288000	5.221100	0.0008292970	0.0007893376
(120,120,60)	0.5922000	8.987600	0.0008901396	0.0007972352
(140,140,70)	0.8618000	12.58410	0.0009151193	0.0008635334
(160,160,80)	1.040300	16.84990	0.0009183804	0.001195548
(180,180,90)	1.510300	24.01900	0.0009902834	0.001812256
(200,200,100)	1.601800	29.00110	0.001041346	0.001032610
(220,220,110)	1.970000	39.47140	0.002613709	0.001061010
(240,240,120)	2.363400	49.85650	0.001227303	0.001083454
(260,260,130)	2.771200	61.15920	0.001224906	0.001153301
(280,280,140)	3.419700	73.69030	0.003155242	0.004107356
(300,300,150)	3.082400	86.92640	0.001296697	0.003963097

Table 3: Comparison with uvGCD. For both of the software, (f, g) is the pair of input polynomials, (f_{output}, g_{output}) is the output pair.

experimental results which indicate that Algorithm NewtonSLRA can be used to solve families of low-rank matrix completion problems which cannot be solved by the convex relaxation. Moreover, we give timings which seem to indicate that the computational complexity of NewtonSLRA is of the same order of magnitude as that of convex optimization techniques.

We follow [9, Section 7] for the generation of instances of the matrix completion problem:

- for $r \in \{1, \ldots, p\}$, we generate a $p \times p$ matrix $M = L \cdot R$ of rank r by sampling two matrices $L \in \mathcal{M}_{p,r}(\mathbb{R})$ and $R \in \mathcal{M}_{r,p}(\mathbb{R})$ whose entries follow i.i.d. Gaussian distributions $\mathcal{N}(0, 1)$;

- we uncover m entries at random in the matrix by sampling a subset $I \subset \{1, \ldots, p\} \times \{1, \ldots, q\}$ of cardinality m uniformly at random;

- the affine space E is the set of matrices $X = (X_{i,j})$ such that $X_{i,j} = M_{i,j}$ if $(i, j) \in I$.

Then we run NewtonSLRA by setting as the starting point of the iteration the matrix $N = (N_{i,j}) \in E$ defined by

$$\begin{cases} N_{i,j} = M_{i,j} & \text{if } (i, j) \in I \\ N_{i,j} = 0 & \text{otherwise} \end{cases}$$

and we stop iterating NewtonSLRA when the size of an iteration becomes smaller than 10^{-4} or after 100 iterations. We consider the problem solved if NewtonSLRA returns a matrix $\hat{M}$ such that
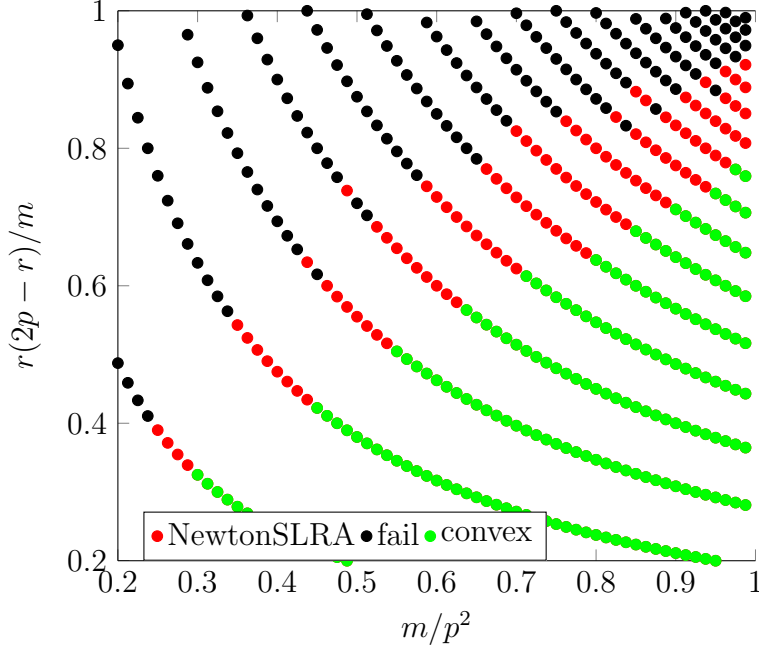
$$\left\| \hat{M} - M \right\| / \|M\| < 10^{-3}$$

Figure 4: Performances of `NewtonSLRA` for low-rank matrix completion

for more than for 75% of randomly generated instances.

Figure 4 reports experimental results for $n = 40$, and should be compared with [9, Figure 1]. Green dots correspond to instances that can be solved by convex methods (green dots correspond to the white/grey area in [9, Figure 1]). Any instance that could be solved by the convex relaxation presented in [9] is also solved by `NewtonSLRA`. Red dots correspond to parameters where the matrix can be completed by `NewtonSLRA` but not by the convex relaxation. Black dots correspond to problems which are not solved by any of these methods. This figure indicates that `NewtonSLRA` extends the range of matrix completion problems that could be treated by convex relaxation.

Timings given in [38] indicate that the semidefinite program obtained via the convex relaxation is solved in approximately 2 minutes on a 2GHz laptop (for the instances that can be solved by this method: the green dots in Figure 4). For `NewtonSLRA`, the timings for solving these instances range between 0.8 seconds and 34 seconds seconds on a QUAD-core Intel i5-3570 3.4GHz.

We also compare our implementation of `NewtonSLRA` with a state-of-the-art Matlab software of Riemannian optimization developed by B. Vandereycken. Figure 5 shows the convergence properties of these two algorithms on an example of matrix completion of a 100×100 matrix of rank 5 where 1950 samples have been observed. The graph shows the fast convergence of `NewtonSLRA` at each iteration and suggests quadratic convergence. The precision is capped at 2^{-48} (the size of the mantissa of a double float) in order to use BLAS routines to compute efficiently the SVD. In practice, the Riemannian optimization software is faster than `NewtonSLRA`, even though it requires more iterations. For the example described in
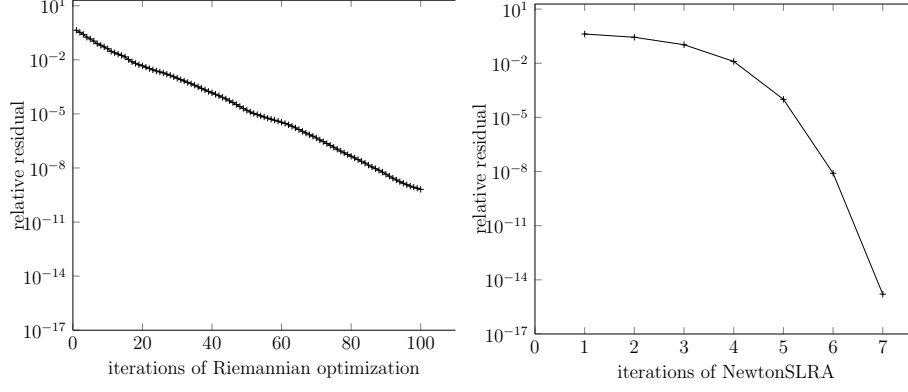
Figure 5: Comparison of Riemannian optimization and NewtonSLRA. The relative residual is the value $\|P_\Omega(M_{input} - M_i)\| / \|P_\Omega(M_{input})\|$, where P_Ω is the orthogonal projection on the linear space of matrices having zeroes outside the set of observed entries [43]. During NewtonSLRA, the relative residual is measured after the computation of the SVD (matrix $\widetilde{M}$ in the pseudocode).

Figure 5, the total running time of the Riemannian optimization software is 0.1s, whereas the total running total running time of NewtonSLRA is 45s. Also, NewtonSLRA is restricted in practice to small matrix sizes and do not apply to large-scale matrix completion problems. Consequently, the fast convergence of NewtonSLRA may yield improvements in applications where we require a very precise completion of a small size matrix. Also, we would like to point out that NewtonSLRA also experimentally converges when the input matrix is slightly noisy, but this phenomenon is beyond the scope of this paper and is not explained by the theoretical convergence analysis in Section 4.

5.3   Low-rank approximation of Hankel matrices

In this section, we finally compare the performances of NewtonSLRA with the STLN approach for Low-Rank Approximation of Hankel matrices proposed in [36]. Let us recall briefly the experimental setting described in [36, Section 4.2] for 7×5 Hankel matrices.

Let H_c be the following rank 4 Hankel matrix:

$$H_c = \begin{bmatrix} \nu_1 & \nu_2 & \ldots & \nu_5 \\ \nu_2 & \nu_3 & \ldots & \nu_6 \\ \vdots & \vdots & \vdots & \vdots \\ \nu_7 & \nu_8 & \ldots & \nu_{11} \end{bmatrix},$$

where $\nu_i = \sum_{\ell=1}^4 \beta_\ell z_\ell^i$, with $\beta = (1, 2, 1/2, 3/2)$, $z = (\exp(-0.1), \exp(-0.2), \exp(-0.3), \exp(-0.35))$.

The perturbed matrix is $H = H_c + \tau\Delta$, where $\tau > 0$ and Δ is a Hankel matrix with entries picked uniformly at random in the interval $[0, 1]$.

In Table 4, we report the number of iterations needed to obtain a rank 4 approximation of H with several algorithms. As in [36], we stop iterating as soon as the smallest singular

τ	STLN1	STLN2	Cadzow	NewtonSLRA
10^{-8}	1.1	1.7	59.8	**2.4**
10^{-7}	1.6	2.3	75.3	**3.4**
10^{-6}	2.2	2.2	83.0	**3.9**
10^{-5}	2.1	3.2	92.4	**3.8**
10^{-4}	2.1	3.9	93.3	**4.0**
10^{-3}	4.0	6.8	100*	**4.1**
10^{-2}	4.5	20.5	100*	**4.2**
10^{-1}	6.9	22.6	100*	**4.2**

Table 4: Number of iterations required by several algorithms to converge towards a rank 4 Hankel matrix. Each entry in the last column is the average of 30 test results. The three first columns recall the experimental results in [36, Table 4.1]. 100* means that the algorithm did not converge within 100 iterations.

τ	STLN1	STLN2	Cadzow	NewtonSLRA
10^{-8}	8	100*	95	4
10^{-7}	8	100*	100*	4
10^{-6}	8	100*	90	4
10^{-5}	8	100*	95	4
10^{-4}	8	100*	99	4
10^{-3}	6	100*	100*	4
10^{-2}	20	100*	100*	**4.1**
10^{-1}	10	100*	100*	**4.4**

Table 5: Number of iterations required by several algorithms to converge towards a rank 4 Hankel matrix in presence of an outlier on the 8th antidiagonal. Each entry in the last column is the average of 30 test results. The three first columns recall the experimental results in [36, Table 4.2].

value becomes less than 10^{-14}. The number of iterations of NewtonSLRA becomes smaller than for STLN when the magnitude of the noise becomes larger.

Another setting which is important for practical applications is the behavior of the algorithm in the presence of an outlier, *i.e.* when one measure is very imprecise compared to the other measures. To investigate this case, we follow the experimental setting in [36, Table 4.2]: we generate Hankel matrices as above, but then we add 0.01 to all entries on the 8th antidiagonal. Experiments seem to indicate that `NewtonSLRA` also behaves well in the presence of such an outlier, as shown by the number of iterations that we report in Table 5. Also, each of these low-rank approximations of Hankel matrices (with and without an outlier) were computed in less than 0.6 seconds with `NewtonSLRA`.

References

[1] P.-A. Absil, L. Amodei, and G. Meyer. Two Newton methods on the manifold of fixed-rank matrices endowed with Riemannian quotient geometries. *arXiv:1209.0068 e-print*, 2012.

[2] E.L. Allgower and K. Georg. *Numerical continuation methods*, volume 13. Springer-Verlag Berlin, 1990.

[3] E. Arbarello, M. Cornalba, P. Griffiths, and J. Harris. *Geometry of algebraic curves I*, volume 268. Springer, 1984.

[4] A. Ben-Israel. A modified Newton-Raphson method for the solution of systems of equations. *Israel Journal of Mathematics*, 3(2):94–98, 1965.

[5] D.A. Bini and P. Boito. Structured matrix-based methods for polynomial-GCD: analysis and comparisons. In *Proceedings of the 2007 international symposium on Symbolic and algebraic computation*, pages 9–16. ACM, 2007.

[6] W. Bruns and U. Vetter. *Determinantal Rings*. Springer, 1988.

[7] J.A. Cadzow. Signal enhancement-a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(1):49–62, 1988.

[8] E.J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[9] E.J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[10] E.J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.

[11] G. Chèze, J.-C. Yakoubsohn, A. Galligo, and B. Mourrain. Computing nearest GCD with certification. In *Proceedings of the 2009 conference on Symbolic numeric computation*, pages 29–34. ACM, 2009.

[12] M.T. Chu, R.E. Funderlic, and R.J. Plemmons. Structured Low Rank Approximation. *Linear algebra and its applications*, 366:157–172, 2003.

[13] L. Condat and A. Hirabayashi. Cadzow denoising upgraded: A new projection method for the recovery of Dirac pulses from noisy linear measurements. preprint, 2012.

[14] R.M. Corless, S.M. Watt, and L. Zhi. QR factoring to compute the GCD of univariate approximate polynomials. *IEEE Transactions on Signal Processing*, 52(12):3394–3402, 2004.

[15] J.-P. Dedieu. *Points fixes, zéros et la méthode de Newton*, volume 54. Springer, 2006.

[16] J.-P. Dedieu and M.-H. Kim. Newton's method for analytic systems of equations with constant rank derivatives. *Journal of Complexity*, 18(1):187–209, 2002.

[17] F. Deutsch. *Best approximation in inner product spaces.* Springer, 2001.

[18] J. Draisma, E. Horobet, G. Ottaviani, B. Sturmfels, and R. R. Thomas. The Euclidean distance degree of an algebraic variety. *ArXiv:1309.0049 e-print*, 2013.

[19] D. Eisenbud. Linear sections of determinantal varieties. *American Journal of Mathematics*, 110(3):541–575, 1988.

[20] I.Z. Emiris, A. Galligo, and H. Lombardi. Certified approximate univariate GCDs. *Journal of Pure and Applied Algebra*, 117:229–251, 1997.

[21] K. Friedrichs. On certain inequalities and characteristic value problems for analytic functions and for functions of two variables. *Transactions of the American Mathematical Society*, 41(3):321–364, 1937.

[22] S. Gao, E. Kaltofen, J. May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials via differential equations. In *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, pages 167–174. ACM, 2004.

[23] M. Golubitsky and V. Guillemin. *Stable mappings and their singularities*, volume 314. Springer-Verlag New York, 1973.

[24] L. Hogben, editor. *Handbook of Linear Algebra.* Discrete Mathematics and Its Applications. Taylor & Francis, 2006.

[25] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. *ArXiv:1212.0467 e-print*, 2012.

[26] E. Kaltofen, J.P. May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials using singular value decomposition. *Journal of Symbolic Computation*, 43(5):359–376, 2008.

[27] E. Kaltofen, Z. Yang, and L. Zhi. Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials. In *Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation*, pages 169–176. ACM, 2006.

[28] E. Kaltofen, Z. Yang, and L. Zhi. Structured low rank approximation of a Sylvester matrix. In *Symbolic-numeric computation*, pages 69–83. Springer, 2007.

[29] N. Karmarkar and Y.N. Lakshman. Approximate polynomial greatest common divisors and nearest singular polynomials. In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, pages 35–39. ACM, 1996.

[30] N.K. Karmarkar and Y.N. Lakshman. On approximate GCDs of univariate polynomials. *Journal of Symbolic Computation*, 26(6):653–666, 1998.

[31] A. S. Lewis and J. Malick. Alternating projections on manifolds. *Mathematics of Operations Research*, 33(1):216–234, 2008.

[32] B. Li, Z. Yang, and L. Zhi. Fast low rank approximation of a Sylvester matrix by structured total least norm. *J. Japan Soc. Symbolic and Algebraic Comp*, 11:165–174, 2005.

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

[34] G. Ottaviani, P.-J. Spaenlehauer, and B. Sturmfels. Algebraic methods for structured low-rank approximation. *ArXiv:1311.2376 e-print*, 2013.

[35] V. Pan. Computation of approximate polynomial GCDs and an extension. *Information and Computation*, 167(2):71–85, 2001.

[36] H. Park, L. Zhang, and J.B. Rosen. Low rank approximation of a Hankel matrix by structured total least norm. *BIT Numerical Mathematics*, 39(4):757–779, 1999.

[37] B. Recht. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, pages 3413–3430, 2011.

[38] B. Recht, W. Xu, and B. Hassibi. Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization. In *47th IEEE Conference on Decision and Control, 2008.*, pages 3065–3070. IEEE, 2008.

[39] J.B. Rosen, H. Park, and J. Glick. Structured total least norm for nonlinear problems. *SIAM Journal on Matrix Analysis and Applications*, 20(1):14–30, 1998.

[40] W.M. Ruppert. Reducibility of polynomials $f(x, y)$ modulo p. *Journal of Number Theory*, 77:62–70, 1999.

[41] A. Schönhage. Quasi-GCD computations. *Journal of Complexity*, 1(1):118–137, 1985.

[42] A. Terui. An iterative method for calculating approximate GCD of univariate polynomials. In *Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation*, pages 351–358. ACM, 2009.

[43] B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 2013. Accepted.

[44] J. Winkler and J. Allan. Structured low rank approximations of the Sylvester resultant matrix for approximate GCDs of Bernstein basis polynomials. *Electronic Transactions on Numerical Analysis*, 31:141–155, 2008.

[45] J.-C. Yakoubsohn, M. Masmoudi, G. Cheze, and D. Auroux. Approximate GCD a la Dedieu. *Applied Mathematics E-Notes*, 11:244–248, 2011.

[46] Z. Zeng and B.H. Dayton. The approximate GCD of inexact polynomials. In *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, pages 320–327. ACM, 2004.

Authors' addresses:

Éric Schost, Western University, Department of Computer Science, Middlesex College; Ontario N6A 3K7, Canada. `eschost@uwo.ca`

Pierre-Jean Spaenlehauer, CARAMEL project, INRIA Grand-Est; LORIA, Campus scientifique, BP239, 54506 Vandoeuvre-lès-Nancy, France, `pierre-jean.spaenlehauer@.inria.fr`