Riemannian Newton optimization methods for the symmetric tensor approximation problem

Rima Khouja, Houssam Khalil, Bernard Mourrain

To cite this version:

Rima Khouja, Houssam Khalil, Bernard Mourrain. Riemannian Newton optimization methods for the symmetric tensor approximation problem. Linear Algebra and its Applications, 2022, 637C, pp.175-211. 10.1016/j.laa.2021.12.008 . hal-02494172v3

Riemannian Newton optimization methods for the symmetric tensor approximation problem

Rima Khouja[a,b,*], Houssam Khalil[b], Bernard Mourrain[a]

[a]Inria d'Université Côte d'Azur, Aromath, Sophia Antipolis, France
[b]Laboratory of Mathematics and its Applications LaMa-Lebanon, Lebanese University, Lebanon

Abstract

The Symmetric Tensor Approximation problem (STA) consists of approximating a symmetric tensor or a homogeneous polynomial by a linear combination of symmetric rank-1 tensors or powers of linear forms of low symmetric rank. We present two new Riemannian Newton-type methods for low rank approximation of symmetric tensor with complex coefficients.

The first method uses the parametrization of the set of tensors of rank at most r by weights and unit vectors. Exploiting the properties of the apolar product on homogeneous polynomials combined with efficient tools from complex optimization, we provide an explicit and tractable formulation of the Riemannian gradient and Hessian, leading to Newton iterations with local quadratic convergence. We prove that under some regularity conditions on non-defective tensors in the neighborhood of the initial point, the Newton iteration (completed with a trust-region scheme) is converging to a local minimum.

The second method is a Riemannian Gauss–Newton method on the Cartesian product of Veronese manifolds. An explicit orthonormal basis of the tangent space of this Riemannian manifold is described. We deduce the Riemannian gradient and the Gauss–Newton approximation of the Riemannian Hessian. We present a new retraction operator on the Veronese manifold.

We analyze the numerical behavior of these methods, with an initial point provided by Simultaneous Matrix Diagonalisation (SMD). Numerical experiments show the good numerical behavior of the two methods in different cases and in comparison with existing state-of-the-art methods.

Keywords: symmetric tensor decomposition, homogeneous polynomials, Riemannian optimization, Newton method, retraction, complex optimization, trust region method, Veronese manifold.
MSC: 15A69, 15A18, 53B20, 53B21, 14P10, 65K10, 65Y20, 90-08

1. Introduction

A symmetric tensor T of order d and dimension n in $\mathcal{T}^d(\mathbb{C}^n) = \mathbb{C}^n \otimes \cdots \otimes \mathbb{C}^n = \mathcal{T}_n^d$ is a special case of tensors, where its entries do not change under any permutation of its d indices. We denote their set by $\mathcal{S}^d(\mathbb{C}^n) = \mathcal{S}_n^d$. The symmetric tensor decomposition problem consists of

[*] Corresponding author
Email address: rima.khouja@inria.fr (Rima Khouja)

decomposing a symmetric tensor $T \in \mathcal{S}_n^d$ into a linear combination of symmetric tensors of rank one i.e.

$$T = \sum_{i=1}^{r} w_i \underbrace{v_i \otimes ... \otimes v_i}_{d \text{ times}}, \quad w_i \in \mathbb{C},\ v_i \in \mathbb{C}^n \tag{1}$$

For a multilinear tensor, its minimal decomposition as a sum of tensor products of vectors is called the Canonical Polyadic Decomposition [1]. We have a correspondence between $\mathcal{S}_n^d$ and the set of homogeneous polynomials of degree d in n variables denoted $\mathbb{C}[x_1, \dots, x_n]_d =: \mathbb{C}[\mathbf{x}]_d$. Using this correspondence, (1) is equivalent to express the homogeneous polynomial $\mathbf{p}$ associated to T as a sum of powers of linear forms, which is by definition the classical Waring decomposition i.e.

$$\mathbf{p} = \sum_{i=1}^{r} w_i (v_{i,1} x_1 + \cdots + v_{i,n} x_n)^d, \quad w_i \in \mathbb{C},\ v_i \in \mathbb{C}^n \tag{2}$$

The smallest r such that this decomposition exists is by definition the symmetric rank of $\mathbf{p}$ denoted by $\text{rank}_s(\mathbf{p})$. Let $d \geq 3$. The generic symmetric rank denoted by r_g, is given by the Alexander–Hirschowitz theorem [2] as follows: $r_g = \left\lceil \frac{1}{n}\binom{n+d-1}{d} \right\rceil$ for all $n, d \in \mathbb{N}$, except for the following cases: $(d, n) \in \{(3, 5), (4, 3), (4, 4), (4, 5)\}$, where it should be increased by 1. We say that T is of subgeneric rank, if its rank $\text{rank}_s(T) = r$ in (2) is strictly lower than r_g. In this case, a strong property of uniqueness of the Waring decomposition holds [3], and the symmetric tensor T is called identifiable, unless in three exceptions which are cited in [3, Theorem 1.1], where there are exactly two Waring decompositions. This identifiability property forms an important key strength of the Waring or equivalently the symmetric tensor decomposition. It can explain why this decomposition problem appears in many applications for instance in the areas of mobile communications, in blind identification of under-determined mixtures, machine learning, factor analysis of k-way arrays, statistics, biomedical engineering, psychometrics, and chemometrics. See e.g. [4, 5, 6, 7, 8] and references therein. The decomposition of the tensor is often used to recover structural information in the application problem.

The Symmetric Tensor Approximation problem (STA) consists of finding the closest symmetric tensor to a given symmetric tensor $T \in \mathcal{S}_n^d$, of low symmetric rank. Equivalently, for a given $r \in \mathbb{N}^*$, it consists of approximating a homogeneous polynomial $\mathbf{p}$ associated to a symmetric tensor T by an element in Σ_r, where $\Sigma_r = \{\mathbf{q} \in \mathbb{C}[\mathbf{x}]_d \mid \text{rank}_s(\mathbf{q}) \leq r\}$, i.e.

$$\text{(STA)} \quad \min_{\mathbf{q} \in \Sigma_r} \tfrac{1}{2} ||\mathbf{p} - \mathbf{q}||_d^2.$$

Since in many problems, the input tensors are often computed from measurements or statistics, they are known with some errors on their coefficients and computing an approximate decomposition of low rank often gives better structural information than the exact or accurate decomposition of the approximate tensor [9, 10, 11].

For matrices, the best low rank approximation can be computed via Singular Value Decomposition (SVD). Higher Order Singular Value Decomposition (HOSVD) has been investigated to compute a multilinear rank approximation of a tensor [6, 12, 13], this, in contrast to the matrix case, does not give the best multilinear rank approximation (see for instance inequality (5) in [14]).

A classical approach for computing an approximate tensor decomposition of low rank is the so-called Alternating Least Squares (ALS) method. It consists of minimizing the distance between a given tensor and a low rank tensor by alternately updating the different factors of the tensor decomposition, solving a quadratic minimization problem at each step. See e.g. [15, 16, 17, 18]. This approach is well-suited for tensor represented in $\mathcal{T}_n^d$ but it looses the symmetry property in the internal steps of the algorithm. The space in which the linear operations are performed is of large dimension n^d compared to the dimension $\binom{n+d-1}{d}$ of $\mathcal{S}_n^d$ when n and d grow. Moreover the convergence is slow [19, 20].

Other iterative methods such as quasi-Newton methods have been considered for low rank tensor approximation problems to improve the convergence speed. See for instance [21, 22, 23, 24, 25, 26]. A Riemannian Gauss–Newton algorithm with trust region scheme was presented in [27], to approximate a given real multilinear tensor by one of low rank. The Riemannian optimization set is a Cartesian product of Segre manifolds (i.e. manifolds of real multilinear tensors of rank one). The retraction on the Segre manifold, called ST-HOSVD, is based on sequentially truncated HOSVD [14, 28, 13]. Moreover, an algorithm, called hot restarts, was introduced in [27] to avoid ill-conditioned decompositions. Closely related to these iterative methods, the condition number of join decompositions such as tensor decompositions is studied in [29].

Optimization techniques based on quasi-Newton iterations for block term decompositions of multilinear tensors over the complex numbers have also been presented in [30, 25]. In [24] quasi-Newton and limited memory quasi-Newton methods for distance optimization on products of Grassmannian manifolds are designed to deal with the Tucker decomposition of a tensor and applied for a low multilinear rank tensor approximation. In all these approaches, an approximation of the Hessian is used to compute the descent direction, and the local quadratic convergence cannot be guaranteed.

Specific investigations have been developed, in the case of best rank-1 approximation. The problem is equivalent to the optimization of a polynomial on the product of unitary spheres (see e.g. [12, 31]). Global polynomial optimization methods can be employed over the real or complex numbers, using for instance convex relaxations and semidefinite programming [32]. However, the approach is facing scalability issues in practice for large size tensors.

In relation with polynomial representation and multivariate Hankel matrix properties, another least square optimization problem is presented in [33], for low rank symmetric tensor approximation. Good approximations of the low rank approximation are obtained for small enough perturbations of low rank tensors. More recently, a method for decomposing real even-order symmetric tensors, called Subspace Power Method (SPM), has been proposed in [34]. It is based on a power method associated to the projection on subspaces of eigenvectors of the Hankel operators and has a linear convergence.

1.1. Contributions

In this paper, we present two new Riemannian Newton-type methods for the low rank approximation problem (STA) for symmetric tensors with complex coefficients.

The first method uses the parametrization of the set of tensors of rank at most r by weights and vectors on the unit sphere. Exploiting the properties of the apolar product on homogeneous polynomials combined with efficient tools from complex optimization, we provide an explicit and tractable formulation of the Riemannian gradient and Hessian, leading to Newton iterations with local quadratic convergence. We prove that under some regularity conditions on non-defective tensors in the neighborhood of the initial point, the iteration (completed with a trust-region scheme) is converging to a local minimum.

The second method is a Riemannian Gauss–Newton method on the Cartesian product of Veronese manifolds. We describe an explicit orthonormal basis of the tangent space of this Riemannian manifold. We use this basis to obtain the Riemannian gradient and the Gauss–Newton approximation of the Riemannian Hessian. We present an approximation method for a given homogeneous polynomial in $\mathbb{C}[\mathbf{x}]_d$ into linear form to the d^{th} power, based on the rank-1 truncation of the SVD of Hankel matrix associated to the homogeneous polynomial. From this approximation method, we propose a new retraction operator on the Veronese manifold. We point out that the Riemannian Gauss–Newton iteration on the Cartesian product of Segre manifolds presented in [27] is related to our approach in the sense that the design of the algorithm depends mainly on the geometry of the targeted manifold (Segre manifold in the multilinear case, Veronese manifold in the symmetric case) and its tangent space. In this context, the Riemannian Gauss–Newton iteration that we describe is adapted to the symmetric setting by considering the reduced vector space $\mathbb{C}[\mathbf{x}]_d$ and by exploiting the apolar identities. In our approach we consider symmetric tensors with complex coefficients. The constraint set is parameterized via the complex Veronese manifolds, which leads us to a complex optimization problem with geometric constraints, and this, to the best of our knowledge, has not been addressed previously in tensor approximation.

We analyze the numerical behavior of these methods, choosing for the initial point the approximate decomposition provided by the Simultaneous Matrix Diagonalisation (SMD) of a pencil of Hankel matrices [35, 36]. Numerical experiments show the good numerical behavior of the new methods for the best rank-1 approximation of real-valued symmetric tensors, for low rank approximation of sparse symmetric tensors, and against perturbations of symmetric tensors of low rank. Comparisons with existing state-of-the-art methods corroborate this analysis.

1.2. Outline

The paper is structured as follows. In section 2 we give the main notation and preliminaries. In section 3, we describe the set of non-defective rank-r symmetric tensors. In subsection 4.1, we formulate the STA problem as a Riemannian least square optimization problem using the parametrization by weights and unit vectors. We compute explicitly the Riemannian gradient vector and the Hessian matrix in subsection 4.1.1 (the proofs are in Appendix A) and describe the retraction in subsection 4.1.2. In subsection 4.2, we describe the Riemannian Gauss–Newton method on the product of Veronese manifolds. We present in subsection 4.2.1 a new retraction operator on the Veronese manifold with its analysis. In subsection 4.3, we recall the trust-region extension scheme, and prove under some regularity assumptions the convergence of the exact Riemannian Newton method with trust region steps to a local minimum of the distance function. Numerical experiments are featured in section 5. The final section is for our conclusions and outlook.

2. Notation and preliminaries

We use similar notation as in [37]. We denote by $\mathcal{T}_n^d = \mathcal{T}^d(\mathbb{C}^n) = \mathbb{C}^n \otimes \cdots \otimes \mathbb{C}^n$ the outer product d times of $\mathbb{C}^n$. The set of symmetric tensors in $\mathcal{T}_n^d$ is denoted $\mathcal{S}_n^d$. We have a correspondence between $\mathcal{S}_n^d$ and the set of the homogeneous polynomials of degree d in n variables $\mathbb{C}[x_1, \ldots, x_n]_d := \mathbb{C}[\mathbf{x}]_d$. This allows to reduce the dimension of the ambient space of the problem from n^d (dimension of $\mathcal{T}_n^d$) to $s_{n,d} := \binom{n+d-1}{d}$ (dimension of $\mathcal{S}_n^d \sim \mathbb{C}[\mathbf{x}]_d$). The bold letters $\mathbf{p}, \mathbf{q}$ denote homogeneous polynomials in $\mathbb{C}[\mathbf{x}]_d$ or equivalently elements in

$\mathcal{S}_n^d$. A homogeneous polynomial $\mathbf{p}$ in $\mathbb{C}[\mathbf{x}]_d$ can be written as: $\mathbf{p} = \sum_{|\alpha|=d} \binom{d}{\alpha} p_\alpha \mathbf{x}^\alpha$, where $\mathbf{x} := (x_1, \ldots, x_n)$ is the vector of the variables $x_1, \ldots, x_n$, $\alpha = (\alpha_1, \ldots, \alpha_n)$ is a vector of the multi-indices in $\mathbb{N}^n$, $|\alpha| = \alpha_1 + \cdots + \alpha_n$, $p_\alpha \in \mathbb{C}$, $\mathbf{x}^\alpha := x_1^{\alpha_1} \ldots x_n^{\alpha_n}$ and $\binom{d}{\alpha} := \frac{d!}{\alpha_1! \ldots \alpha_n!}$. The superscripts $.^t$, $.^*$ and $.^{-1}$ are used respectively for the transpose, Hermitian conjugate, and the inverse matrix. Let $A \in \mathbb{C}^{n \times n}$, we denote by $\sqrt{A}$ a matrix $B \in \mathbb{C}^{n \times n}$ such that $A = B^2$. The complex conjugate is denoted by an overbar, e.g., $\bar{w}$. We use parentheses to denote vectors e.g. $W = (w_i)_{1 \le i \le r}$, and the square brackets to denote matrices e.g. $V = [v_i]_{1 \le i \le r}$ where v_i are column vectors. The concatenation of vectors $v_1, v_2, \ldots$ is denoted $(v_1; v_2; \ldots)$.

Definition 2.1. For $\mathbf{p} = \sum_{|\alpha|=d} \binom{d}{\alpha} p_\alpha \mathbf{x}^\alpha$ and $\mathbf{q} = \sum_{|\alpha|=d} \binom{d}{\alpha} q_\alpha \mathbf{x}^\alpha$ in $\mathbb{C}[\mathbf{x}]_d$, their apolar product is

$$\langle \mathbf{p}, \mathbf{q} \rangle_d := \sum_{|\alpha|=d} \binom{d}{\alpha} \bar{p}_\alpha q_\alpha.$$

The apolar norm of $\mathbf{p}$ is $||\mathbf{p}||_d = \sqrt{\langle \mathbf{p}, \mathbf{p} \rangle_d} = \sqrt{\sum_{|\alpha|=d} \binom{d}{\alpha} \bar{p}_\alpha p_\alpha}$.

The following properties of the apolar product can be verified by direct calculus:

Lemma 2.2. *Let* $\mathbf{l} = (v_1 x_1 + \cdots + v_n x_n)^d := (v^t \mathbf{x})^d \in \mathbb{C}[\mathbf{x}]_d$ *where* $v = (v_i)_{1 \le i \le n}$ *is a vector in* $\mathbb{C}^n$*,* $\mathbf{q} \in \mathbb{C}[\mathbf{x}]_{(d-1)}$*, we have the following two properties:*

1. $\langle \mathbf{l}, \mathbf{p} \rangle_d = \mathbf{p}(\bar{v})$*,* $\langle \mathbf{p}, \mathbf{l} \rangle_d = \bar{\mathbf{p}}(v)$*,*
2. $\langle \mathbf{p}, x_i \mathbf{q} \rangle_d = \frac{1}{d} \langle \partial_{x_i} \mathbf{p}, \mathbf{q} \rangle_{d-1}, \langle x_i \mathbf{q}, \mathbf{p} \rangle_d = \frac{1}{d} \langle \mathbf{q}, \partial_{x_i} \mathbf{p} \rangle_{d-1}, \quad \forall 1 \le i \le n.$

3. The set of non-defective rank-r symmetric tensors

Let $\Sigma_r \subset \mathcal{S}_n^d$ be the set of symmetric tensors of symmetric rank at most r. A symmetric tensor $\mathbf{t} \in \Sigma_r$ is the sum of d^{th} powers

$$\mathbf{t} = \sum_{i=1}^{r} (v_i^t \mathbf{x})^d, \text{ for } v_i \in \mathbb{C}^n. \tag{3}$$

It is a point in the image of the following map:

$$\begin{aligned} \psi_r : \mathbb{C}^{n \times r} := \mathbb{C}^n \times \cdots \times \mathbb{C}^n &\longrightarrow \mathbb{C}[\mathbf{x}]_d \\ [v_i]_{1 \le i \le r} &\longmapsto \psi_r((v_i)_{1 \le i \le r}) = \sum_{i=1}^{r} (v_i^t \mathbf{x})^d. \end{aligned}$$

The d^{th} power $(v_i^t \mathbf{x})^d$ with $v_i \neq 0$ are symmetric tensors of rank-1, which are on the so-called Veronese manifold.

Definition 3.1. Let $\psi : \mathbb{C}^n \to \mathbb{C}[\mathbf{x}]_d,\ v \mapsto (v^t \mathbf{x})^d = \sum_{|\alpha|=d} \binom{d}{\alpha} v^\alpha \mathbf{x}^\alpha$. The Veronese manifold in $\mathbb{C}[\mathbf{x}]_d$ denoted by $\mathcal{V}^{n,d}$ is the set of linear forms in $\mathbb{C}[\mathbf{x}]_1 - \{0\}$ to the d^{th} power. It is the image of $\mathbb{C}^n - \{0\}$ by ψ.

The Veronese variety studied in algebraic geometry is the algebraic variety of the projective space $\mathbb{P}^{s_{n,d}-1}$ associated to $\mathcal{V}^{n,d}$, where $s_{n,d} = \dim \mathbb{C}[\mathbf{x}]_d$ [38, 39, 40]. The tangent space of $\mathcal{V}^{n,d}$ at a point $p = (v^t\mathbf{x})^d$ is the vector space spanned by $\langle x_1(v^t\mathbf{x})^{d-1}, \ldots, x_n(v^t\mathbf{x})^{d-1}\rangle$, that is the linear space $T_p(\mathcal{V}^{n,d}) = \{(u^t\mathbf{x})(v^t\mathbf{x})^{d-1} \mid u \in \mathbb{C}^n\}$.

The Zariski closure $\overline{\Sigma}_r$ of Σ_r is called the $(r-1)$*th-secant variety* of the Veronese variety. For $r > 1$, the algebraic variety $\overline{\Sigma}_r$ is not smooth and contrarily to the case of matrices, singular points of $\overline{\Sigma}_r$ can have a rank $> r$, as shown in the following example. For $d > 2$, $\mathbf{p} = (v_0^t\mathbf{x})(v_1^t\mathbf{x})^{d-1} \in \mathbb{C}[\mathbf{x}]_d$ with $v_0 \neq v_1 \in \mathbb{C}^n$ is in the (Zariski) closure of $\overline{\Sigma}_2$ since $(v_0^t\mathbf{x})(v_1^t\mathbf{x})^{d-1} = \lim_{\delta\to 0} \frac{1}{d\,\delta}(((v_1 + \delta v_0)^t\mathbf{x})^d - (v_1^t\mathbf{x})^d)$ but its symmetric rank is $d > 2$ [37, Proposition 5.6].

To avoid these singularities, we will restrict our theoretical analysis to points of Σ_r where the map ψ_r is a local embedding, since in the vicinity of singularities, the best low rank approximation problem is ill-posed (as shown by the previous example). The map ψ_r is a local embedding at $y = [v_i]_{1\le i\le r} \in \mathbb{C}^{n\times r}$ iff

$$J\psi_r(y) = d\ [x_1(v_1^t\mathbf{x})^{d-1}, \ldots, x_n(v_1^t\mathbf{x})^{d-1}, \ldots, x_1(v_r^t\mathbf{x})^{d-1}, \ldots, x_n(v_r^t\mathbf{x})^{d-1}]$$

is of rank $n\,r$. The tensors $\psi_r(y)$ with $y \in \mathbb{C}^{n\times r}$ such that rank $J\psi_r(y) = n\,r$ are called *non-defective*. The set of non-defective tensors of rank r, locally embedded in $\mathbb{C}[\mathbf{x}]_d$, is the image by a local diffeomorphism of a Riemannian manifold and it is denoted Σ_r^{reg}. The map ψ_r is a local diffeomorphism between an open subset of $\mathbb{C}^{n\times r}$ and $\Sigma_r^{\mathrm{reg}} \subset \overline{\Sigma}_r$.

Hereafter, we consider the cases where $d > 2$ and the rank r is strictly subgeneric, i.e. $r < r_g = \lceil \frac{1}{n}\binom{n+d-1}{d} \rceil$, where r_g is the generic symmetric rank (except for $(d, n) \in \{(3, 5), (4, 3), (4, 4), (4, 5)\}$ or $d = 2$) by Alexander–Hirschowitz theorem [2]. Using "Terracini's lemma" (see e.g. [40, Lemma 5.3.1.1]), we have that Σ_r^{reg} is a dense open subset of $\overline{\Sigma}_r$ iff the dimension of $\overline{\Sigma}_r$ is the expected dimension $n\,r$. In this case, $\overline{\Sigma}_r$ is also said to be *non-defective*. Alexander and Hirschowitz [2] proved that $\overline{\Sigma}_r$ is non-defective when $r < r_g$ (the exceptional defective cases for $d > 2$ being $(d, n, r) \in \{(3, 5, 7), (4, 3, 5), (4, 4, 9), (4, 5, 14)\}$).

It is also known that for $r < r_g$, generic tensors of ψ_r have a unique decomposition, i.e. a unique inverse image by ψ_r up to permutations, except for $(d, n, r) \in \{(6, 2, 9), (4, 3, 8), (3, 5, 9)\}$, see [3, Theorem 1.1].

4. Riemannian optimization for the STA problem

In this section, we use the framework of Riemannian optimization [41] to solve the STA problem. See also [27, 28, 42] for real multilinear tensors. We develop a Riemannian Newton algorithm and a Riemannian Gauss–Newton algorithm exploiting the properties of symmetric tensors to obtain explicit and simplified formulation. We consider distance minimization problems for symmetric tensors with complex decompositions for both algorithms.

Riemannian optimization methods are solving optimization problems over a Riemannian manifold $\mathcal{M}$ [41]. Given $\mathbf{p} \in \mathcal{S}_n^d \sim \mathbb{C}[\mathbf{x}]_d$, we consider hereafter the following least square minimization problem

$$\min_{y\in\mathcal{M}} f(y) \tag{4}$$

where $f : \mathcal{M} \to \mathbb{R}$ is half the square distance function to $\mathbf{p}$ i.e. $f(y) = \frac{1}{2}\|F(y)\|_d^2$ with $F(y) = \Phi_r(y) - \mathbf{p}$, such that $\Phi_r : \mathcal{M} \to \mathbb{C}[\mathbf{x}]_d$ is a parametrization map of Σ_r the set of symmetric tensors of symmetric rank bounded by r, and $\mathcal{M}$ is a Riemannian manifold. A Riemannian optimization method for solving (4) requires a Riemannian metric. Since we will assume that $\mathcal{M}$ is embedded in some space $\mathbb{R}^M$, we will take the metric induced by the Euclidean space $\mathbb{R}^M$.

We propose to parametrize Σ_r, first via weights and unit vectors. We describe an exact Riemannian Newton method for this formulation in subsection 4.1. Secondly, we parametrize Σ_r via sums of the d^{th} power of linear forms that is as sums of tensors in $\mathcal{V}^{n,d}$. We develop a Riemannian Gauss–Newton method for this formulation in subsection 4.2. A dogleg trust-region scheme in subsection 4.3 is added to the two algorithms.

Recall that a Riemannian Newton method for solving (4) [41, Chapter 6] consists of starting with an initial guess $y_0 \in \mathcal{M}$ and generating a sequence $y_1, y_2, \ldots$ in $\mathcal{M}$, with respect to the following process:

$$y_{k+1} \leftarrow R_{y_k}(\eta_k) \quad \text{with} \operatorname{Hess} f(y_k)[\eta_k] = -\operatorname{grad} f(y_k); \tag{5}$$

where $\operatorname{grad} f(y_k)$ and $\operatorname{Hess} f(y_k)$ are respectively the Riemannian gradient and Hessian of f at y_k on $\mathcal{M}$, and $R_{y_k} : T_{y_k}\mathcal{M} \to \mathcal{M}$ is a retraction operator from the tangent space $T_{y_k}\mathcal{M}$ to $\mathcal{M}$.

A Riemannian Gauss–Newton method [41, Chapter 8] is a Riemannian quasi-Newton method where the Riemannian Hessian in (5) is replaced by $(DF(y_k))^* \circ (DF(y_k))$, namely the Gauss–Newton approximation of the Hessian.

The properties of a retraction map are described hereafter:

Definition 4.1. [41, 43, 42] Let $\mathcal{M}$ be a manifold and $y \in \mathcal{M}$. A retraction R_y is a map $T_y\mathcal{M} \to \mathcal{M}$, which satisfies the following properties :

1. $R_y(0_y) = y$;
2. there exists an open neighborhood $\mathcal{U}_y \subset T_y\mathcal{M}$ of 0_y such that the restriction on $\mathcal{U}_y$ is well-defined and a smooth map;
3. R_y satisfies the local rigidity condition

$$DR_y(0_y) = id_{T_y\mathcal{M}},$$

where $id_{T_y\mathcal{M}}$ denotes the identity map on $T_y\mathcal{M}$.

We will also use the following property which is straightforward to show:

Lemma 4.2. *Let $\mathcal{M}_1, \ldots, \mathcal{M}_r$ be manifolds, $y_i \in \mathcal{M}_i$ and $\mathcal{M} = \mathcal{M}_1 \times \cdots \times \mathcal{M}_r$ and $y = (y_1, \ldots, y_r) \in \mathcal{M}$. Let $R_i : T_{y_i}\mathcal{M}_i \to \mathcal{M}_i$ be retractions. Then $R_y : T_y\mathcal{M} \to \mathcal{M}$ defined as follows: $R_y(\xi_1, \ldots, \xi_r) = (R_{y_1}(\xi_1), \ldots, R_{y_r}(\xi_r))$ for $\xi_i \in T_{y_i}\mathcal{M}_i$, $1 \leq i \leq r$, is a retraction on $\mathcal{M}$.*

4.1. Riemannian Newton method for STA

We normalize the decomposition (3) by choosing unit vectors for v_i and positive weights. Namely, we decompose a symmetric tensor $\mathbf{p} \in \Sigma_r$ as $\mathbf{p} = \sum_{i=1}^r w_i\,(v_i^t\mathbf{x})^d$ with $w_i \in \mathbb{R}_+^*$ and $||v_i|| = 1$, for $1 \leq i \leq r$; by normalizing v_i and multiplying by $w_i := ||v_i||^d$ if v_i is not a unit vector. The vector $(w_i)_{1\leq i\leq r}$ in this decomposition is called "the weight vector", and is denoted by W. Let $V = [v_i]_{1\leq i\leq r} \in \mathbb{C}^{n\times r}$ be the matrix of the normalized vectors.

The objective function expressed in terms of these weights and unit vectors is given by $f(W,V) = \frac{1}{2}||F(W,V)||_d^2$, with $F(W,V) = \sum_{i=1}^r w_i\,(v_i^t\mathbf{x})^d - \mathbf{p}$.

The function f is a real valued function of complex variables; such function is non-analytic, because it cannot satisfy the Cauchy–Riemann conditions [44]. To apply the Riemannian Newton method, we need the second order differentials of f. As discussed in [30], we overcome the non-analytic problem by converting the optimization problem to the real domain, regarding f as a function of the real and imaginary parts of its complex variables.

Let $\mathcal{N}_r = \{(W, \Re(V), \Im(V)) \mid W \in {\mathbb{R}_+^*}^r, V \in \mathbb{C}^{n\times r}, (\Re(v_i), \Im(v_i)) \in \mathbb{S}^{2n-1},\ \forall\, 1 \leq i \leq r\}$, where $\mathbb{S}^{2n-1}$ is the unit sphere in $\mathbb{R}^{2n}$. Let $\varphi_r : (w, v_1, \ldots, v_r, v_1', \ldots, v_r') \in \mathcal{N}_r \mapsto \sum_{i=1}^{r} w_i((v_i + \mathbf{i}\, v_i')^t \mathbf{x})^d$. Hereafter in this subsection, we use the following formulation to compute the different ingredients of a Riemannian Newton method:

$$(\mathrm{STA})_{\mathcal{N}_r} \quad \min_{y \in \mathcal{N}_r} f(y),$$

where $f(y) = \frac{1}{2}||F(y)||_d^2$, with $F(y) = \varphi_r(y) - \mathbf{p}$.

4.1.1. Computation of the gradient vector and the Hessian matrix

In this section, we present the explicit expressions of the Riemannian gradient and Hessian on $\mathcal{N}_r$. We first describe an orthonormal basis of $T_y\mathcal{N}_r$ for $y \in \mathcal{N}_r$. Then we detail the computation of the gradient and Hessian in this basis, via the differentials of maps in complex and conjugate variables.

Lemma 4.3. *Let $y = (w, v_1, \ldots, v_r, v_1', \ldots, v_r') \in \mathcal{N}_r$. For all $i = 1, \ldots, r$ let $\check{v}_i = (v_i; v_i') \in \mathbb{S}^{2n-1}$ and let*

$$(I_{2n} - \check{v}_i \check{v}_i^t) = Q_i R_i P_i$$

be a rank-revealing QR-decomposition of the projector on $\check{v}_i^\perp$ in $\mathbb{R}^{2n}$, where $Q_i Q_i^t = I_{2n}$, R_i is upper triangular, and P_i is a permutation matrix.

Let $Q_{i,re}$ (resp. $Q_{i,im}$) be the matrix given by the first n rows (resp. the last n rows) and the first $2n-1$ columns of Q_i. Let $\tilde{Q} = \begin{bmatrix} Q_{re} \\ Q_{im} \end{bmatrix} \in \mathbb{R}^{2nr\times(2n-1)r}$, where $Q_{re} = \mathrm{diag}(Q_{i,re})_{1\leq i\leq r}$ and $Q_{im} = \mathrm{diag}(Q_{i,im})_{1\leq i\leq r}$. Then the columns of $Q = \mathrm{diag}(I_r, \tilde{Q})$ form an orthonormal basis of $T_y\mathcal{N}_r$.

Proof. We have $T_y\mathcal{N}_r \simeq T_w(\mathbb{R}_+^*)^r \times T_Z\mathcal{S}_r$, where $\mathcal{S}_r = \{(\Re(V), \Im(V)) \mid V \in \mathbb{C}^{n\times r}, ||v_i||^2 = 1,\ \forall 1 \leq i \leq r\}$ and $Z = (\Re(V), \Im(V)) = (v_1, \ldots, v_r, v_1', \ldots, v_r') \in \mathbb{R}^{n\times 2r}$.

As $T_w(\mathbb{R}_+^*)^r = \mathbb{R}^r$, I_r represents an orthonormal basis of $T_w(\mathbb{R}_+^*)^r$.

We verify now that $\tilde{Q}$ is an orthonormal basis of $T_Z\mathcal{S}_r$. For $i = 1, \ldots, r$, $\check{v}_i \in \mathbb{S}^{2n-1} \subset \mathbb{R}^{2n}$ and the first $(2n-1)$ columns of the factor Q_i of a rank-revealing QR-decomposition of $I_{2n} - \check{v}_i\check{v}_i^t$ give an orthonormal basis of the image $\check{v}_i^\perp$ of $(I_{2n} - \check{v}_i\check{v}_i^t)$, that is $T_{\check{v}_i}\mathbb{S}^{2n-1}$.

The vector space $T_Z\mathcal{S}_r$, of dimension $r(2n-1)$, is the Cartesian product of the tangent spaces $T_{\check{v}_i}\mathbb{S}^{2n-1}$. Therefore, by construction, the columns of $\tilde{Q}$ form an orthonormal basis of $T_Z\mathcal{S}_r$.

We deduce that $Q = \mathrm{diag}(I_r, \tilde{Q})$ represents an orthonormal basis of $T_y\mathcal{N}_r$ in the canonical basis of $\mathbb{R}^{2nr}$. □

Let $\mathcal{R}_r = \{(W, \Re(V), \Im(V)) \in \mathbb{R}^r \times \mathbb{R}^{n\times r} \times \mathbb{R}^{n\times r} \mid W \in \mathbb{R}^r, V \in \mathbb{C}^{n\times r}\}$ and let f_R be the function f seen as a function on $\mathcal{R}_r$. The gradient and the Hessian of f_R at a point $p^R \in \mathcal{R}_r$ are called the real gradient and the real Hessian. We denote them by G^R and H^R. We will describe their computation, after the next proposition, relating them to the Riemannian gradient and Hessian.

Proposition 4.4. *Let $p = (w, v_1, \dots, v_r, v'_1, \dots, v'_r) \in \mathcal{N}_r$, $Q \in \mathbb{R}^{(r+2nr)\times(r+(2n-1)r)}$ such that its columns form an orthonormal basis of $T_y\mathcal{N}_r$. Let $G^R = (g_0, g_1, \dots, g_r, g'_1, \dots, g'_r) \in \mathbb{R}^{r+2nr}$ (resp. $H^R \in \mathbb{R}^{(r+2nr)\times(r+2nr)}$) be the gradient vector (resp. the Hessian matrix) of f_R at p^R in the canonical basis. The Riemannian gradient vector (resp. Hessian matrix) of f at p with respect to the basis Q is given by:*

$$G = Q^t G^R, H = Q^t(H^R + S)Q,$$

where $S = \operatorname{diag}(0_{r\times r}, \tilde{S}, \tilde{S})$, with $\tilde{S} = \operatorname{diag}(s_1 I_n, \dots, s_r I_n)$, $s_i = \langle v_i, g_i\rangle + \langle v'_i, g'_i\rangle$.

The proof is given in Appendix A.1.
Let us describe now explicitly the real gradient G^R:

Proposition 4.5. *The gradient G^R of f_R on $\mathcal{R}_r$ is the vector*

$$G^R = \begin{pmatrix} G_1 \\ \Re(G_2) \\ -\Im(G_2) \end{pmatrix} \in \mathbb{R}^{r+2nr},$$

where

- $G_1 = (\sum_{i=1}^r w_i \Re((v_j^* v_i)^d) - \Re(\bar{\mathbf{p}}(v_j)))_{1\le j\le r} \in \mathbb{R}^r$,
- $G_2 = (d\sum_{i=1}^r w_i w_j (v_i^* v_j)^{(d-1)} \bar{v}_i - w_j \nabla_{\mathbf{x}} \bar{\mathbf{p}}(v_j))_{1\le j\le r} \in \mathbb{C}^{nr}$.

The matrix of the real Hessian can be computed as follows:

Proposition 4.6. *The real Hessian matrix H^R is the following block matrix:*

$$H^R = \begin{bmatrix} A & \Re(B)^t & -\Im(B)^t \\ \Re(B) & \Re(C+D) & -\Im(C+D) \\ -\Im(B) & \Im(D-C) & \Re(D-C) \end{bmatrix} \in \mathbb{R}^{(r+2nr)\times(r+2nr)},$$

with

- $A = \Re([(v_i^* v_j)^d]_{1\le i,j\le r}) \in \mathbb{R}^{r\times r}$,
- $B = [dw_i(v_j^* v_i)^{d-1}\bar{v}_j + \delta_{i,j}(d\sum_{l=1}^r w_l (v_l^* v_i)^{d-1}\bar{v}_l - \nabla_{\mathbf{x}}\bar{\mathbf{p}}(v_j))]_{1\le i,j\le r} \in \mathbb{C}^{nr\times r}$, *where $\delta_{i,j}$ is the Kronecker delta,*
- $C = \operatorname{diag}[d(d-1)[\sum_{i=1}^r w_i w_j \overline{v_{i,k} v_{i,l}}(v_i^* v_j)^{d-2}]_{1\le k,l\le n} - w_j \Delta_{\mathbf{x}}\bar{\mathbf{p}}(v_j)]_{1\le j\le r} \in \mathbb{C}^{nr\times nr}$, *where* $\Delta_{\mathbf{x}}\bar{\mathbf{p}}(v_j) := [\partial_{x_k}\partial_{x_l}\bar{\mathbf{p}}(v_j)]_{1\le k,l\le n}$,
- $D = [dw_i w_j (v_i^* v_j)^{d-2}((v_i^* v_j)I_n + (d-1)v_j v_i^*)]_{1\le i,j\le r} \in \mathbb{C}^{nr\times nr}$.

The Appendix A.2 is devoted to discussing the computation details of the real gradient and Hessian, where the proofs of propositions 4.5 and 4.6 are covered respectively in Appendix A.2.1 and Appendix A.2.2.

4.1.2. Retraction on $\mathcal{N}_r$

To complete this Riemannian Newton method, we need to define a retraction operator on $\mathcal{N}_r$. Let us assume that the Riemannian Newton equation is solved at a point $y = (w, v_1, \ldots, v_r, v'_1, \ldots v'_r) \in \mathcal{N}_r$, in local coordinates with respect to the basis Q as in lemma 4.3. It yields a solution vector $\hat{\eta} \in \mathbb{R}^{r+r(2n-1)}$. The tangent vector $\eta \in T_y\mathcal{N}_r$ of size $r + 2nr$ is given by $\eta = Q\,\hat{\eta} = (\nu, \eta_1, \ldots, \eta_r, \eta'_1, \ldots, \eta'_r)$. The new point $R_y(\eta) = (\tilde{w}, \tilde{v}_1, \ldots, \tilde{v}_r, \tilde{v}'_1, \ldots, \tilde{v}'_r) \in \mathcal{N}_r$ is defined using the product of the retractions on each component, that is the identity map on $\mathbb{R}^r$ and the projection map on the sphere $\mathbb{S}^{2n-1}$ [45] as follows:

- $\tilde{w} = R_w(\nu) = w + \nu$;
- $(\tilde{v}_j, \tilde{v}'_j) = R_{(v_j;v'_j)}(\eta_j, \eta'_j) = \frac{(v_j+\eta_j;v'_j+\eta'_j)}{||(v_j+\eta_j;v'_j+\eta'_j)||}$.

By lemma 4.2, this defines a retraction from $T_y\mathcal{N}_r$ to $\mathcal{N}_r$ since R_w (resp. $R_{(v_j;v'_j)}$) is a retraction on $\mathbb{R}^r$ (resp. $\mathbb{S}^{2n-1}$).

4.2. Riemannian Gauss–Newton for STA

In this subsection, we consider the STA problem over the product of r Veronese manifolds $\mathcal{V}^{n,d}$. By separating the real and imaginary parts of the coefficients of a polynomial, the non-zero points $(v^t\mathbf{x})^d$ with $v \in \mathbb{C}^n \setminus \{0\}$ form a smooth Riemannian variety in $\mathbb{C}[\mathbf{x}]_d$. We equip the $\mathbb{R}$-vector space $\mathbb{C}[\mathbf{x}]_d \sim \mathbb{R}^{2s_{n,d}}$ with the real inner product:

$$\forall \mathbf{p}, \mathbf{q} \in \mathbb{C}[\mathbf{x}]_d, \quad \langle \mathbf{p}, \mathbf{q}\rangle_d^{\mathbb{R}} = \Re(\langle \mathbf{p}, \mathbf{q}\rangle_d).$$

Let $\mathcal{V}_r := \mathcal{V}^{n,d} \times \cdots \times \mathcal{V}^{n,d}$. The map $\sigma_r : y = (y_1, \ldots, y_r) \in \mathcal{V}_r \mapsto y_1 + \cdots + y_r \in \mathbb{C}[\mathbf{x}]_d$ is a parameterization of the set Σ_r of symmetric tensors of symmetric rank at most r. We formulate the STA problem as a Riemannian least square problem over $\mathcal{V}_r$ as follows:

$$(\mathrm{STA})_{\mathcal{V}_r} \quad \min_{y \in \mathcal{V}_r} f(y),$$

where $f(y) = \frac{1}{2}||F(y)||_d^2$, with $F(y) = \sigma_r(y) - \mathbf{p}$ for $y \in \mathcal{V}_r$.

The differential map $DF = D\sigma_r$ at $y = (y_1, \ldots, y_r) \in \mathcal{V}_r$ with $y_i = (v_i^t\mathbf{x})^d$, $v_i \in \mathbb{C}^n$ is

$$\begin{array}{rcl} D\sigma_r(y) : T_{y_1}\mathcal{V}^{n,d} \times \cdots \times T_{y_r}\mathcal{V}^{n,d} & \to & T_{\sigma_r(y)}\mathbb{C}[\mathbf{x}]_d = \mathbb{C}[\mathbf{x}]_d \\ (\eta_1, \ldots, \eta_r) & \mapsto & \eta_1 + \cdots + \eta_r, \end{array}$$

where $T_{y_i}\mathcal{V}^{n,d} = \left\{(u^t\mathbf{x})(v_i^t\mathbf{x})^{d-1} | \ u \in \mathbb{C}^n\right\}$ is of dimension $2n$ over $\mathbb{R}$.

Recall that the Gauss–Newton equation is given by [41, Chapter 8]:

$$(DF(y))^* \circ (DF(y))[\eta] = -(DF(y))^*[F(y)], \tag{6}$$

where $DF(y) : T_y\mathcal{V}_r \to \mathbb{C}[\mathbf{x}]_d$ is the differential map of F at y. The map $(DF(y))^* \circ (DF(y)) : T_y\mathcal{V}_r \to T_y\mathcal{V}_r$ is the so-called Gauss–Newton approximation of the Hessian of f at y.

We are going to describe explicitly the matrix of this map in a convenient basis of $T_y\mathcal{V}_r$. For a non-zero complex vector $v \in \mathbb{C}^n$, we define the inner product: $\forall u, u' \in \mathbb{C}^n$,

$$\langle u, u'\rangle_v = \Re\big(u^*u' + (d-1)(u^*v)(v^*u')\|v\|^{-2}\big)$$

It is a positive definite inner product on $\mathbb{C}^n \sim \mathbb{R}^{2n}$ since $\langle u, u\rangle_v = \|u\|^2 + (d-1)|(v^*u)|^2\|v\|^{-2} \geq 0$ and it vanishes iff $u = 0$. Notice that $\langle v, v\rangle_v = d\|v\|^2$. The symmetric matrix associated to this inner product in the canonical basis of $\mathbb{R}^{2n}$ is

$$M_v := I_{2n} + (d-1)\|v\|^{-2}(v_R v_R^t + v_I v_I^t)$$

where $v_R = (\Re(v); \Im(v)), v_I = (-\Im(v); \Re(v))$ are the vectors of $\mathbb{R}^{2n}$ obtained by concatenating the real and imaginary part (resp. opposite imaginary and real part) of $v \in \mathbb{C}^n$.
Let $u_1 = \frac{v_R}{\|v\|}$ and $u_2 = \frac{v_I}{\|v\|}$. We notice that u_1 and u_2 can be completed to an orthonormal basis of $\mathbb{R}^{2n}$. Let U denotes the matrix of this basis i.e. $U = [u_1, \ldots, u_{2n}]$. Then $UU^t = I_{2n}$, so that an eigenvalue decomposition of the symmetric matrix M_v of $\langle \cdot, \cdot\rangle_v$ in the canonical basis of $\mathbb{R}^{2n}$ can be written as follows:

$$M_v = U\text{diag}(1 + (d-1), 1 + (d-1), 1, \ldots, 1)U^t = U\text{diag}(d, d, 1, \ldots, 1)U^t. \qquad (7)$$

For shortness, we denote the strictly positive diagonal matrix $\text{diag}(d, d, 1, \ldots, 1)$ by S.

Lemma 4.7. *Let $v \neq 0 \in \mathbb{C}^n \sim \mathbb{R}^{2n}$ and $p = (v^t\mathbf{x})^d \in \mathbb{C}[\mathbf{x}]_d$. Let $u_1, \ldots, u_{2n} \in \mathbb{C}^n$ be an orthonormal $\mathbb{R}$-basis for the inner product $\langle \cdot, \cdot\rangle_v$ with $u_1 = \frac{v}{\sqrt{d}\|v\|}$. Then*

$$\mathbf{q}_i = \sqrt{d}\|v\|^{-d+1}(u_i^t\mathbf{x})(v^t\mathbf{x})^{d-1}, i = 1, \ldots, 2n$$

is an orthonormal basis of $T_p\mathcal{V}^{n,d}$ for the inner product $\langle \cdot, \cdot\rangle_d^{\mathbb{R}}$.

Proof. Using the apolar identities in lemma 2.2, we have

$$\begin{aligned}
\langle \mathbf{q}_i, \mathbf{q}_j\rangle_d^{\mathbb{R}} &= \sqrt{d}\|v\|^{-d+1}\Re\big(\langle (u_i^t\mathbf{x})(v^t\mathbf{x})^{d-1}, \mathbf{q}_j\rangle_d\big) \\
&= \sqrt{d^{-1}}\|v\|^{-d+1}\Re\big((u_i^*\nabla_{\mathbf{x}}\mathbf{q}_j)(\overline{v})\big) \\
&= \|v\|^{-2d+2}\Re\big((u_i^*u_j)(v^*v)^{d-1} + (d-1)(u_i^*v)(v^*u_j)(v^*v)^{d-2}\big) \\
&= \Re\big((u_i^*u_j) + (d-1)(u_i^*v)(v^*u_j)\|v\|^{-2}\big) = \langle u_i, u_j\rangle_v.
\end{aligned}$$

We deduce that $\langle \mathbf{q}_i, \mathbf{q}_j\rangle_d^{\mathbb{R}} = \delta_{i,j}$ and $(\mathbf{q}_i)_{i=1,\ldots,2n}$ is an orthonormal basis of $T_{(v^t\mathbf{x})^d}\mathcal{V}^{n,d}$ for the inner product $\langle \cdot, \cdot\rangle_d^{\mathbb{R}}$. □

We describe now how to compute an orthonormal basis for $\langle \cdot, \cdot\rangle_v$.

Lemma 4.8. *Let $M_v = USU^t$ be the eigenvalue decomposition of M_v as in (7). Let $\hat{u}_1 = \sqrt{S^{-1}}U^t\frac{v_R}{\sqrt{d}\|v\|}$ and let $Q \in \mathbb{R}^{2n\times 2n}$ be the orthogonal factor of a rank-revealing QR-decomposition of $I_{2n} - \hat{u}_1\hat{u}_1^t = QRP$ where R is upper triangular and P is a permutation matrix. Let*

$$u_{R,1} = \frac{v_R}{\sqrt{d}\|v\|}, u_{R,i} = U\sqrt{S^{-1}}Q_{[:,i-1]} \quad i = 2, \ldots, 2n.$$

Then the orthonormal $\mathbb{R}$-basis $u_1, \ldots, u_{2n} \in \mathbb{C}^n$ for $\langle \cdot, \cdot\rangle_v$ is such that $u_i = (u_{R,i})_{[1:n]} + \mathbf{i}\,(u_{R,i})_{[n+1:2n]} \in \mathbb{C}^n$ for $i = 1, \ldots, 2n$.

Proof. As $M_v = USU^t$ with $UU^t = I_{2n}$ and $S \in \mathbb{R}^{2n\times 2n}$ a strictly positive diagonal matrix, we have $\sqrt{S^{-1}}U^t M_v U \sqrt{S^{-1}} = I_{2n}$. Thus the column vectors of $U\sqrt{S^{-1}}$ form an orthonormal basis of $\mathbb{R}^{2n}$ for $\langle \cdot, \cdot \rangle_v$.

The vector $\hat{u}_1 = \sqrt{S^{-1}}U^t \frac{v_R}{\sqrt{d}\|v\|}$ is representing the vector $\frac{v_R}{\sqrt{d}\|v\|}$ in this orthonormal basis.

The first $2n-1$ columns of the factor Q in a rank-revealing QR-decomposition of $I_{2n} - \hat{u}_1\hat{u}_1^t = QRP$ are orthonormal vectors $\hat{u}_2, \ldots, \hat{u}_{2n}$ for $\langle \cdot, \cdot \rangle_v$, expressed in the basis $U\sqrt{S^{-1}}$. An orthonormal basis $u_{R,1}, u_{R,2}, \ldots, u_{R,2n} \in \mathbb{R}^{2n}$ for $\langle \cdot, \cdot \rangle_v$ is thus given by $u_{R,1} = \frac{v_R}{\sqrt{d}\|v\|}$, $u_{R,i} = U\sqrt{S^{-1}}Q_{[:,i-1]}$, $i = 2, \ldots, 2n$. The corresponding vectors $\in \mathbb{C}^n$ are $u_i = (u_{R,i})_{[1:n]} + \mathbf{i}\,(u_{R,i})_{[n+1:2n]} \in \mathbb{C}^n$ for $i = 1, \ldots, 2n$. □

Notice that when v is real and u, u' are real such that $\langle v, u \rangle = \langle v, u' \rangle = 0$, $\langle u, u' \rangle_v = \langle u, u' \rangle$ is the standard inner product of u, u'. Consequently in the real case, an orthonormal basis $(u_i)_{i=1,\ldots,n} \subset \mathbb{R}^n$ can be obtained directly from $u_1 = \frac{v}{\|v\|}$ and a rank-revealing QR-decomposition of $I_n - u_1 u_1^t$.

For $y = (y_1, \ldots, y_r) \in \mathcal{V}_r$ with $y_i = (v_i^t \mathbf{x})^d \in \mathcal{V}^{n,d}$, $\forall 1 \le i \le r$, let $(\mathbf{q}_{i,j})_{j=1,\ldots,2n}$ be the orthonormal basis associated to v_i defined in lemma 4.7 and let $Q_i = [\mathbf{q}_{i,1}, \ldots, \mathbf{q}_{i,2n}] \in \mathbb{R}^{2s_{n,d}\times 2n}$ be the coefficient matrix of the polynomials $(\mathbf{q}_{i,j})_{j=1,\ldots,2n}$ in the canonical $\mathbb{R}$-basis of $\mathbb{C}[\mathbf{x}]_d$. The columns of the matrix

$$Q = \operatorname{diag}(Q_i)_{1\le i \le r},$$

represent an orthonormal basis of $T_y\mathcal{V}_r$ for the inner product induced by $\langle \cdot, \cdot \rangle_d^{\mathbb{R}}$ on each component.

Therefore, the Jacobian matrix J of σ_r at y, which is the matrix associated to $D\sigma_r(y) = DF(y)$, with respect to the orthonormal basis Q on $T_y\mathcal{V}_r$ and the standard real basis on $\mathbb{C}[\mathbf{x}]_d$ is given by:

$$J = [Q_1, \ldots, Q_r] \in \mathbb{R}^{2s_{n,d}\times 2nr}.$$

Proposition 4.9. *The Gauss–Newton equation (6) in the orthonormal basis Q of $T_y\mathcal{V}_r$ is of the form*

$$H\,\tilde{\eta} = -G,$$

where $\tilde{\eta}^t = (\tilde{\eta}_1^t, \ldots, \tilde{\eta}_r^t) \in \mathbb{R}^{2nr}$ is the unknown coordinate vector of an element of the tangent space $T_y\mathcal{V}_r$ in the basis Q and

- *$G = [G_k]_{k=1,\ldots,2nr}$ with for $1 \le i \le r,\ 1 \le j \le 2n$,*
  $G_{2n(i-1)+j} = \sqrt{d^{-1}}\|v_i\|^{-d+1}\Big(d\sum_{k=1}^r \Re\big((u_{i,j}^* v_k)(v_i^* v_k)^{d-1}\big) - \Re\big(u_{i,j}^* \nabla_{\mathbf{x}}\mathbf{p}(\overline{v_i})\big)\Big)$,
- *$H = [H_{k,k'}]_{1\le k,k' \le 2nr}$ with for $1 \le i, i' \le r,\ 1 \le j, j' \le 2n$,*
  $H_{2n(i-1)+j,2n(i'-1)+j'} = \|v_i\|^{-d+1}\|v_{i'}\|^{-d+1}\Big(\Re\big((u_{i,j}^* u_{i',j'})(v_i^* v_{i'})^{d-1}\big) + (d-1)\Re\Big((u_{i,j}^* v_{i'})(v_i^* u_{i',j'})$
  $(v_i^* v_{i'})^{d-2}\Big)\Big)$.

Proof. As the matrix of $D\sigma_r(y) = DF(y)$ in the orthonormal basis Q on $T_y\mathcal{V}_r$ and the standard real basis on $\mathbb{C}[\mathbf{x}]_d$ is J, we have that the Gauss–Newton equation (6) is $H\tilde{\eta} = -G$ with

- $G = J^t vec(\sigma_r(y) - \mathbf{p}) = (\langle \mathbf{q}_{i,j}, \sigma_r(y) - \mathbf{p} \rangle_d^{\mathbb{R}})$,
- $H = J^t J = [Q_1, \ldots, Q_r]^t [Q_1, \ldots, Q_r] = (\langle \mathbf{q}_{i,j}, \mathbf{q}_{i',j'} \rangle_d^{\mathbb{R}})$.

By the apolar identities in lemma 2.2, we have

$$\begin{aligned}\langle \mathbf{q}_{i,j}, \sigma_r(y) - \mathbf{p}\rangle_d^{\mathbb{R}} &= \sqrt{d^{-1}}\|v_i\|^{-d+1}\Big(\sum_{k=1}^{r} \Re(u_{i,j}^* \nabla_{\mathbf{x}} (v_k^t \mathbf{x})^d(\overline{v_i}) - u_{i,j}^* \nabla_{\mathbf{x}} \mathbf{p}(\overline{v_i}))\Big) \\ &= \sqrt{d^{-1}}\|v_i\|^{-d+1}\Big(\sum_{k=1}^{r} \Re\big(d(u_{i,j}^* v_k)(v_i^* v_k)^{d-1} - u_{i,j}^* \nabla_{\mathbf{x}} \mathbf{p}(\overline{v_i})\big)\Big).\end{aligned}$$

Similarly,

$$\begin{aligned}&\langle \mathbf{q}_{i,j}, \mathbf{q}_{i',j'}\rangle_d^{\mathbb{R}} \\ &= \|v_i\|^{-d+1}\|v_{i'}\|^{-d+1}\Re\big(u_{i,j}^* \nabla_{\mathbf{x}}((u_{i',j'}^t \mathbf{x})(v_{i'}^t \mathbf{x})^{d-1})(\overline{v_i})\big) \\ &= \|v_i\|^{-d+1}\|v_{i'}\|^{-d+1}\Re\big((u_{i,j}^* u_{i',j'})(v_i^* v_{i'})^{d-1} + (d-1)(u_{i,j}^* v_{i'})(v_i^* u_{i',j'})(v_i^* v_{i'})^{d-2}\big),\end{aligned}$$

which ends the proof of the proposition. □

The Gauss–Newton equation

$$H\,\tilde{\eta} = -G,$$

solved in local coordinate with respect to the basis Q, yields a vector $\tilde{\eta} = (\tilde{\eta}_1; \ldots; \tilde{\eta}_r) \in \mathbb{R}^{2nr}$. The components of the tangent vector $\eta = (\eta_1, \ldots, \eta_r) \in T_y\mathcal{V}_r \in \mathbb{C}[\mathbf{x}]_d$ are then

$$\eta_i = \sqrt{d}||v_i||^{-d+1}(v_i^t \mathbf{x})^{d-1} \sum_{k=1}^{2n} \tilde{\eta}_{i,k}(u_{i,k}^t \mathbf{x}), \quad i = 1, \ldots, r.$$

4.2.1. Retraction on the Veronese manifold

We define the retraction of a tangent vector $\eta \in T_y\mathcal{V}_r$ to a new point $\tilde{y}$ on the manifold $\mathcal{V}_r$ as follows:

$$\tilde{y} = (\tilde{y}_1, \ldots, \tilde{y}_r) = (R_{y_1}(\eta_1), \ldots, R_{y_r}(\eta_r)),$$

where $R_{y_i} : T_{y_i}\mathcal{V}^{n,d} \to \mathcal{V}^{n,d}$ is a retraction operator on the Veronese manifold for $i \in \{1, \ldots, r\}$ that we describe hereafter (see lemma 4.2). The retraction on the Veronese manifold that we are going to describe is implemented directly on $y_i + \eta_i$ in the ambient space $\mathbb{C}[\mathbf{x}]_d$ without considering any tensor compression techniques on the symmetric tensor associated to $y_i + \eta_i$, as is elaborated for instance in [42] for multilinear tensors.

We will use the following matrix construction to define the retraction on $\mathcal{V}^{n,d}$.

Definition 4.10. The Hankel matrix of degree $(k, d-k)$ associated to a polynomial $\mathbf{p}$ in $\mathbb{C}[\mathbf{x}]_d$ is given by:

$$H_{\mathbf{p}}^{k,d-k} = (\langle \mathbf{p}, \mathbf{x}^{\alpha+\beta}\rangle_d)_{|\alpha|=k, |\beta|=d-k}.$$

This matrix is also known as the *Catalecticant matrix* of the symmetric tensor $\mathbf{p}$ in degree $(k, d-k)$ or the *flattening* of $\mathbf{p}$ in degree $(k, d-k)$. In this definition, we implicitly assume that we have chosen a monomial ordering (for instance the lexicographic ordering on the monomials indexing the rows and columns of $H_{\mathbf{p}}^{k,d-k}$) to build the Hankel matrix. The properties of Hankel matrices that we will use are independent of this ordering. Such a matrix is called a Hankel matrix since, as in the classical case, the entries of the matrix depend on the sum of the exponents of the monomials indexing the corresponding rows and columns.

When $k = 1$, using the apolar relations $\langle \mathbf{p}, x_i \mathbf{x}^\beta \rangle_d = \frac{1}{d} \langle \partial_{x_i} \mathbf{p}, \mathbf{x}^\beta \rangle_{d-1}$, we see that $H_{\mathbf{p}}^{1,d-1}$ is nothing else than the transposed of the coefficient matrix of the gradient $\frac{1}{d} \nabla_{\mathbf{x}} \mathbf{p}$ in the basis $\left(\mathbf{x}^\beta \binom{d-1}{\beta}^{-1} \right)_{|\beta|=d-1}$. When $\mathbf{p} = (v^t \mathbf{x})^d \in \mathcal{V}^{n,d}$, $H_{\mathbf{p}}^{1,d-1}$ can thus be written as the rank-1 matrix $v \otimes (v^t \mathbf{x})^{d-1}$.

Our construction of a retraction on $\mathcal{V}^{n,d}$ is described in the following definition.

Definition 4.11. For $v \in \mathbb{C}^n \setminus \{0\}$, let $\pi_v : \mathbb{C}[\mathbf{x}]_d \to \mathcal{V}^{n,d}$ be the map such that $\forall \mathbf{q} \in \mathbb{C}[\mathbf{x}]_d$,

$$\pi_v(\mathbf{q}) = \frac{\langle \psi(v), \mathbf{q} \rangle_d}{\|\psi(v)\|_d^2} \psi(v), \tag{8}$$

where $\psi : v \in \mathbb{C}^n \mapsto (v^t \mathbf{x})^d \in \mathcal{V}^{n,d}$ is the parametrization of the Veronese variety. For $\mathbf{p} \in \mathbb{C}[\mathbf{x}]_d$, let $\theta(\mathbf{p}) \in \mathbb{C}^n$ be the first left singular vector of $H_{\mathbf{p}}^{1,d-1}$. For $\mathbf{p} \in \mathcal{V}^{n,d}$, let

$$\begin{array}{rcl} R_{\mathbf{p}} : T_{\mathbf{p}} \mathcal{V}^{n,d} & \to & \mathcal{V}^{n,d} \\ \mathbf{q} & \mapsto & \pi_{\theta(\mathbf{p}+\mathbf{q})}(\mathbf{p} + \mathbf{q}). \end{array}$$

The retraction that we are going to describe on the Veronese manifold is closely related to the one on the Segre manifold used in [27]. In fact, since the Segre manifold coincides with the manifold of tensors of multilinear rank $(1, \ldots, 1)$, the retraction in [27] is deduced from the truncated multilinear rank $(1, \ldots, 1)$ HOSVD of a real multilinear tensor, i.e. from the truncated rank one SVD of the matricization in the different modes [14]. For a symmetric tensor, the matricization with respect to any mode gives the same Catalecticant matrix in degree (1, d-1). Hereafter, we show, by different techniques, that a single truncated SVD of the Catalecticant matrix in degree $(1, d-1)$ gives a retraction on the Veronese manifold.

By the apolar identities, we check that $R_{\mathbf{p}}(\mathbf{q}) = (\mathbf{p}(\bar{u}) + \mathbf{q}(\bar{u}))\,(u^t \mathbf{x})^d$ where $u = \theta(\mathbf{p} + \mathbf{q})$. We also verify that $\pi_{\lambda u} = \pi_u$ for any $\lambda \in \mathbb{C} \setminus \{0\}$ and any $u \in \mathbb{C}^n \setminus \{0\}$.

By the relation (8), for any $v \in \mathbb{C}^n \setminus \{0\}$, $\pi_v(\mathbf{q})$ is the vector on the line spanned by $\psi(v)$, which is the closest to $\mathbf{q}$ for the apolar norm. In particular, we have $\pi_v(\psi(v)) = \psi(v)$.

We verify now that $R_{\mathbf{p}}$ is a retraction on $\mathcal{V}^{n,d}$.

Lemma 4.12. *Let $\mathbf{p} \in \mathcal{V}^{n,d}$. Then, $\mathbf{p}$ is a fixed point by π_u where u is the first left singular vector of $H_{\mathbf{p}}^{1,d-1}$.*

Proof. If $\mathbf{p} = (v^t \mathbf{x})^d = \psi(v) \in \mathcal{V}^{n,d}$ with $v \in \mathbb{C}^n \setminus \{0\}$, then the first left singular vector u of $H_{\mathbf{p}}^{1,d-1}$ is up to a scalar equal to v. Thus we have $\pi_u(\mathbf{p}) = \pi_v(\psi(v)) = \psi(v) = \mathbf{p}$. □

Proposition 4.13. *Let $\mathbf{p} \in \mathcal{V}^{n,d}$. There exists a neighborhood $\mathcal{U}_{\mathbf{p}} \subset \mathbb{C}[\mathbf{x}]_d$ of $\mathbf{p}$ such that the map $\rho : \mathbf{q} \in \mathcal{U}_{\mathbf{p}} \mapsto \pi_{\theta(\mathbf{q})}(\mathbf{q})$ is well-defined and C^∞ smooth.*

Proof. Let $\mathbf{p} \in \mathcal{V}^{n,d}$ and $\theta : \mathbf{q} \in \mathbb{C}[\mathbf{x}]_d \to q \in \mathbb{C}^n$ where q is the first left singular vector of the SVD decomposition of $H_{\mathbf{q}}^{1,d-1}$. Let $\gamma : \mathbb{C}[\mathbf{x}]_d \to \mathcal{V}^{n,d} = \psi \circ \theta$ be the composition map by the parametrization map ψ of $\mathcal{V}^{n,d}$.

By construction, we have $\rho : \mathbf{q} \mapsto \langle \mathbf{q}, \gamma(\mathbf{q}) \rangle_d \gamma(\mathbf{q})$. Let $\mathcal{O}$ denotes the open set of homogeneous polynomials $\mathbf{q} \in \mathbb{C}[\mathbf{x}]_d$ such that the Hankel matrix $H_{\mathbf{q}}^{1,d-1}$ has a nonzero gap between the first and the second singular values. It follows from [46] that the map θ is well-defined and smooth on $\mathcal{O}$. As $\mathbf{p}$ is in $\mathcal{V}^{n,d}$ and $H_{\mathbf{p}}^{1,d-1}$ is of rank 1, $\mathbf{p} \in \mathcal{O}$. Let $\mathcal{U}_{\mathbf{p}}$ be a neighborhood of $\mathbf{p}$ in $\mathbb{C}[\mathbf{x}]_d$ such that $\psi_{|\mathcal{U}_{\mathbf{p}}}$ is well-defined and smooth. As the apolar product $\langle \cdot, \cdot \rangle_d$ and the multiplication are well-defined and smooth on $\mathbb{C}[\mathbf{x}]_d \times \mathbb{C}[\mathbf{x}]_d$, ρ is well-defined and smooth on $\mathcal{U}_{\mathbf{p}}$, which ends the proof. □

As $\psi : v \in \mathbb{C}^n \mapsto (v^t \mathbf{x})^d \in \mathcal{V}^{n,d}$ is a parametrization of the Veronese variety $\mathcal{V}^{n,d}$, the tangent space of $\mathcal{V}^{n,d}$ at a point $\psi(v)$ is spanned by the first order vectors $D\psi(v)\, q$ of the Taylor expansion of $\psi(v + t\, q) = \psi(v) + t\, D\psi(v)\, q + O(t^2)$ for $q \in \mathbb{C}^n$. We are going to use this observation to prove the rigidity property of R_p.

Proposition 4.14. *For* $\mathbf{p} \in \mathcal{V}^{n,d}, \mathbf{q} \in T_{\mathbf{p}}(\mathcal{V}^{n,d})$*,*

$$\mathbf{p} + t\,\mathbf{q} - R_{\mathbf{p}}(t\,\mathbf{q}) = O(t^2).$$

Proof. As $\mathbf{p} \in \mathcal{V}^{n,d}, \mathbf{q} \in T_{\mathbf{p}}\mathcal{V}^{n,d}$, there exist $v, q \in \mathbb{C}^n$ such that $\mathbf{p} = \psi(v), \mathbf{q} = D\psi(v)\, q$. In particular, we have $\mathbf{p} + t\,\mathbf{q} - \psi(v + t\, q) = O(t^2)$. This implies that $H^{1,d-1}_{\mathbf{p}+t\,\mathbf{q}} - H^{1,d-1}_{\psi(v+t\,q)} = O(t^2)$. By differentiability of simple non-zero singular values and their singular vectors [47], we have $u_t - v_t = O(t^2)$ where $u_t = \theta(\mathbf{p} + t\,\mathbf{q})$ and $v_t = \theta(\psi(v + t\, q))$ are respectively the first left singular vectors of $H^{1,d-1}_{\mathbf{p}+t\,\mathbf{q}}$ and $H^{1,d-1}_{\psi(v+t\,q)}$.

Since $H^{1,d-1}_{\psi(v+t\,q)}$ is a matrix of rank 1 and its image is spanned by $v + t\, q$, v_t is a non-zero scalar multiple of $v + t\, q$ and we have $\pi_{v_t} = \pi_{v+t\,q}$. By continuity of the projection on a line, we have

$$\pi_{u_t}(\mathbf{p} + t\,\mathbf{q}) = \pi_{v_t}(\mathbf{p} + t\,\mathbf{q}) + O(t^2) = \pi_{v+t\,q}(\mathbf{p} + t\,\mathbf{q}) + O(t^2).$$

Since $\psi(v + t\, q) = \psi(v) + t\, D\psi(v)q + O(t^2) = \mathbf{p} + t\,\mathbf{q} + O(t^2)$, we have

$$\pi_{v+t\,q}(\mathbf{p} + t\,\mathbf{q}) = \pi_{v+t\,q}(\psi(v + t\, q)) + O(t^2) = \psi(v + t\, q) + O(t^2).$$

We deduce that

$$\begin{aligned}
\mathbf{p} + t\,\mathbf{q} - R_{\mathbf{p}}(t\,\mathbf{q}) &= \mathbf{p} + t\,\mathbf{q} - \pi_{u_t}(\mathbf{p} + t\,\mathbf{q}) \\
&= \mathbf{p} + t\,\mathbf{q} - \psi(v + t\, q) + (\psi(v + t\, q) - \pi_{v_t}(\mathbf{p} + t\,\mathbf{q})) \\
&\quad + (\pi_{v_t}(\mathbf{p} + t\,\mathbf{q}) - \pi_{u_t}(\mathbf{p} + t\,\mathbf{q})) \\
&= \psi(v) + t\, D\psi(v)q - \psi(v + t\, q) + O(t^2) = O(t^2),
\end{aligned}$$

which proves the proposition. □

Proposition 4.15. *Let* $\mathbf{p} \in \mathcal{V}^{n,d}$*. The map* $R_{\mathbf{p}} : T_{\mathbf{p}}\mathcal{V}^{n,d} \to \mathcal{V}^{n,d},\ \ \mathbf{q} \mapsto R_{\mathbf{p}}(\mathbf{q}) = \pi_{\theta(\mathbf{p}+\mathbf{q})}(\mathbf{p} + \mathbf{q})$ *is a retraction operator on the Veronese manifold* $\mathcal{V}^{n,d}$*.*

Proof. We have to prove that $R_{\mathbf{p}}$ verifies the three properties in definition 4.1.

1. $R_{\mathbf{p}}(0_{\mathbf{p}}) = \pi_{\theta(\mathbf{p})}(\mathbf{p} + 0_{\mathbf{p}}) = \pi_{\theta(\mathbf{p})}(\mathbf{p}) = \mathbf{p}$, by using lemma 4.12.
2. Let $S_{\mathbf{p}} : T_{\mathbf{p}}\mathcal{V}^{n,d} \to \mathbb{C}[\mathbf{x}]_d$, $\mathbf{q} \mapsto \mathbf{p} + \mathbf{q}$. The map $S_{\mathbf{p}}$ is well-defined and smooth on $T_{\mathbf{p}}\mathcal{V}^{n,d}$. By proposition 4.13, π is well-defined and smooth in a neighborhood $\mathcal{U}_{\mathbf{p}}$ of $\mathbf{p} \in \mathcal{V}^{n,d}$. Thus $R_{\mathbf{p}} = \rho \circ S_{\mathbf{p}}$ is well-defined and smooth in a neighborhood $\mathcal{U}'_{\mathbf{p}} \subset T\mathcal{V}^{n,d}$ of $0_{\mathbf{p}}$.
3. By proposition 4.14,
$$(\mathbf{p} + t\mathbf{q}) - R_{\mathbf{p}}(t\,\mathbf{q}) = O(t^2),$$
which implies that $\frac{d}{dt}R_{\mathbf{p}}(t\,\mathbf{q}) \mid_{t=0}= \mathbf{q}$, or equivalently $DR_{\mathbf{p}}(0_{\mathbf{p}})\mathbf{q} = \mathbf{q}$. Therefore we have $DR_{\mathbf{p}}(0_{\mathbf{p}}) = id_{T_{\mathbf{p}}\mathcal{V}^{n,d}}$.

□

4.3. Adding a trust-region scheme (c.f. [41, Chapter 7])

Recall that Riemannian Newton (resp. Gauss–Newton) is looking for a critical point of a real-valued function f, without distinguishing between local minimizer, saddle point and local maximizer. Furthermore, the convergence of this algorithm may not occur from the beginning. For these reasons, a trust region scheme is usually added to such algorithm in order to enhance the algorithm, with the desirable properties of convergence to a local minimum, with a local superlinear rate of convergence. In fact, trust region method ensures that f decreases at each iteration, which reinforces, when convergence occurs, the possibility of finding a local minimizer. Nevertheless, a global convergence to a local minimizer from any initial points is not guaranteed even after adding the trust region scheme (see [41, Subsection 7.4.1] for the global convergence of Riemannian trust-region methods). We prove in proposition 4.17 that under regularity assumptions, a local convergence for the Riemannian–Newton algorithm with trust region scheme can be obtained. Studying global or further local convergence properties for the Riemannian Newton (resp. Gauss Newton) method with trust region scheme for the STA problem is beyond the scope of this article.

Let $\mathcal{M}$ denote the Riemannian manifold $\mathcal{N}_r$ in subsection 4.1 (resp. $\mathcal{V}_r$ in subsection 4.2), and let $y_k \in \mathcal{M}$. The idea is to approximate the objective function f to its second order Taylor series expansion in a ball of center $0_{y_k} \in T_{y_k}\mathcal{M}$ and radius Δ_k denoted by $B_{\Delta_k} := \{\eta \in T_{y_k}\mathcal{M} \mid ||\eta|| \leq \Delta_k\}$, and to solve the subproblem

$$\min_{\eta \in B_{\Delta_k}} m_{y_k}(\eta), \tag{9}$$

where $m_{y_k}(\eta) := f(y_k) + G_k^t\eta + \frac{1}{2}\eta^t H_k \eta$, G_k is the gradient of f at y_k and H_k is respectively the Hessian of f at y_k for the Riemannian Newton method and the Gauss–Newton approximation of f at y_k for the Riemannian Gauss–Newton method.

By solving (9), we obtain a solution $\eta_k \in T_{y_k}\mathcal{M}$. Accepting or rejecting the candidate new point $y_{k+1} = R_{y_k}(\eta_k)$ is based on the quotient $\rho_k = \frac{f(y_k)-f(y_{k+1})}{m_{y_k}(0)-m_{y_k}(\eta_k)}$.
If ρ_k exceeds 0.2 then the current point y_k is updated, otherwise the current point y_k remains unchanged.
The radius of the trust region Δ_k is also updated based on ρ_k. We choose to update the trust region as in [27] with a few changes.

Let $\Delta_{y_0} := 10^{-1}\sqrt{\frac{d}{r}\sum_{i=1}^r ||w_i^0||^2}$ in the Riemannian Newton iteration (resp. $\Delta_{y_0} := 10^{-1}$ $\sqrt{\frac{d}{r}\sum_{i=1}^r ||v_i^0||^{2d}}$ in the Riemannian Gauss–Newton iteration), $\Delta_{\max} := \frac{1}{2}||\mathbf{p}||_d$. We take the initial radius as $\Delta_0 = \min\{\Delta_{y_0}, \Delta_{\max}\}$, if $\rho_k > 0.6$ then the trust region is enlarged as follows: $\Delta_{k+1} = \min\{2||\eta_k||, \Delta_{\max}\}$. Otherwise the trust region is shrinked by taking $\Delta_{k+1} = \min\{(\frac{1}{3} + \frac{2}{3}(1 + e^{-14(\rho_k - \frac{1}{3})})^{-1})\Delta_k, \Delta_{\max}\}$.

We choose the so-called dogleg method to solve the subproblem (9) [48]. Let η_N be the Newton direction given by $H\eta_N = -G$, let η_c denote the Cauchy point given by $\eta_c = -\frac{G^tG}{G^tHG}G$, and let η_I be the intersection of the boundary of the sphere B_Δ and the vector pointing from η_c to η_N. Then the optimal solution η^* of (9) by the dogleg method is given as follows:

$$\eta^* = \begin{cases} \eta_N & \text{if } ||\eta_N|| \leq \Delta, \\ -\frac{\Delta}{||G||}G & \text{if } ||\eta_N|| > \Delta \text{ and } ||\eta_c|| \geq \Delta, \\ \eta_I & \text{otherwise.} \end{cases}$$

The algorithm of the Riemannian Newton (resp. Gauss–Newton) method with trust region scheme for the STA problem is denoted by RNE-N-TR (resp. RGN-V-TR) and is given in pseudo-code by algorithm 1.

Algorithm 1 Riemannian Newton (resp. Gauss–Newton) algorithm with trust region sheme for the STA problem "RNE-N-TR"(resp. "RGN-V-TR")

```
Input: The homogeneous polynomial $\mathbf{p} \in \mathbb{C}[\mathbf{x}]_d$ associated to the symmetric tensor to approximate, $r < r_g$.
Choose initial point $y_0 \in \mathcal{N}_r$ (resp. $y_0 \in \mathcal{V}_r$).
while the method has not converged do
    1. Compute the gradient vector and the Hessian matrix (resp. Gauss–Newton Hessian approximation);
    2. Solve the subproblem (9) for the search direction $\eta_k \in B_{\Delta_k}$ by using the dogleg method;
    3. Compute the candidate next new point $y_{k+1} = R_{y_k}(\eta_k)$;
    4. Compute the quotient $\rho_k$;
    5. Accept or reject $y_{k+1}$ based on the quotient $\rho_k$;
    6. Update the trust region radius $\Delta_k$.
end while
Output: $y_* \in \mathcal{N}_r$ (resp. $y_* \in \mathcal{V}_r$).
```

The algorithm 1 is stopped when $\Delta_k \leq \Delta_{\min}$ (by default $\Delta_{\min} = 10^{-3}$), or when the maximum number of iterations exceeds $N_{\max}$.

Remark 4.16. In order to handle ill-conditioned Hessian (resp. Gauss–Newton Hessian approximation) matrices in algorithm 1, we use the Moore-Penrose pseudoinverse [49, 50, 51]. This can appear in cases where some vectors v_i of the rank-r approximation span close lines, which yields a singularity problem in the iteration. In particular, this is the case when the symmetric border rank of the symmetric tensor is not equal to its symmetric rank [27, 37], [40, section 2.4]. For example, the tensor $\mathbf{p} = (v_0^t\mathbf{x})(v_1^t\mathbf{x})^{d-1} + \epsilon T$, with $v_0, v_1 \in \mathbb{R}^n$, $T \in \mathbb{R}[\mathbf{x}]_d$ and ϵ very small, is close to the tensor $(v_0^t\mathbf{x})(v_1^t\mathbf{x})^{d-1} = \lim_{\delta\to 0} \frac{1}{d\,\delta}(((v_1 + \delta v_0)^t\mathbf{x})^d - (v_1^t\mathbf{x})^d)$ of border rank 2 and symmetric rank d. It can be very well approximated by a tensor of rank 2, with two vectors of almost the same direction.

Under some regularity assumption, it is possible to guarantee that RNE-N-TR algorithm converges to a local minimum of the distance function f.

Proposition 4.17. *Let $\mathbf{p} \in \mathbb{C}[\mathbf{x}]_d$, let $\mathbf{p}_0 \in \Sigma_r$ be the initial point of RNE-N-TR and let $B_0 = B(\mathbf{p}, ||\mathbf{p} - \mathbf{p}_0||_d)$ be the ball of center $\mathbf{p}$ and radius $||\mathbf{p} - \mathbf{p}_0||_d$ in $\mathbb{C}[\mathbf{x}]_d$. Assume that $B_0 \cap \overline{\Sigma}_r \subset \Sigma_r^{\mathrm{reg}}$ (i.e. all points of $\overline{\Sigma}_r$ in B_0 are non-defective), then RNE-N-TR converges to a local minimum $y \in \mathcal{N}_r$ of the distance function f to Σ_r.*

Proof. Let $\Sigma_r^0 := B_0 \cap \overline{\Sigma}_r = B_0 \cap \Sigma_r^{\mathrm{reg}}$ be the set of non-defective tensors of rank r in B_0. As $\varphi_r : (W, V_R, V_I) \in \mathcal{N}_r \mapsto \sum_{i=1}^r w_i((v_{R,i} + \mathbf{i}\, v_{I,i})^t\mathbf{x})^d \subset \overline{\Sigma}_r \subset \mathbb{C}[\mathbf{x}]_d$ is locally injective at a non-defective tensor, it defines a local diffeomorphism between $\mathcal{N}_{r,0} = \varphi_r^{-1}(\Sigma_r^0)$ and Σ_r^0. As B_0 is compact, $\mathcal{N}_{r,0} = \varphi_r^{-1}(\Sigma_r^0)$ is a compact Riemannian manifold. By construction, the distance between $\mathbf{p}$ and the iterates $\mathbf{p}_i$ is decreasing in RNE-N-TR, so that their decomposition is in $\mathcal{N}_{r,0} = \varphi_r^{-1}(\Sigma_r^{\mathrm{reg}} \cap B_0)$. As $\mathcal{N}_{r,0}$ is a compact Riemannian manifold and f is smooth on $\mathcal{N}_{r,0}$ (as a polynomial function), [41, Corollary 7.4.6] implies that the iterates of RNE-N-TR of Riemannian Newton method with a trust region sheme on $\mathcal{N}_{r,0}$ converge to a local minimum of the distance function f. □

The regularity assumption $B_0 \cap \overline{\Sigma}_r \subset \Sigma_r^{\mathrm{reg}}$ implies that the ball centered at $\mathbf{p}$ and containing the initial point of the iteration does not contain a defective tensor. In this case, the iterates, which distance to $\mathbf{p}$ decreases, remain in the ball and the limit decomposition is a non-defective low rank tensor. This assumption, satisfied when $\mathbf{p}$ is far enough from the singular locus of $\overline{\Sigma}_r$, is a sufficient condition to ensure the regularity of the iteration points and their limit.

5. Numerical experiments

In this section, we present four numerical experiments using the RNE-N-TR and RGN-V-TR algorithms. These algorithms are implemented in the package `TensorDec.jl`[1]. We use a Julia implementation for the method SPM tested in subsection 5.4. The solvers from Tensorlab v3 [52] are run in MATLAB 7.10. The experimentation was done on a Dell Windows desktop with 8 GB memory and Intel Core i5-5300U, 2.3 GHz CPU.

5.1. Choice of the initial point

The choice of the initial point is a crucial step in iterative methods. We use the direct algorithm of [35], based on the computation of generalized eigenvectors and generalized eigenvalues of pencils of Hankel matrices (see also [36]), to compute an initial rank-r approximation. This algorithm, denoted SMD, works only with $r < r_g$ such that $\iota \leq \lfloor \frac{d-1}{2} \rfloor$ where ι denotes the interpolation degree of the points in the rank-r decomposition [53, Chapter 4]. This implies that $r < \binom{n+d'-1}{d'}$ where $d' = \lfloor \frac{d-1}{2} \rfloor$. It first computes a SVD decomposition of the Hankel matrix of the tensor $\mathbf{t}$ in degree $\left(\lfloor \frac{d-1}{2} \rfloor, d - \lfloor \frac{d-1}{2} \rfloor\right)$, extracts the first r singular vectors, computes a simultaneous diagonalisation of the matrices of multiplication by the variables x_i by taking a random combination of them, computing its eigenvectors and deducing the points and weights in the approximate decomposition of $\mathbf{t}$. The rationale behind choosing the initial point with this method is when the symmetric tensor is already of symmetric rank r with $r < r_g$ and $\iota \leq \lfloor \frac{d-1}{2} \rfloor$, then this computation gives a good numerical approximation of the exact decomposition, so that the Riemannian Newton (resp. Gauss–Newton) algorithm needs few iterations to converge numerically. We will see in the following numerical experiments that this initial point is an efficient choice to get a good low rank approximation of a symmetric tensor.

5.2. Best rank-1 approximation and spectral norm

Let $\mathbf{p} \in \mathcal{S}^d(\mathbb{R}^n)$, a best real rank-1 approximation of $\mathbf{p}$ is a minimizer of the optimization problem

$$\mathrm{dist}_1(\mathbf{p}) := \min_{\mathbf{t} \in \mathcal{S}^d(\mathbb{R}^n), \mathrm{rank}_s(\mathbf{t})=1} ||\mathbf{p} - \mathbf{t}||_d^2 = \min_{(w,v) \in \mathbb{R} \times \mathbb{S}^{n-1}} ||\mathbf{p} - w(v^t \mathbf{x})^d||_d, \tag{10}$$

where $\mathbb{S}^{n-1} = \{v \in \mathbb{R}^n \mid ||v|| = 1\}$ is the unit sphere. This problem is equivalent to $\min_{\mathbf{t} \in \mathcal{T}^d(\mathbb{R}^n), \mathrm{rank}(\mathbf{t})=1} ||\mathbf{p} - \mathbf{t}||_F^2$ since at least one global minimizer is a symmetric rank-1 tensor [31].

[1] It can be obtained from https://gitlab.inria.fr/AlgebraicGeometricModeling/TensorDec.jl and run in Julia version 1.1.1. See functions rne_n_tr and rgn_v_tr.

The real spectral norm of $\mathbf{p} \in \mathcal{S}^d(\mathbb{R}^n)$, denoted by $||\mathbf{p}||_{\sigma,\mathbb{R}}$ is by definition:

$$||\mathbf{p}||_{\sigma,\mathbb{R}}^2 := \max_{v \in \mathbb{S}^{n-1}} |\mathbf{p}(v)|. \tag{11}$$

The two problems (10) and (11) are related by the following equality:

$$\mathrm{dist}_1(\mathbf{p})^2 = ||\mathbf{p}||_d^2 - ||\mathbf{p}||_{\sigma,\mathbb{R}}^2,$$

which we deduce by simple calculus and properties of the apolar norm (see also [12, 31]):

$$\begin{aligned}
\mathrm{dist}_1(\mathbf{p})^2 &= \min_{(w,v)\in\mathbb{R}\times\mathbb{S}^{n-1}} ||\mathbf{p} - w(v^t\mathbf{x})^d||_d^2 \\
&= \min_{(w,v)\in\mathbb{R}\times\mathbb{S}^{n-1}} ||\mathbf{p}||_d^2 - 2\langle \mathbf{p}, w(v^t\mathbf{x})^d\rangle_d + ||w(v^t\mathbf{x})^d||_d^2 \\
&= \min_{(w,v)\in\mathbb{R}\times\mathbb{S}^{n-1}} ||\mathbf{p}||_d^2 - 2w\,\mathbf{p}(v) + w^2 \\
&= \min_{v\in\mathbb{S}^{n-1}} ||\mathbf{p}||_d^2 - |\mathbf{p}(v)|^2 = ||\mathbf{p}||_d^2 - \max_{v\in\mathbb{S}^{n-1}} |\mathbf{p}(v)|^2 = ||\mathbf{p}||_d^2 - ||\mathbf{p}||_{\sigma,\mathbb{R}}^2.
\end{aligned}$$

Therefore, if v is a global maximizer of (11) such that $w = \mathbf{p}(v)$, then $w\,v^{\otimes d}$ is a best rank-1 approximation of $\mathbf{p}$. Herein, a rank-1 approximation $w\,v^{\otimes d}$, such that $w = \mathbf{p}(v)$ and $||v|| = 1$, is better when $|w|$ is higher. Therefore, in the following experimentation, we report the weight w obtained by the different methods.

In [32] the authors present an algorithm called "SDP" based on semidefinite relaxations to find a best real rank-1 approximation of a real symmetric tensor by finding a global optimum of $\mathbf{p}$ on $\mathbb{S}^{n-1}$. We choose two examples from [32], on which we apply the RNE-N-TR with initial point chosen according to the SMD algorithm adapted for 1×1 matrices. The reason behind using RNE-N-TR instead of RGN-V-TR is to take advantage of the local quadratic rate of convergence that distinguishes the exact Riemannian Newton iteration in RNE-N-TR [41, Theorem 6.3.2]. We compare these methods with the method CCPD-NLS which is a non-linear least-square solver for the symmetric decomposition from Tensorlab v3 [52] in MATLAB 7.10, where we run 50 instances (i.e. 50 random initial points obeying Gaussian distributions), and we take the absolute value of the weight in average for this method.

We denote by $|w_{\text{sdp}}|$ (resp. $|w_{\text{rne}}|$) the weight in absolute value given by SDP (resp. RNE-N-TR), and $|w_{\text{ccpd}}|$ denotes the absolute value of the weight in average given by CCPD-NLS. Note that $|w_{\text{sdp}}|$ is the spectral norm of $\mathbf{p}$, since SDP gives a best rank-1 approximation. We report the time spent by SDP from [32] (resp. RNE-N-TR including the computation time of the initial point) in seconds (s) and we denote it by t_{sdp} (resp. t_{rne}). We denote by N_{rne} the number of iterations in RNE-N-TR. We denote by d_0 the norm between $\mathbf{p}$ and the initial point of RNE-N-TR, and by d_* the norm between $\mathbf{p}$ and the solution obtained by RNE-N-TR. We denote by t_{ccpd} (resp. N_{ccpd}) the time in seconds (s) (resp. number of iterations) in average for CCPD-NLS.

Example 5.1. [32, Example 3.5]. Consider the tensor $\mathbf{p} \in \mathcal{S}^3(\mathbb{R}^n)$ with entries:

$$(\mathbf{p})_{i_1,i_2,i_3} = \frac{(-1)^{i_1}}{i_1} + \frac{(-1)^{i_2}}{i_2} + \frac{(-1)^{i_3}}{i_3},$$

corresponding to the polynomial $\mathbf{p} = \sum_{|\alpha|=3}(\sum_{i=1}^n \alpha_i \frac{(-1)^i}{i}) \binom{3}{\alpha} \mathbf{x}^\alpha$.

Example 5.2. [32, Example 3.7]. Consider the tensor $\mathbf{p} \in \mathcal{S}^5(\mathbb{R}^n)$ given as:

$$(\mathbf{p})_{i_1,\ldots,i_5} = (-1)^{i_1}\log(i_1) + \cdots + (-1)^{i_5}\log(i_5),$$

corresponding to the polynomial $\mathbf{p} = \sum_{|\alpha|=5}(\sum_{i=1}^n \alpha_i(-1)^i \log(i)) \binom{5}{\alpha} \mathbf{x}^\alpha$.

Table 1: Comparison of RNE-N-TR, CCPD-NLS and SDP for Example 5.1 and Example 5.2.

	Example 5.1					Example 5.2				
n	10	20	30	40	50	5	10	15	20	25
$\lvert w_{\text{rne}}\rvert$	**17.8**	**34.2**	**50.1**	**65.9**	**81.6**	**1.100e+2**	**8.833e+2**	**2.697e+3**	**6.237e+3**	**11.504e+3**
d_0	32.4	28.4	44	64.6	78.3	526.1	6.559e+3	26.318e+3	64.268e+3	132.213e+3
d_*	13.2	28.3	43.8	59.5	75.3	477.5	6.096e+3	24.643e+3	60.435e+3	121.892e+3
t_{rne}	0.038	0.304	1.5	3.3	12.1	0.058	0.282	3.8	18.3	34.8
N_{rne}	5	4	4	4	6	5	4	6	6	6
$\lvert w_{\text{ccpd}}\rvert$	14.0	29.3	43.3	60.0	75.6	78.9	8.68e+2	2.354e+3	6.148e+3	10.587e+3
t_{ccpd}	0.173	0.109	0.105	0.122	0.143	0.093	0.187	1.2	5.5	16.7
N_{ccpd}	27	25	22	23	22	19	29	16	23	17
$\lvert w_{\text{sdp}}\rvert$	**17.8**	**34.2**	**50.1**	**65.9**	**81.6**	**1.100e+2**	**8.833e+2**	**2.697e+3**	**6.237e+3**	
t_{sdp}	2.0	6.0	30.0	245.0	1965.0	1.0	22.0	78.0	1350.0	

The results in Table 1 show that the RNE-N-TR finds a global minimizer, starting from the initial point given by the SMD algorithm. The RNE-N-TR algorithm converges to this point in few iterations, and with very reduced time compared to the SDP algorithm especially when n grows. On the other hand, $|w_{\text{ccpd}}|$ is smaller than $|w_{\text{sdp}}|$, implying that CCPD computes, in several cases, a local minimum, which is not a global minimum i.e. a best rank-1 approximation. In comparison for these cases, RGN-V-TR took more iterations (∼20) than RNE-N-TR and consequently more time, while reaching the same optimimum.

The fact that RNE-N-TR finds the best rank-1 approximation in these examples comes from the good initial point provided by SMD algorithm. However, we have no guarantee that RNE-N-TR with this initial point will always converge to a best rank-1 approximation. This experimentation shows that RNE-N-TR combined with SMD algorithm for the initial point is an efficient method to get a good real rank-1 approximation of a real symmetric tensor.

5.3. Symmetric rank-r approximation

We consider two examples of a real and a complex valued sparse symmetric tensors, in order to compare the performance of RNE-N-TR and RGN-V-TR with state-of-the-art non-linear least-square solvers CCPD-NLS and SDF-NLS for symmetric decomposition from Tensolab v3 with random initial point following a standard normal distribution. These solvers employ factor matrices as parameterization and use a Gauss–Newton method with dogleg trust region steps called "NLS-GNDL". We fix 200 iterations as maximal number of iterations, and we run 50 instances for these methods and we report the minimal, median and maximal residual error denoted 'err', such that, err := $||\mathbf{p} - \mathbf{p}_*||_d$, where $\mathbf{p}$ is the symmetric tensor to approximate and $\mathbf{p}_*$ is the approximate symmetric tensor of rank-r. In the computation of the initial point by SMD algorithm in RNE-N-TR and RGN-V-TR, we compute eigenvectors of a random linear combination of multiplication operators. This computation is sensitive to the choice of the linear combination, when the operators are not commuting, which explains why we report also the minimal, median and maximal err for these two methods. The average of time t is in seconds, and the average number of iterations N is rounded to the closest integer.

Example 5.3. Let $\mathbf{p} \in \mathcal{S}^3(\mathbb{R}^{10})$ such that:

$$(\mathbf{p})_{i_1,i_2,i_3} = \begin{cases} i_1^2 + 1 & \text{if } i_1 = i_2 = i_3, \\ 1 & \text{if } [i_1, i_2, i_3] \equiv [i, i, j] \text{ with } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

($[i_1, i_2, i_3] \equiv [j_1, j_2, j_3]$ iff there exists a permutation $\sigma \in S_3$ such that $[i_{\sigma(1)}, i_{\sigma(2)}, i_{\sigma(3)}] = [j_1, j_2, j_3]$). This sparse symmetric tensor corresponds to the polynomial $\mathbf{p} = \sum_{i=1}^{10} i^2 x_i^3 + (\sum_{i=1}^{10} x_i^2) \times (\sum_{i=1}^{10} x_i)$.

Example 5.4. Let $\mathbf{p} \in \mathcal{S}^3(\mathbb{C}^{10})$ such that:

$$(\mathbf{p})_{i_1,i_2,i_3} = \begin{cases} e^{\sqrt{i_1} + i_1^2\sqrt{-1}} + \frac{i_1}{10}\sqrt{-1} & \text{if } i_1 = i_2 = i_3, \\ \frac{i}{10}\sqrt{-1} & \text{if } [i_1, i_2, i_3] \equiv [i, i, j] \text{ with } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

This sparse symmetric tensor corresponds to the polynomial $\mathbf{p} = \sum_{i=1}^{10} e^{\sqrt{i} + i^2\sqrt{-1}} x_i^3 + \sqrt{-1}(\sum_{i=1}^{10} \frac{i}{10} x_i^2) \times (\sum_{i=1}^{10} x_i)$.

Table 2: Comparison of **RNE-N-TR**, **RGN-V-TR**, **CCPD-NLS**, **SDF-NLS** for Examples 5.3 and 5.4.

Example 5.3						Example 5.4					
r	$\mathrm{err}_{\mathrm{rne}}$			t_{rne}	N_{rne}	r	$\mathrm{err}_{\mathrm{rne}}$			t_{rne}	N_{rne}
	min	med	max	avg	avg		min	med	max	avg	avg
3	70.6	96	134.3	0.03	2	3	22.4	28.8	30.9	0.04	2
5	33.3	54.2	91.8	0.08	3	5	14.1	17.4	24.6	0.07	3
10	0.884	0.884	94.1	0.465	6	10	0.164	0.168	0.369	0.113	2
r	$\mathrm{err}_{\mathrm{rgn}}$			t_{rgn}	N_{rgn}	r	$\mathrm{err}_{\mathrm{rgn}}$			t_{rgn}	N_{rgn}
	min	med	max	avg	avg		min	med	max	avg	avg
3	70.6	96	136.8	0.064	3	3	22.4	27.6	36.1	0.065	3
5	33.3	48.8	105.3	0.149	4	5	14.1	17.1	24.6	0.101	3
10	0.886	0.886	10.1	0.836	7	10	0.162	0.164	0.169	0.219	2
r	$\mathrm{err}_{\mathrm{ccpd}}$			t_{ccpd}	N_{ccpd}	r	$\mathrm{err}_{\mathrm{ccpd}}$			t_{ccpd}	N_{ccpd}
	min	med	max	avg	avg		min	med	max	avg	avg
3	71	102	137.1	0.067	14	3	22.9	26.8	35.2	0.084	14
5	34.2	54.7	121	0.116	26	5	14.9	17	26.6	0.104	18
10	7.8	7.8	9.7	0.5	90	10	4.8	4.8	11.2	0.506	60
r	$\mathrm{err}_{\mathrm{sdf}}$			t_{sdf}	N_{sdf}	r	$\mathrm{err}_{\mathrm{sdf}}$			t_{sdf}	N_{sdf}
	min	med	max	avg	avg		min	med	max	avg	avg
3	71	96.3	136	0.155	14	3	22.9	27.4	35.2	0.254	15
5	34.2	49.4	105.3	0.212	16	5	14.9	17.8	26.5	0.35	19
10	7.8	8.2	38.3	2.3	158	10	4.8	6.2	12.6	2.5	144

The numerical results in Table 2 show that the number of iterations of RNE-N-TR and RGN-V-TR method is low compared to the other methods. The iterations in RNE-N-TR and RGN-V-TR are more expensive. The numerical quality of approximation is better for RNE-N-TR and RGN-V-TR than the other methods in this test. It is of the same order as the other methods for $r = 3, 5$ but much better for $r = 10$. This can be explained by the fact that the initial point provided by SMD method is close to a good rank-10 approximation. Notice that when $r = 3, 5$ the initial point provided by SMD method, based on truncated SVD and eigenvector computations, yields the same behavior as a random initial point (a random linear combination of the matrices of a pencil is used to compute the eigenvectors in SMD method).

5.4. Approximation of perturbations of low rank symmetric tensors

In this section, we consider perturbations of random low rank tensors. For a given rank r, we choose r random vectors v_i of size n, obeying Gaussian distributions and compute the symmetric tensor $\mathbf{t} = \sum_{i=1}^{r} (v_i^t \mathbf{x})^d$ of order d. We choose a random symmetric tensor $\mathbf{t}_{\text{err}}$ of order d, with coefficients also obeying Gaussian distributions, normalize it so that its apolar norm is ϵ and add it to $\mathbf{t}$: $\tilde{\mathbf{t}} = \mathbf{t} + \epsilon \frac{\mathbf{t}_{\text{err}}}{\|\mathbf{t}_{\text{err}}\|_d}$. We apply the different approximation algorithms to $\tilde{\mathbf{t}}$ and compute the relative error factor ref $:= \frac{\|\mathbf{t}_* - \mathbf{t}\|_d}{\epsilon}$ between the approximation $\mathbf{t}_*$ of rank r computed by the algorithm and the rank-r tensor $\mathbf{t}$. We run this computation for 100 random instances and report the geometric average of the relative error. The average number of iterations N is rounded to the closest integer, and the average time t is in seconds.

As the initial tensor $\tilde{\mathbf{t}}$ is in a ball of radius ϵ centered at the tensor $\mathbf{t}$ of rank r, we expect $\mathbf{t}_*$ to be at distance to $\mathbf{t}$ smaller than ϵ and the relative error factor to be less than 1.

We compare the RNE-N-TR and RGN-V-TR methods with the initial point computed by SMD algorithm, with the recent Subspace Power Method (SPM) of [34] and the state-of-the-art implementation CPD-NLS of the package Tensorlab v3. Note that CPD-NLS is designed for the canonical polyadic decomposition [1]. Nevertheless, in practice it is often observed that applying a general tensor rank approximation method (like CPD-NLS) from a symmetric starting point will usually result in a symmetric approximation. Since CPD-NLS is an efficient tensor decomposition routine of Tensorlab v3, we choose to compare our methods with this algorithm in this numerical experiment, using symmetric initial points and verifying that the obtained tensor approximations are symmetric. As SPM works for even order tensors with real coefficients, the comparison in Table 3 is run for tensors in $\mathcal{S}^4(\mathbb{R}^{10})$. In Table 4, we compare CPD-NLS, RNE-N-TR, and RGN-V-TR for tensors in $\mathcal{S}^d(\mathbb{C}^{10})$ of order $d = 4$ and with complex coefficients. These tables also provide a numerical comparison with the low rank approximation methods tested in Example 5.4 of [33], since the setting is the same. We also run this tensor perturbation test on some complex examples in which the approximation rank is higher than the mode size of the tensor (see Table 5). We test this with the three methods RNE-N-TR, RGN-V-TR, and CPD-NLS. We run 20 instances, for each example of tensor and ϵ.

The computational time for the methods RNE-N-TR and RGN-V-TR includes the computation of the initial point by the SMD algorithm. We fix 200 iterations as maximal number of iterations for RNE-N-TR, RGN-V-TR and CPD-NLS. For SPM, the iterations are stopped when the distance between two consecutive iterates is less than 10^{-10} or when the maximal number of iterations ($N = 400$ in this experimentation) is reached.

In Tables 3, 4, the number of iterations of the RNE-N-TR and RGN-V-TR methods is significantly smaller than the number of iterations of the other methods. In SPM, the number of iterations to get an approximation of a single rank-1 term of the approximation is about 30, indicating a practical linear convergence as predicted by the theory [34, Theorem 5.10]. As the method CPD-NLS is based on a quasi-Newton iteration, its local convergence is subquadratic, which also explains the relatively high number of iterations. The low number of iterations in RNE-N-TR and RGN-V-TR can be explained by the choice of the initial point by SMD algorithm. This provides a good initialization such that a solution by RNE-N-TR and RGN-V-TR can be obtained in a few number of iterations.

The cost of an iteration appears to be higher in RNE-N-TR and RGN-V-TR than in the other methods. Nevertheless, the total time is of the same order. Note that the cost of an iteration seems higher in RGN-V-TR than RNE-N-TR. Despite the fact that the first algorithm

Table 3: Computational results of **SPM**, **RNE-N-TR**, and **RGN-V-TR** for rank-r approximations in $\mathcal{S}^4(\mathbb{R}^{10})$.

r	ϵ	$\mathrm{ref}_{\mathrm{spm}}$	t_{spm}	N_{spm}	$\mathrm{ref}_{\mathrm{rne}}$	t_{rne}	N_{rne}	$\mathrm{ref}_{\mathrm{rgn}}$	t_{rgn}	N_{rgn}
1	1	0.103	0.04	28	0.105	0.07	2	0.11	0.083	3
	10^{-1}	0.103	0.039	28	0.104	0.04	2	0.11	0.069	3
	10^{-2}	0.1	0.04	28	0.1	0.04	2	0.103	0.058	2
	10^{-4}	0.101	0.041	29	0.101	0.041	2	0.166	0.044	2
	10^{-6}	0.104	0.041	30	0.104	0.041	2	0.17	0.045	2
2	1	0.15	0.1	69	0.175	0.137	3	0.159	0.16	3
	10^{-1}	0.15	0.091	65	0.153	0.076	2	0.159	0.13	3
	10^{-2}	0.144	0.086	66	0.149	0.072	2	0.15	0.111	2
	10^{-4}	0.148	0.089	66	0.157	0.076	2	0.199	0.076	2
	10^{-6}	0.146	0.087	67	0.151	0.073	2	0.195	0.073	2
3	1	0.185	0.126	109	0.194	0.172	3	0.194	0.208	3
	10^{-1}	0.185	0.135	111	0.195	0.128	2	0.195	0.198	3
	10^{-2}	0.187	0.119	113	0.208	0.099	2	0.195	0.175	2
	10^{-4}	0.182	0.102	106	0.197	0.092	2	0.217	0.095	2
	10^{-6}	0.183	0.101	105	0.196	0.094	2	0.206	0.097	2
4	1	0.217	0.159	168	0.25	0.546	8	0.225	0.278	3
	10^{-1}	0.218	0.161	168	0.245	0.319	4	0.228	0.241	3
	10^{-2}	0.211	0.163	162	0.241	0.134	2	0.219	0.239	3
	10^{-4}	0.216	0.167	169	0.26	0.128	2	0.261	0.136	2
	10^{-6}	0.227	0.167	168	0.259	0.126	2	0.259	0.133	2
5	1	0.244	0.207	217	0.339	1.199	13	0.252	0.594	5
	10^{-1}	0.244	0.221	220	0.255	0.252	2	0.252	0.317	3
	10^{-2}	0.247	0.223	218	0.292	0.175	2	0.254	0.321	3
	10^{-4}	0.246	0.215	213	0.304	0.16	2	0.304	0.165	2
	10^{-6}	0.249	0.231	226	0.307	0.158	2	0.311	0.165	2

Table 4: Computational results of **CPD-NLS**, **RNE-N-TR**, and **RGN-V-TR** for rank-r approximations in $\mathcal{S}^4(\mathbb{C}^{10})$.

r	ϵ	$\mathrm{ref}_{\mathrm{cpd}}$	t_{cpd}	N_{cpd}	$\mathrm{ref}_{\mathrm{rne}}$	t_{rne}	N_{rne}	$\mathrm{ref}_{\mathrm{rgn}}$	t_{rgn}	N_{rgn}
1	1	0.117	0.05	10	0.11	0.06	2	0.115	0.069	3
	10^{-1}	0.118	0.046	10	0.108	0.054	2	0.112	0.084	3
	10^{-2}	0.116	0.043	10	0.107	0.044	2	0.11	0.06	2
	10^{-4}	0.114	0.042	10	0.107	0.037	2	0.227	0.038	2
	10^{-6}	0.113	0.037	11	0.112	0.036	2	0.237	0.037	2
2	1	0.167	0.072	14	0.162	0.078	2	0.166	0.118	3
	10^{-1}	0.169	0.077	14	0.164	0.063	2	0.167	0.111	3
	10^{-2}	0.162	0.071	14	0.163	0.061	2	0.163	0.09	2
	10^{-4}	0.171	0.071	14	0.163	0.062	2	0.204	0.063	2
	10^{-6}	0.175	0.069	13	0.162	0.062	2	0.23	0.064	2
3	1	0.201	0.115	16	0.204	0.135	2	0.204	0.163	3
	10^{-1}	0.223	0.109	17	0.206	0.091	2	0.203	0.157	3
	10^{-2}	0.228	0.117	17	0.209	0.086	2	0.203	0.152	2
	10^{-4}	0.202	0.103	15	0.205	0.091	2	0.243	0.093	2
	10^{-6}	0.284	0.124	19	0.211	0.088	2	0.234	0.091	2
4	1	0.235	0.149	18	0.234	0.192	3	0.234	0.23	3
	10^{-1}	0.232	0.165	19	0.244	0.132	2	0.238	0.215	3
	10^{-2}	0.237	0.142	17	0.25	0.113	2	0.232	0.219	3
	10^{-4}	0.238	0.158	19	0.25	0.112	2	0.255	0.117	2
	10^{-6}	0.232	0.161	19	0.254	0.111	2	0.274	0.116	2
5	1	0.275	0.21	22	0.261	0.269	3	0.261	0.345	3
	10^{-1}	0.264	0.186	19	0.269	0.211	2	0.261	0.288	3
	10^{-2}	0.266	0.211	22	0.305	0.148	2	0.264	0.292	3
	10^{-4}	0.265	0.169	18	0.293	0.158	2	0.299	0.163	2
	10^{-6}	0.266	0.206	21	0.298	0.158	2	0.301	0.161	2

computes the Gauss–Newton approximation of the Hessian matrix, whereas the second algorithm computes the exact Hessian matrix. This can be explained by the use of a parametrization in the first algorithm (i.e. the Cartesian product of Veronese manifolds), which involves a more expensive retraction using SVD decomposition on larger matrices.

These experimentation also show a good numerical behavior for the Riemannian methods. In particular, the numerical quality of the low rank approximation is good for RNE-N-TR and RGN-V-TR, in comparison with SPM and CPD-NLS. The average of the relative error factor in RNE-N-TR and RGN-V-TR is less than 1. The numerical results in [33, Example 5.4] for GP method and small perturbations ($\epsilon \in \{10^{-2}, 10^{-4}, 10^{-6}\}$), show that the numerical quality in GP-OPT method is worse than with these methods.

Table 5: Computational results of **CPD-NLS**, **RNE-N-TR**, and **RGN-V-TR**.

d n r	ϵ	ref_{cpd}		t_{cpd}	N_{cpd}	ref_{rne}		t_{rne}	N_{rne}	ref_{rgn}		t_{rgn}	N_{rgn}
		min	max	avg	avg	min	max	avg	avg	min	max	avg	avg
5 4 10	1	0.803	7.8	2.1	162	0.834	25.7	3.4	273	0.815	1.1	1.2	49
	10^{-2}	0.849	881.2	2.3	172	0.718	0.933	0.197	16	0.718	0.933	0.078	4
	10^{-4}	1.5	9.8e+4	2	157	0.711	0.933	0.0379	3	0.711	0.933	0.0535	3
	10^{-6}	776.4	1.9e+7	2	184	0.789	0.912	0.042	3	0.789	0.912	0.071	4
5 15 20	1	0.15	1.9e+3	9.8	45	0.153	0.172	22.6	3	0.153	0.172	27.1	3
	10^{-2}	0.149	1.2e+5	12.7	62	0.151	0.183	13.6	2	0.148	0.169	26.9	3
	10^{-4}	0.152	9.2e+6	13.8	67	0.152	0.181	13.6	2	0.152	0.181	14.7	2
	10^{-6}	0.155	1.4e+9	11.9	59	0.156	0.173	13.8	2	0.156	0.174	14.8	2
6 5 12	1	0.515	109.7	1.4	61	0.467	0.706	0.342	4	0.467	0.622	0.353	4
	10^{-2}	0.519	2.4e+4	3.8	143	0.472	0.62	0.155	3	0.472	0.62	0.205	3
	10^{-4}	0.518	9.6e+5	2.9	137	0.493	0.622	0.222	4	0.493	0.622	0.35	5
	10^{-6}	1.1	9.7e+7	2.3	112	0.647	0.6	0.098	2	0.492	0.591	0.211	3
7 8 15	1	0.183	2.1e+3	54.9	46	0.171	0.21	8.3	3	0.171	0.21	8.5	3
	10^{-2}	0.174	5.3e+4	52.3	47	0.137	0.171	4.7	2	0.169	0.201	8.1	3
	10^{-4}	0.168	3.5e+6	63.1	54	0.138	0.169	4.5	2	0.138	0.169	4.7	2
	10^{-6}	0.179	1.1e+9	75.6	65	0.142	0.177	4.4	2	0.142	0.177	4.6	2

We also compare CPD-NLS, RNE-N-TR and RGN-V-TR for perturbation of random tensors of rank $r > n$ and report the minimal and maximal relative error with the average number of iterations N (rounded to the closest integer) and the average time t (in seconds) in Table 5. The considered cases in Table 5 are for the degree d, the number of variables n and the rank r such that (d, n, r) is respectively (5,4,10), (5,15,20), (6,5,12), and (7,8,15). We see that the maximal relative error factor *ref* reached by RNE-N-TR and RGN-V-TR with initial point by SMD is less than 1. There is an exception in the first case when $\epsilon = 1$, where a large number of iterations is needed for RNE-N-TR and RGN-V-TR. On the other hand, the minimal relative error of CPD-NLS is less than 1 in almost all Table 5, whereas its maximal relative error is higher than 1 in all Table 5.

This numerical experiment indicates that for these examples of random low rank tensors with random noise, SMD provides a good initial point, close enough to a good solution, so that RNE-N-TR and RGN-V-TR need a few number of iterations. In this context, the combination of an adaptive choice of initial point and a Newton-type method is successful.

5.5. *Symmetric tensor with large differences in the scale of the weight vector*

Consider the case of a real symmetric tensor $\mathbf{t} = \sum_{i=1}^{r} w_i (v_i^t \mathbf{x})^d$, $\|v_i\| = 1$, $w_i > 0$, with large differences in the scale of the weights w_i i.e. $\frac{\max_i w_i}{\min_i w_i}$ is large. More precisely, there are large differences in the norms of the rank-1 symmetric tensors $w_i (v_i^t \mathbf{x})^d$. We randomly sample real symmetric tensors of order $d = 3$ and dimension $n = 7$ with $r \in \{5, 10\}$, according to the following model:

$$\mathbf{t} = \sum_{i=1}^{r} 10^{\frac{is}{r}} (v_i^t \mathbf{x})^d, \ \|v_i\| = 1.$$

The components of the weight vector increase exponentially from $10^{\frac{s}{r}}$ to 10^{s}.

We aim to compare the performance of RNE-N-TR and RGN-V-TR methods (hereafter called respectively RNE and RGN for shortness) in this configuration. We run the following test:

- Take $\mathbf{t}$ as above, and create a perturbated tensor $\mathbf{t}_p = \frac{\mathbf{t}}{\|\mathbf{t}\|} + 10^{-5}\frac{\mathbf{t}_{\text{err}}}{\|\mathbf{t}_{\text{err}}\|}$, where $\mathbf{t}_{\text{err}} \in \mathbb{R}[\mathbf{x}]_d$ is a random symmetric tensor with coefficients obeying Gaussian distributions;
- run 20 random initial points obeying Gaussian distributions;
- run RNE and RGN with a maximum of iterations $N_{\max} = 500$, and report in average respectively: the relative error (in geometric average) $\text{err}_{\text{rel}} := \left\|\frac{\mathbf{t}}{\|\mathbf{t}\|} - \mathbf{t}_*\right\|_d$, where $\mathbf{t}_*$ is a rank-r symmetric decomposition obtained by these methods; the number of iterations N_{iter}; and the computation time t in seconds (s). We also report the number N_{opt} of instances where $\text{err}_{\text{rel}} \leq 1.1.10^{-5}$.

Table 6: Computational results for RNE-N-TR and RGN-V-TR for scaled weights.

$r = 5$						
s	1		2		3	
Alg	RNE	RGN	RNE	RGN	RNE	RGN
err_{rel}	0.456	5.8e-6	0.411	5.5e-6	0.246	1.4e-5
N_{iter}	120	39	165	61	175	77
t	2.0	1.1	2.4	1.4	2.5	1.8
N_{opt}	0	20	0	20	0	17
$r = 10$						
s	1		2		3	
Alg	RNE	RGN	RNE	RGN	RNE	RGN
err_{rel}	0.372	9.4e-6	0.195	1.6e-6	0.224	6.8e-5
N_{iter}	423	87	270	186	392	206
t	16.9	6.3	10.8	13.7	15.5	15.0
N_{opt}	0	18	0	20	0	9

The results in Table 6 show that RGN outperforms RNE. In fact, the average of the relative error in RGN is better, up to five order of magnitude, than in RNE. Moreover, starting from the same 20 random initial points in the two methods; RGN succeeded to reach an optimum, at least in 9 instances with the different order of scale s, while RNE could not find any optimum. Notice that, as we mentionned before, the cost of one iteration in RGN is higher than in RNE. The good performance of RGN compared to RNE in this test was expected, since the orthonormal basis of the tangent space computed in RGN method is independent of the weight factor. This behavior was also observed in [27, Subsection 3.4] for real multilinear tensors, parametrized by Segre manifolds.

6. Conclusion

We presented two Riemannian Newton optimization methods for approximating a given complex-valued symmetric tensor by a low rank symmetric tensor. We used in subsection 4.1 the weighted normalized factor matrices parametrization for the constraint set. We developed an exact Riemannian Newton iteration with exact computation of the Hessian matrix (RNE-N-TR). We exploited in subsection 4.1.1 the properties of the apolar product and of partial complex derivatives, to deduce a simplified and explicit computation of the gradient and Hessian of the square distance function in terms of the points, weights of the decomposition and the tensor to approximate. We proved that under some regularity conditions on non-defective tensors in the neighborhood of the initial point, the iteration is converging to a local minimum. In subsection 4.2, we parametrized the constraint set via Cartesian product of Veronese manifolds. Taking into account the geometry of the Veronese manifold, we constructed a suitable basis for its tangent space at a given point on this manifold. Using this basis, we developed a Gauss–Newton iteration (RGN-V-TR). In subsection 4.2.1, we presented a retraction operator on the Veronese manifold. We showed that, combined with SMD method for choosing the initial point, the two methods have a good practical behavior in several experiments: in subsection 5.2 to compute a best real rank-1 approximation of a real symmetric tensor, in subsection 5.3 to compute a low rank approximation of sparse symmetric tensors, and in subsection 5.4 to compute low rank approximations of random perturbations of low rank symmetric tensors. In subsection 5.5, we showed that the numerical behavior of RNE-N-TR is affected by large differences in the scaling of the rank-1 symmetric tensor, where RGN-V-TR outperformed this algorithm in this case.

In future work, we plan to investigate the computation of initial points for the Riemannian Newton iterations applied to tensors of higher rank and the low rank approximation problem for other families of tensors, such as multi-symmetric or skew symmetric tensors.

7. Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments that improved this article.

References

[1] F. L. Hitchcock, The expression of a tensor or a polyadic as a sum of products, Journal of Mathematics and Physics 6 (1-4) (1927) 164–189.

[2] J. Alexander, A. Hirschowitz, Polynomial interpolation in several variables, Vol. 4, 1995, pp. 201–222.

[3] L. Chiantini, G. Ottaviani, N. Vannieuwenhoven, On generic identifiability of symmetric tensors of subgeneric rank, Transactions of the American Mathematical Society 369 (6) (2016) 4021–4042.

[4] P. Comon, Tensor decompositions, state of the art and applications, in: J. G. McWhirter, I. K. Proudler (Eds.), Mathematics in Signal Processing V, Clarendon Press, Oxford, 2002, pp. 1–24.

[5] P. Comon, M. Rajih, Blind identification of under-determined mixtures based on the characteristic function, Signal Processing 86 (9) (2006) 2271 – 2281, special Section: Signal Processing in UWB Communications.

[6] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM Journal on Matrix Analysis and Applications 21 (4) (2000) 1253–1278.

[7] A. Smilde, R. Bro, P. Geladi, Multi-way Analysis with Applications in the Chemical Sciences, John Wiley, West Sussex, UK, 2004.

[8] R. Khouja, P.-A. Mattei, B. Mourrain, Tensor decomposition for learning Gaussian mixtures from moments (2021). arXiv:2106.00555.

[9] E. S. Allman, C. Matias, J. A. Rhodes, Identifiability of parameters in latent structure models with many observed variables, Annals of Statistics 37 (6A) (2009) 3099–3132.

[10] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, M. Telgarsky, Tensor decompositions for learning latent variable models, Journal of Machine Learning Research 15 (2014) 2773–2832.

[11] L. D. Garcia, M. Stillman, B. Sturmfels, Algebraic geometry of Bayesian networks, Journal of Symbolic Computation 39 (3-4) (2005) 331–355.

[12] L. De Lathauwer, B. De Moor, J. Vandewalle, On the best rank-1 and rank-(R_1, R_2, . . ., R_n) approximation of higher-order tensors, SIAM Journal on Matrix Analysis and Applications 21 (4) (2000) 1324–1342.

[13] N. Vannieuwenhoven, R. Vandebril, K. Meerbergen, A new truncation strategy for the higher-order singular value decomposition, SIAM Journal on Scientific Computing 34 (2) (2012) A1027–A1052.

[14] D. Kressner, M. Steinlechner, B. Vandereycken, Low-rank tensor completion by Riemannian optimization, BIT Numerical Mathematics 54 (2) (2014) 447–468.

[15] J. D. Carroll, J.-J. Chang, Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition, Psychometrika 35 (3) (1970) 283–319.

[16] B. Chen, S. He, Z. Li, S. Zhang, Maximum block improvement and polynomial optimization, SIAM Journal on Optimization 22 (1) (2012) 87–107.

[17] R. Harshman, Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis, UCLA Working Papers in Phonetics 16 (1970) 1–84.

[18] T. G. Kolda, B. W. Bader, Tensor decompositions and applications, SIAM Review 51 (3) (2009) 455–500.

[19] M. Espig, W. Hackbusch, A. Khachatryan, On the convergence of alternating least squares optimisation in tensor format representations, arXiv preprint arXiv:1506.00062 (2015).

[20] A. Uschmajew, Local convergence of the alternating least squares algorithm for canonical tensor approximation, SIAM Journal on Matrix Analysis and Applications 33 (2) (2012) 639–652.

[21] C. Hayashi, F. Hayashi, A new algorithm to solve Parafac-model, Behaviormetrika 9 (11) (1982) 49–60.

[22] P. Paatero, The multilinear Engine–A Table-Driven, least squares program for solving multilinear problems, including the n-way parallel factor analysis model, Journal of Computational and Graphical Statistics 8 (4) (1999) 854–888.

[23] A.-H. Phan, P. Tichavský, A. Cichocki, Low complexity damped Gauss–Newton algorithms for CANDECOMP/PARAFAC, SIAM Journal on Matrix Analysis and Applications 34 (1) (2013) 126–147.

[24] B. Savas, L.-H. Lim, Quasi-Newton methods on Grassmannians and multilinear approximations of tensors, SIAM Journal on Scientific Computing 32 (6) (2010) 3352–3393.

[25] L. Sorber, M. Van Barel, L. De Lathauwer, Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(l_r, l_r, 1)$ terms, and a new generalization, SIAM Journal on Optimization 23 (2) (2013) 695–720.

[26] G. Tomasi, R. Bro, A comparison of algorithms for fitting the PARAFAC model, Computational Statistics & Data Analysis 50 (7) (2006) 1700–1734.

[27] P. Breiding, N. Vannieuwenhoven, A Riemannian trust region method for the canonical tensor rank approximation problem, SIAM Journal on Optimization 28 (3) (2018) 2435–2465.

[28] W. Hackbusch, Tensor Spaces and Numerical Tensor Calculus, Springer Series in Computational Mathematics, Springer Berlin Heidelberg, 2012.

[29] P. Breiding, N. Vannieuwenhoven, The condition number of join decompositions, SIAM Journal on Matrix Analysis and Applications 39 (1) (2018) 287–309.

[30] L. Sorber, M. V. Barel, L. D. Lathauwer, Unconstrained optimization of real functions in complex variables, SIAM Journal on Optimization 22 (3) (2012) 879–898.

[31] X. Zhang, C. Ling, L. Qi, The best rank-1 approximation of a symmetric tensor and related spherical optimization problems, SIAM Journal on Matrix Analysis and Applications 33 (3) (2012) 806–821.

[32] J. Nie, L. Wang, Semidefinite relaxations for best rank-1 tensor approximations, SIAM Journal on Matrix Analysis and Applications 35 (3) (2014) 1155–1179.

[33] J. Nie, Low rank symmetric tensor approximations, SIAM Journal on Matrix Analysis and Applications 38 (4) (2017) 1517–1540.

[34] J. Kileel, J. M. Pereira, Subspace power method for symmetric tensor decomposition and generalized PCA (2019). arXiv:1912.04007.

[35] J. Harmouch, H. Khalil, B. Mourrain, Structured low rank decomposition of multivariate Hankel matrices, Linear Algebra and Its Applications 542 (2018) 161–185.

[36] B. Mourrain, Polynomial-exponential decomposition from moments, Foundations of Computational Mathematics 18 (6) (2018) 1435–1492.

[37] P. Comon, G. Golub, L.-H. Lim, B. Mourrain, Symmetric tensors and symmetric tensor rank, SIAM Journal on Matrix Analysis and Applications 30 (3) (2008) 1254–1279.

[38] J.Harris, Algebraic Geometry: A First Course, Graduate Texts in Mathematics, Springer-Verlag, New York, NY, 1998.

[39] F. Zak, Tangents and Secants of Algebraic Varieties, Translations of Mathematical Monographs, AMS, Providence, RI, 1993.

[40] J. Landsberg, Tensors: Geometry and Applications, Graduate studies in mathematics, American Mathematical Society, 2011.

[41] P.-A. Absil, R. Mahony, R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, Princeton, NJ, 2008.

[42] D. Kressner, M. Steinlechner, B. Vandereycken, Low-rank tensor completion by Riemannian optimization, BIT Numerical Mathematics 54 (2) (2014) 447–468.

[43] R. L. Adler, J. Dedieu, J. Y. Margulies, M. Martens, M. Shub, Newton's method on Riemannian manifolds and a geometric model for the human spine, IMA Journal of Numerical Analysis 22 (3) (2002) 359–390.

[44] R. Remmert, R. Burckel, Theory of Complex Functions, Graduate Texts in Mathematics, Springer New York, 1991.

[45] P.-A. Absil, J. Malick, Projection-like retractions on matrix manifolds, SIAM Journal on Optimization 22 (1) (2012) 135–158.

[46] J.-L. Chern, L. Dieci, Smoothness and periodicity of some matrix decompositions, SIAM Journal on Matrix Analysis and Applications 22 (3) (2001) 772–792.

[47] G. W. Stewart, Matrix Algorithms: Volume II: Eigensystems, Society for Industrial and Applied Mathematics, 2001.

[48] J. Nocedal, S. Wright, Numerical Optimization, 2nd Edition, Springer series in operations research and financial engineering, Springer, New York, NY, 2006.

[49] A. Bjorck, Numerical Methods for Least Squares Problems, Society for Industrial and Applied Mathematics, 1996.

[50] K. Konstantinides, K. Yao, Statistical analysis of effective singular values in matrix rank determination, IEEE Transactions on Acoustics, Speech, and Signal Processing 36 (5) (1988) 757–763.

[51] G. W. Stewart, Rank degeneracy, SIAM Journal on Scientific and Statistical Computing 5 (2) (1984) 403–413.

[52] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, L. De Lathauwer, Tensorlab 3.0 (Mar. 2016).
URL https://www.tensorlab.net

[53] D. Eisenbud, The Geometry of Syzygies: A Second Course in Commutative Algebra and Algebraic Geometry, Springer, 2005.

[54] P. A. Absil, R. Mahony, J. Trumpf, An extrinsic look at the Riemannian Hessian, in: F. Nielsen, F. Barbaresco (Eds.), Geometric Science of Information, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 361–368.

[55] D. H. Brandwood, A complex gradient operator and its application in adaptive array theory, IEE Proceedings F: Communications Radar and Signal Processing 130 (1) (1983) 11–16.

[56] Z. Nehari, Introduction to Complex Analysis, Allyn & Bacon, 1968.

Appendix A. Computation details

This appendix gives first the proof of proposition 4.4 which relates the Riemannian gradient and Hessian to the real gradient and Hessian (Appendix A.1), then the proof of the explicit formulas of the real gradient (Appendix A.2.1) and Hessian (Appendix A.2.2) stated respectively in propositions 4.5 and 4.6.

Appendix A.1. Proof of proposition 4.4

Let $y = (w, v_1, \dots, v_r, v'_1, \dots, v'_r) \in \mathcal{N}_r$. Let $\mathcal{P}_y$ be the orthogonal projector on $T_y\mathcal{N}_r$. Let $Q \in \mathbb{R}^{(r+2nr)\times(r+(2n-1)r)}$ such that its columns form an orthonormal basis of the image of $\mathcal{P}_y$ or equivalently of $T_y\mathcal{N}_r$. As the Riemannian gradient of f is the projection of Df_R, the first order differentials of f_R, on the tangent space $T_y\mathcal{N}_r$ [41, Chapter 5], we have $G = Q^t G^R$, where G^R is the vector which represents the classical first order partial derivatives of f_R at y^R in the canonical basis.
Let $\eta \in T_y\mathcal{N}_r$, $z \in T_y\mathcal{N}_r^{\perp}$. We have from [54] that the Riemannian Hessian matrix of f at y is given by the formula: $H\eta = \mathcal{P}_y H^R \eta + \mathfrak{U}_y(\eta, \mathcal{P}_y^{\perp} G^R)$, where H^R is the matrix of the second order derivatives of f_R at y^R in the canonical basis, $\mathfrak{U}_y$ is the Weingarten map on $\mathcal{N}_r$ at y given by $\mathfrak{U}_y(\eta, z) = \mathcal{P}_y D_\eta \mathcal{P} z$, where $\mathcal{P}$ is a matrix valued function on $\mathcal{N}_r$ determined as follows: $\mathcal{P} : y \in \mathcal{N}_r \mapsto \mathcal{P}_y$, and $D_\eta \mathcal{P} z$ represent the time derivative of $y \mapsto \mathcal{P}_y z$ in terms of the time derivative of y i.e. $\dot{y} \in T_y\mathcal{N}_r$ applied at $\dot{y} = \eta$, and $\mathcal{P}_y^{\perp} = I - \mathcal{P}_y$ is the orthogonal projector on $T_y\mathcal{N}_r^{\perp}$.
As $y \in \mathcal{N}_r$ we have $w \in {\mathbb{R}_+^*}^r$, and $\check{v}_i := (v_i, v'_i) \in \mathbb{S}^{2n-1}$, $\forall 1 \le i \le r$. Let $u = (u_0, u_1, \dots, u_r, u'_1, \dots, u'_r) \in \mathbb{R}^{r+2nr}$, such that $\check{u}_i = (u_i, u'_i)$, $\forall 1 \le i \le r$. Let $\mathcal{P}_w$ (resp. $\mathcal{P}_{\check{v}_i}$) denote the orthogonal projector on $T_w(\mathbb{R}_+^*)^r = \mathbb{R}^r$ (resp. $T_{\check{v}_i}\mathbb{S}^{2n-1}$), we have that: $\mathcal{P}_w(u_0) = u_0$, $\mathcal{P}_{\check{v}_i}\check{u}_i = (I_{2n} - \check{v}_i\check{v}_i^t)\check{u}_i$, $\forall 1 \le i \le r$, thus:

$$\mathcal{P}_y u = \begin{pmatrix} u_0 \\ ((I_{2n} - \check{v}_1\check{v}_1^t)\check{u}_1)[1:n] \\ \vdots \\ ((I_{2n} - \check{v}_r\check{v}_r^t)\check{u}_r)[1:n] \\ ((I_{2n} - \check{v}_1\check{v}_1^t)\check{u}_1)[n+1:2n] \\ \vdots \\ ((I_{2n} - \check{v}_r\check{v}_r^t)\check{u}_r)[n+1:2n] \end{pmatrix} = \begin{pmatrix} u_0 \\ u_1 - v_1\check{v}_1^t\check{u}_1 \\ \vdots \\ u_r - v_r\check{v}_r^t\check{u}_r \\ u'_1 - v'_1\check{v}_1^t\check{u}_1 \\ \vdots \\ u'_r - v'_r\check{v}_r^t\check{u}_r \end{pmatrix}, \mathcal{P}_y^{\perp} u = \begin{pmatrix} 0_r \\ v_1\check{v}_1^t\check{u}_1 \\ \vdots \\ v_r\check{v}_r^t\check{u}_r \\ v'_1\check{v}_1^t\check{u}_1 \\ \vdots \\ v'_r\check{v}_r^t\check{u}_r \end{pmatrix}.$$

Let $\mathfrak{U}_{\check{v}_i}$ be the Weingarten map on $\mathbb{S}^{2n-1}$ at $\check{v}_i$. For $\eta = (\eta_0, \eta_1, \dots, \eta_r, \eta'_1, \dots, \eta'_r) \in T_y\mathcal{N}_r$, and $z = (z_0, z_1, \dots, z_r, z'_1, \dots, z'_r) \in T_y\mathcal{N}_r^{\perp}$ with $\check{\eta}_i = (\eta_i, \eta'_i) \in T_{\check{v}_i}\mathbb{S}^{2n-1}$ and $\check{z}_i = (z_i, z'_i) \in T_{\check{v}_i}{\mathbb{S}^{2n-1}}^{\perp}$, $\forall 1 \le i \le r$, we have from [54]: $\mathfrak{U}_{\check{v}_i}(\check{\eta}_i, \check{z}_i) = -\check{\eta}_i\check{v}_i^t\check{z}_i$. Thus, with respect to the parameterization that we consider we find that:

$$\mathfrak{U}_y(\eta, z) = -\begin{pmatrix} 0_r \\ \eta_1 \check{v}_1^t \check{z}_1 \\ \vdots \\ \eta_r \check{v}_r^t \check{z}_r \\ \eta'_1 \check{v}_1^t \check{z}_1 \\ \vdots \\ \eta'_r \check{v}_r^t \check{z}_r \end{pmatrix}.$$

Let $G^R = (g_0; g_1; \ldots; g_r; g'_1; \ldots; g'_r) \in \mathbb{R}^{r+2nr}$ and $\check{g}_i = (g_i, g'_i)$, $l_i = \check{v}_i \check{v}_i^t \check{g}_i$ for $i = 1, \ldots, r$. We obtain $\mathfrak{U}_y(\eta, \mathcal{P}_y^{\perp} G^R)$ by substituting $\check{z}_i$ by l_i in $\mathfrak{U}_y(\eta, z)$. Since $\check{v}_i^t \check{v}_i = ||\check{v}_i||^2 = 1$, we find that

$$\mathfrak{U}_y(\eta, \mathcal{P}_y^{\perp} G^R) = \begin{pmatrix} 0_r \\ \eta_1 \check{v}_1^t \check{g}_1 \\ \vdots \\ \eta_r \check{v}_r^t \check{g}_r \\ \eta'_1 \check{v}_1^t \check{g}_1 \\ \vdots \\ \eta'_r \check{v}_r^t \check{g}_r \end{pmatrix} = S\eta, \text{ where } S = \operatorname{diag}(0_{r\times r}, \tilde{S}, \tilde{S}), \text{ with } \tilde{S} = \operatorname{diag}(s_1 I_n, \ldots, s_r I_n),$$

$s_i = \check{v}_i^t \check{g}_i = \langle v_i, g_i \rangle + \langle v'_i, g'_i \rangle$. Since, $\mathfrak{U}_y(\eta, z) = \mathcal{P}_y D_\eta \mathcal{P} z$, and $\mathcal{P}_y \circ \mathcal{P}_y = \mathcal{P}_y$, we can write $\mathfrak{U}_y(\eta, z) = \mathcal{P}_y \mathfrak{U}_y(\eta, z)$. Hence, $\mathfrak{U}_y(\eta, \mathcal{P}_y^{\perp} G^R) = \mathcal{P}_y S \eta = \mathcal{P}_y S \mathcal{P}_y \eta$, since $\mathcal{P}_y \eta = \eta$ for $\eta \in T_y \mathcal{N}_r$. Thus we have: $H\eta = \mathcal{P}_y (H^R + S) \mathcal{P}_y \eta$, and then $H = \mathcal{P}_y (H^R + S) \mathcal{P}_y$. Herein, H can be written with respect to the basis Q as follows: $H = Q^t (H^R + S) Q$, which ends the proof.

Appendix A.2. Real gradient and Hessian

In order to give the proofs of propositions 4.5 and 4.6, we need the following discussion and auxiliary lemma.

We describe the real gradient and Hessian, by using complex variables and their conjugates. Recall from Brandwood [55] that transforming the pair $(\Re(z), \Im(z))$ of real and imaginary parts of a given complex variable z into the pair $(z, \overline{z})$ is a simple linear transformation, which allows us to achieve explicit and simple computation of the gradient and Hessian of f.

Recall that $\mathcal{R}_r = \{(W, \Re(V), \Im(V)) \in \mathbb{R}^r \times \mathbb{R}^{n\times r} \times \mathbb{R}^{n\times r} \mid W \in \mathbb{R}^r, V \in \mathbb{C}^{n\times r}\}$, and that f_R is the function f seen as a function on $\mathcal{R}_r$.

Let $\mathcal{C}_r = \left\{(W, V, \overline{V}) \in \mathbb{R}^r \times \mathbb{C}^{n\times r} \times \mathbb{C}^{n\times r} \mid W \in \mathbb{R}^r,\ \ V \in \mathbb{C}^{n\times r}\right\}$ and

$$K = \begin{bmatrix} I_r & 0_{r\times 2nr} \\ 0_{2nr\times r} & J \end{bmatrix} \tag{A.1}$$

where $J = \begin{bmatrix} I_{nr} & \mathbf{i} I_{nr} \\ I_{nr} & -\mathbf{i} I_{nr} \end{bmatrix}$. The linear map K is an isomorphism between the $\mathbb{R}$-vector spaces $\mathcal{R}_r$ and $\mathcal{C}_r$. Its inverse is given by $K^{-1} = \begin{bmatrix} I_r & 0_{r\times 2nr} \\ 0_{2nr\times r} & \frac{1}{2} J^* \end{bmatrix}$.

Let f_C be the function f seen as a function on $\mathcal{C}_r$. Considering f_C for the computation of the gradient and the Hessian yields more elegant expressions than considering f_R. For this reason, we compute first the gradient and the Hessian of f_C, and then we use the isomorphism K in (A.1) to get the real gradient and the Hessian of f_R.

Lemma A.1. *The complex gradient G^C can be transformed into the real gradient G^R as follows:*

$$G^R = K^t G^C. \tag{A.2}$$

Similarly H^R and H^C are related by the following formula:

$$H^R = K^t H^C K. \tag{A.3}$$

Proof. See [30] and the references therein. □

We can now present the proofs of propositions 4.5 and 4.6.

Appendix A.2.1. ***Proof of proposition 4.5***

We can write f_C as $f_C = \frac{1}{2}(f_1 - f_2 - f_3 + f_4)$, where

$$f_1 = \Big|\Big|\sum_{i=1}^{r} w_i (v_i^t \mathbf{x})^d\Big|\Big|_d^2 = \sum_{|\alpha|=d} \binom{d}{\alpha}\Big(\sum_{i=1}^{r} w_i \bar{v}_i^{\alpha}\Big)\Big(\sum_{i=1}^{r} w_i v_i^{\alpha}\Big) \quad \text{(by definition 2.1)},$$

$$f_2 = \Big\langle \sum_{i=1}^{r} w_i (v_i^t \mathbf{x})^d, \mathbf{p}\Big\rangle_d = \sum_{i=1}^{r} w_i \mathbf{p}(\bar{v}_i) \quad \text{(by 1. in lemma 2.2)},$$

$$f_3 = \bar{f}_2 = \sum_{i=1}^{r} w_i \bar{\mathbf{p}}(v_i), \text{and } f_4 = ||\mathbf{p}||_d^2.$$

Let us decompose G^C as $G^C = \begin{pmatrix} G_1 \\ \tilde{G}_2 \\ \tilde{G}_3 \end{pmatrix}$, with $G_1 = (\frac{\partial f_C}{\partial w_j})_{1\le j\le r}$, $\tilde{G}_2 = (\frac{\partial f_C}{\partial v_j})_{1\le j\le r}$ and $\tilde{G}_3 = (\frac{\partial f_C}{\partial \bar{v}_j})_{1\le j\le r}$. As f_C is a real valued function, we have that $\frac{\partial f_C}{\partial \bar{v}_j} = \overline{\frac{\partial f_C}{\partial v_j}}$ [56, 44], thus $\tilde{G}_3 = \overline{\tilde{G}}_2$. Let us start by the computation of G_1:

$$\begin{aligned}
\frac{\partial f_1}{\partial w_j} &= \frac{\partial}{\partial w_j}\Big(\sum_{|\alpha|=d}\binom{d}{\alpha}\Big(\sum_{i=1}^{r} w_i \bar{v}_i^{\alpha}\Big)\Big(\sum_{i=1}^{r} w_i v_i^{\alpha}\Big)\Big) \\
&= \sum_{|\alpha|=d}\binom{d}{\alpha}\Big(\bar{v}_j^{\alpha}\Big(\sum_{i=1}^{r} w_i v_i^{\alpha}\Big) + v_j^{\alpha}\Big(\sum_{i=1}^{r} w_i \bar{v}_i^{\alpha}\Big)\Big) \\
&= \sum_{i=1}^{r} w_i (v_j^* v_i)^d + \sum_{i=1}^{r} w_i (v_i^* v_j)^d = 2\sum_{i=1}^{r} w_i \Re((v_j^* v_i)^d);
\end{aligned}$$

the third equality is deduced by using definition 2.1 and 1. of lemma 2.2. In addition, we have $\frac{\partial f_2}{\partial w_j} = \frac{\partial}{\partial w_j}(\sum_{i=1}^{r} w_i \mathbf{p}(\bar{v}_i)) = \mathbf{p}(\bar{v}_j)$, $\frac{\partial f_3}{\partial w_j} = \bar{\mathbf{p}}(v_j)$, and $\frac{\partial f_4}{\partial w_j} = 0$. Thus, $\frac{\partial f_C}{\partial w_j} = \sum_{i=1}^{r} w_i \Re((v_j^* v_i)^d) - \Re(\bar{\mathbf{p}}(v_j))$.

Now, for the computation of $\tilde{G}_2$, let $\mathbf{p} = \sum_{|\alpha|=d}\binom{d}{\alpha}\check{v}_{\alpha}\mathbf{x}^{\alpha}$, and $1 \le k \le n$,

$$\begin{aligned}
\frac{\partial f_1}{\partial v_{j,k}} &= \sum_{|\alpha|=d}\binom{d}{\alpha}\Big(\sum_{i=1}^{r} w_i \bar{v}_i^{\alpha}\Big)(w_j \alpha_k v_j^{\alpha - e_k}) = w_j \sum_{i=1}^{r} w_i \langle \partial_{x_k}(v_i^t \mathbf{x})^d, (v_j^t \mathbf{x})^{d-1}\rangle_{d-1} \\
&= d w_j \sum_{i=1}^{r} w_i \langle (v_i^t \mathbf{x})^d, x_k (v_j^t \mathbf{x})^{d-1}\rangle_d = d w_j \sum_{i=1}^{r} w_i \bar{v}_{i,k}(v_i^* v_j)^{d-1},
\end{aligned}$$

the second (resp. third and fourth) equality are deduced by using lemma 2.2. Moreover, we have $\frac{\partial f_2}{\partial v_{j,k}} = 0$, $\frac{\partial f_3}{\partial v_{j,k}} = w_j \sum_{|\alpha|=d} \binom{d}{\alpha} \bar{v}_\alpha \alpha_k v_j^{\alpha-e_k} = w_j \partial_{x_k} \bar{\mathbf{p}}(v_j)$, and $\frac{\partial f_4}{\partial v_{j,k}} = 0$. Thus, $\frac{\partial f_C}{\partial v_j} = \frac{1}{2}\Big(d w_j \sum_{i=1}^r w_i (v_i^* v_j)^{(d-1)} \bar{v}_i - w_j \nabla_{\mathbf{x}} \bar{\mathbf{p}}(v_j)\Big)$.

We have $G^R = K^t G^C$ from (A.2). By multiplication of these two matrices, we obtain: $G^R = \begin{pmatrix} G_1 \\ \tilde{G}_2 + \overline{\tilde{G}_2} \\ \mathbf{i}(\tilde{G}_2 - \overline{\tilde{G}_2}) \end{pmatrix} = \begin{pmatrix} G_1 \\ 2\Re(\tilde{G}_2) \\ -2\Im(\tilde{G}_2) \end{pmatrix}$. Finally dividing by 2, we get $G^R = \begin{pmatrix} G_1 \\ \Re(G_2) \\ -\Im(G_2) \end{pmatrix}$, where $G_2 = 2\tilde{G}_2$, which ends the proof.

Appendix A.2.2. ***Proof of proposition 4.6***

H^C is given by the following block matrix:

$$H^C = \begin{bmatrix} \left[\frac{\partial^2 f_C}{\partial w_i \partial w_j}\right]_{1\le i,j\le r} & \left[\frac{\partial^2 f_C}{\partial w_i \partial v_j^t}\right]_{1\le i,j\le r} & \left[\frac{\partial^2 f_C}{\partial w_i \partial \bar{v}_j^t}\right]_{1\le i,j\le r} \\ \left[\frac{\partial^2 f_C}{\partial v_i \partial w_j}\right]_{1\le i,j\le r} & \left[\frac{\partial^2 f_C}{\partial v_i \partial v_j^t}\right]_{1\le i,j\le r} & \left[\frac{\partial^2 f_C}{\partial v_i \partial \bar{v}_j^t}\right]_{1\le i,j\le r} \\ \left[\frac{\partial^2 f_C}{\partial \bar{v}_i \partial w_j}\right]_{1\le i,j\le r} & \left[\frac{\partial^2 f_C}{\partial \bar{v}_i \partial v_j^t}\right]_{1\le i,j\le r} & \left[\frac{\partial^2 f_C}{\partial \bar{v}_i \partial \bar{v}_j^t}\right]_{1\le i,j\le r} \end{bmatrix}.$$

We have that $\frac{\partial^2 f}{\partial \bar{z} \partial \bar{z}^t} = \overline{\frac{\partial^2 f}{\partial z \partial z^t}}$, and $\frac{\partial^2 f}{\partial z \partial \bar{z}^t} = \overline{\frac{\partial^2 f}{\partial \bar{z} \partial z^t}}$, for a complex variable z and a real valued function with complex variables f. Using these two relations, we find that $\left[\frac{\partial^2 f_C}{\partial w_i \partial w_j}\right]_{1\le i,j\le r}$, $\left[\frac{\partial^2 f_C}{\partial v_i \partial w_j}\right]_{1\le i,j\le r}$, $\left[\frac{\partial^2 f_C}{\partial v_i \partial v_j^t}\right]_{1\le i,j\le r}$, and $\left[\frac{\partial f_C}{\partial \bar{v}_i \partial v_j^t}\right]_{1\le i,j\le r}$ determine H^C. We denote them respectively by A, $\tilde{B}$, $\tilde{C}$, and $\tilde{D}$. Herein, we can decompose H^C as:

$$H^C = \begin{bmatrix} A & \tilde{B}^t & \tilde{B}^* \\ \tilde{B} & \tilde{C} & \tilde{D}^t \\ \overline{\tilde{B}} & \tilde{D} & \overline{\tilde{C}} \end{bmatrix}.$$

The computation of these four matrices can be done by taking the formula of $\frac{\partial f_C}{\partial w_j}$ and $\frac{\partial f_C}{\partial v_j}$ obtained in the proof of proposition 4.5, and using the apolar identities in lemma 2.2. Using (A.3) we obtain: $H^R = \begin{bmatrix} A & 2\Re(\tilde{B})^t & -2\Im(\tilde{B})^t \\ 2\Re(\tilde{B}) & 2\Re(\tilde{C}+\tilde{D}) & -2\Im(\tilde{C}+\tilde{D}) \\ -2\Im(\tilde{B}) & 2\Im(\tilde{D}-\tilde{C}) & 2\Re(\tilde{D}-\tilde{C}) \end{bmatrix}$. Finally, for the simplification by 2, as in the previous proof, we redefine the formula of H^R as it is given in proposition 4.6, where B, C, and D are respectively equal to two times $\tilde{B}$,$\tilde{C}$, and $\tilde{D}$.