Noname manuscript No.
(will be inserted by the editor)

Global convergence of Riemannian line search methods with a Zhang-Hager-type condition

Harry Oviedo

Received: date / Accepted: date

Abstract In this paper, we analyze the global convergence of a general non–monotone line search method on Riemannian manifolds. For this end, we introduce some properties for the tangent search directions that guarantee the convergence, to a stationary point, of this family of optimization methods under appropriate assumptions. A modified version of the non–monotone line search of Zhang and Hager is the chosen globalization strategy to determine the step–size at each iteration. In addition, we develop a new globally convergent Riemannian conjugate gradient method that satisfies the direction assumptions introduced in this work. Finally, some numerical experiments are performed in order to demonstrate the effectiveness of the new procedure.

Keywords Descent method · non–monotone line search · inexact line–search · global convergence · Riemannian manifolds.

Mathematics Subject Classification (2010) 65K05 · 90C30 · 90C56 · 53C21.

1 Introduction

This paper focuses on the line–search methods for minimizing smooth functions on a Riemannian manifold. This class of optimization problems is mathematically formulated as follows

$$\min_{x\in\mathcal{M}} F(x), \tag{1}$$

where $F : \mathcal{M} \to \mathbb{R}$ is a continuously differentiable function and the pair $(\mathcal{M}, \langle,\rangle)$ is a Riemannian manifold with associated metric $\langle,\rangle$. Problem (1) arises in many applications, for example, joint diagonalization [1,22], nearest low–rank correlation matrix problem [13,34], sparse principal component analysis [19,30], Kohn–Sham total energy minimization [40,41], low–rank matrix completion [5,23,26], dimension reduction techniques in statistical data analysis [20], RGB image chromaticity denoising [21], training of unitary recurrent neural networks (RNNs) [3,24], among others.

The line–search methods in the Euclidean context ($\mathcal{M} = \mathbb{R}^n$), construct a sequence of points $\{x_k\} \subset \mathcal{M}$, where each x_{k+1} is updated using the parametrization of the line determined by the previous point x_k, and a search direction $d_k \in \mathbb{R}^n$, that is

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

where $\alpha_k > 0$ is the step–size, and d_k must satisfy the relation $\nabla F(x_k)^\top d_k < 0$, which guarantees that the vector d_k is a descent direction of $F(\cdot)$ at x_k. There are many alternative line–search rules or conditions to select a suitable step–size α_k along the ray $\{x_k + \alpha d_k : \alpha > 0\}$, for which it can be shown that the line–search method (2) converges to a stationary point. Namely, the Armijo rule, the strong Wolfe conditions, the Goldstein rule, for details see [28]. All these rules belong to the monotone approach, i.e., these

H. Oviedo
Escola de Matemática Aplicada, Fundação Getulio Vargas (FGV/EMAp). Rio de Janeiro, RJ, Brazil. E-mail: harry.leon@fgv.br

rules build a sequence of objective values $\{F(x_k)\}$ that verifies the inequality $F(x_{k+1}) < F(x_k)$, for all $k \in \mathbb{N}$. In the literature, there are also non–monotone strategies for regulating the step–size. In particular, the non–monotone approach promotes a decreasing tendency in $\{F(x_k)\}$ throughout the iterations, but does not necessarily verify the relation $F(x_{k+1}) < F(x_k)$ in each iteration. At present, the non–monotone rules are widely used, because they tend to reduce the number of line searches per iteration. Two of the most used non–monotone rules are the globalization strategy introduced by Grippo et. al. in [12] and the Zhang–Hager technique presented in [44]. In [44], the authors proved the global convergence of line–search methods, for unconstrained optimization problems (when $\mathcal{M} = \mathbb{R}^n$), under two very important direction assumptions.

Since 1972, Luenberger [27] promoted the idea of performing a line search along geodesics to address the problem of minimizing a cost function on a Riemannian manifold. However, early attempts to extend the line–search methods (2) to the context of optimization problems on Riemannian manifolds were introduced by Gabay [10], who introduced a Riemannian steepest descent, a Riemannian Newton method and a Riemannian quasi–Newton procedure, also stating their global and local convergence. Udriste [39] also analyzed the steepest descent on Riemannian manifolds and proved (linear) convergence of this approach under the assumption of exact step–size selection. In addition, in [42] Yang established the global convergence, and the rate of convergence of the Riemannian steepest descent and the Riemannian Newton's method by controlling the step–size with the Armijo's rule [28].

To generalize the linear scheme (2) to the Riemannian framework, all approaches listed above, substitute the linear step in (2), by a step along a geodesic. However, geodesics may be difficult to obtain, and even for some Riemannian manifold there is not always an explicit formula to compute it. In [9], Edelman et. al. introduced two Riemannian line–search methods based on geodesics to tackle minimization problems over the Stiefel manifold. In particular, in [9] is developed a closed formula for the geodesic and extending the ideas of the conjugate gradient method and the steepest descent method. Unfortunately, these two methods are computationally expensive because the geodesic evaluation involve a computation of the exponential matrix, which is very inefficient.

In alternative approaches, the geodesics are replaced by more general paths, based on retractions (a retraction $R_x(\cdot)$ is a functional from the tangent space $T_x\mathcal{M}$ to the manifold $\mathcal{M}$ at $x \in \mathcal{M}$, see Definition 1 in Section 2). Absil et al. in [2], extended the line–search methods for optimization on Riemannian manifolds considering retractions instead of geodesics, in order to preserve feasibility and at the same time, improve the efficiency of this class of methods. Additionally, in [2] was analyzed the global convergence of the general Riemannian line–search methods using the Armijo's rule to determined the step–size, and under the assumption that all the search directions are gradient–related. The asymptotic convergence analysis of some specific line–search methods in the Riemannian setting have been analyzed by many papers, for instance, the Newton method is analyzed in [4,16,35], Riemannian quasi–Newton methods [14,17,38], Riemannian conjugate gradient methods [22,32,35–37,43], several gradient–type methods are proposed and analyzed in [7,8,15,18,31,33]. All these papers listed above, present convergence analysis by choosing the step–size, in such a way that there is a monotone decrease in the objective function values, which is achieved using the Armijo rule, the strong Wolfe conditions, among others. Recently in [25], the global convergence of the Riemannian line–search procedures was analyzed under a modified version of the Armijo condition. However, only a few papers have studied the global convergence of some specific Riemannian line–search methods under non–monotone approach, see [11,15,18,29].

In this work, we extend the Zhang–Hager convergence analysis presented in [44], and demonstrate the global convergence of the general line–search methods in the Riemannian setting, under some directions assumptions. These directions conditions are the natural generalizations of the conditions for the searches directions introduced in [44]. Additionally, we introduce a new globally convergent Riemannian conjugate gradient method that satisfies the directions properties described in this work. Finally, we present two sets of numerical experiments to demonstrate the effectiveness and efficiency of our proposal.

The remainder of this paper is organized as follows. Section 2 presents some definitions and auxiliary notations related to Riemannian geometry that are important to our study. The general Riemannian line–search method with a modified Zhang–Hager conditions and the two directions assumptions are presented in Section 3. Section 4 is devoted to the asymptotic convergence analysis of the Riemannian line–search methods. Section 5 introduces a new globally convergent Riemannian conjugate gradient method that verifies our direction assumptions. Section 6 presents some numerical experiment to illustrate the effectiveness of our new proposed algorithm. Finally, the last section provides the conclusions of this work.

2 Elements of Riemannian manifolds

In this section, we briefly review some relevant concepts and notations of Riemannian geometry, by summarizing [2].

Let $\mathcal{M}$ be a finite–dimensional differentiable manifold, this concept appear in [2,15]. Let $x \in \mathcal{M}$ an arbitrary point on $\mathcal{M}$, a tangent vector ξ_x to $\mathcal{M}$ at x is a function such that there exists a curve $\gamma : \mathbb{R} \to \mathcal{M}$ satisfying that $\gamma(0) = x$ and

$$\xi_x f := \dot{\gamma}(0) f = \frac{\partial(f(\gamma(t)))}{\partial t}\Big|_{t=0}, \quad \forall f \in \Im_x(\mathcal{M}),$$

where $\Im_x(\mathcal{M})$ denotes the set of smooth real–valued functions defined on a neighborhood of $x \in \mathcal{M}$. The tangent space of $\mathcal{M}$ at x, denoted by $T_x\mathcal{M}$, is defined as the set of all tangent vectors to $\mathcal{M}$ at x. Now, if we endow every tangent space $T_x\mathcal{M}$ with a smooth varying inner product $\langle \cdot, \cdot \rangle_x$, then the pair $(\mathcal{M}, \langle \cdot, \cdot \rangle)$ becomes a Riemannian manifold. The smooth varying inner product is so–called the *Riemannian metric*, since this induces a metric on $T_x\mathcal{M}$. In fact, observe that the the local inner product $\langle \cdot, \cdot \rangle_x$ induces the norm $||\xi_x||_x := \sqrt{\langle \xi_x, \xi_x \rangle_x}$ on $T_x\mathcal{M}$. The tangent bundle $\mathcal{TM}$ of $\mathcal{M}$ is defined by $\mathcal{TM} := \cup_{x \in \mathcal{M}} T_x\mathcal{M}$. Let $f : \mathcal{M} \to \mathbb{R}$ be a function defined on a Riemannian $\mathcal{M}$, then the Riemannian gradient of f at x, denoted by $\nabla_{\mathcal{M}} f(x)$, is the unique vector in $T_x\mathcal{M}$ satisfying

$$Df(x)[\xi_x] = \langle \nabla_{\mathcal{M}} f(x), \xi_x \rangle_x, \quad \forall \xi_x \in T_x\mathcal{M},$$

where $Df(x)[\xi_x]$ is the derivative of $f(\gamma(t))$ at $t = 0$ and γ is any curve on the manifold that verifies $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi_x$.

In this paper, we study Riemannian line–search methods based on *retractions*, whose rigorous definition is presented below.

Definition 1 ([2]) A retraction on a manifold $\mathcal{M}$ is a smooth mapping $R(\cdot)$ from the tangent bundle $\mathcal{TM}$ onto $\mathcal{M}$ with the following properties. Let $R_x(\cdot)$ denote the restriction of $R(\cdot)$ to $T_x\mathcal{M}$.

1. $R_x(0_x) = x$, where 0_x denotes the zero element of $T_x\mathcal{M}$.
2. With the canonical identification, $T_{0_x} T_x\mathcal{M} \simeq T_x\mathcal{M}$, $R_x(\cdot)$ satisfies

$$\mathcal{D}R_x(0_X) = \mathrm{id}_{T_x\mathcal{M}},$$

where $\mathrm{id}_{T_x\mathcal{M}}$ denotes the identity mapping on $T_x\mathcal{M}$.

The second condition in Definition 1 is known as *local rigidity condition.*

Another fundamental concept that is necessary to define some line–search methods on manifolds is *vector transport.* A vector transport is a linear mapping that provides us a mechanism to move vectors from a tangent space to another. Now we present its mathematical definition.

Definition 2 ([2]) A vector transport $\mathcal{T}(\cdot)$ on a manifold $\mathcal{M}$ is a smooth mapping

$$\mathcal{T} : \mathcal{TM} \oplus \mathcal{TM} \to \mathcal{TM} : (\eta, \xi) \mapsto \mathcal{T}_\eta(\xi) \in \mathcal{TM},$$

satisfying the following properties for all $x \in \mathcal{M}$ where $\oplus$ denote the Whitney sum, that is,

$$\mathcal{TM} \oplus \mathcal{TM} = \{(\eta_x, \xi_x) : \eta_x, \xi_x \in T_x\mathcal{M}, x \in \mathcal{M}\}.$$

1. There exists a retraction $R(\cdot)$, called the retraction associated with $\mathcal{T}(\cdot)$, such that

$$\pi(\mathcal{T}_{\eta_x}(\xi_x)) = R_x(\eta_x), \quad \eta, \xi \in T_x\mathcal{M},$$

where $\pi(\mathcal{T}_{\eta_x}(\xi_x))$ denotes the foot of the tangent vector $\mathcal{T}_{\eta_x}(\xi_x)$.
2. $\mathcal{T}_{0_x}(\xi_x) = \xi_x$ for all $\xi_x \in T_x\mathcal{M}$.
3. $\mathcal{T}_{\eta_x}(a\xi_x + b\zeta_x) = a\mathcal{T}_{\eta_x}(\xi_x) + b\mathcal{T}_{\eta_x}(\zeta_x)$, for all $a, b \in \mathbb{R}$ and $\eta_x, \xi_x, \zeta_x \in T_x\mathcal{M}$.

Note that the first item of Definition 2 implies that $\mathcal{T}_{\eta_x}(\xi_x) \in T_{R_x(\eta_x)}\mathcal{M}$, for all $\eta_x, \xi_x \in T_x\mathcal{M}$.

A vector transport $\mathcal{T}(\cdot)$ on a manifold $\mathcal{M}$ is called *non–expansive* if it satisfies

$$\langle \mathcal{T}_{\eta_x}(\xi_x), \mathcal{T}_{\eta_x}(\xi_x)\rangle_{R_x(\eta_x)} \leq \langle \xi_x, \xi_x\rangle_x, \tag{3}$$

for all $\eta_x, \xi_x \in T_x\mathcal{M}$, where $R_x(\cdot)$ is the retraction associated with $\mathcal{T}(\cdot)$.

3 Riemannian line–search optimization methods

In this section, we briefly review the Riemannian line–search methods and present two general conditions that the search directions must satisfy, to guarantee the global convergence of this class of iterative process. In general, Riemannian line–search methods, in each iteration, compute the new iterate x_{k+1} from the previous point $x_k \in \mathcal{M}$, through the following recursion

$$x_{k+1} = R_{x_k}(\alpha_k z_k). \tag{4}$$

Notice that the update scheme (4) requires three ingredients: the step–size $\alpha_k > 0$, a tangent search direction $z_k \in T_{x_k}\mathcal{M}$, and a retraction mapping which ensures that the new iterate is a feasible point, i.e. $x_{k+1} \in \mathcal{M}$. In this way, the method iteratively constructs a feasible sequence of points $\{x_k\} \subset \mathcal{M}$ with the hope that it will converge to a local minimizer of problem (1), or at least to a stationary point.

On the other hand, the selection of an appropriate step–size has a large impact on the robustness and efficiency of the line search method. In particular, to assure a reduction in the value of the objective function, it is crucial to select a suitable step–size such that it is neither too large nor too small. In this paper, we combine the ideas of Zhang–Hager technique [44] with the Armijo–type condition [45]. Specifically, we select a step–size $\alpha_k > 0$ that verifies the following inequality

$$F(R_{x_k}(\alpha z_k)) \leq C_k + \rho_1\alpha\langle \nabla_{\mathcal{M}}F(x_k), z_k\rangle_{x_k} - \rho_2\alpha^2||z_k||^2_{x_k}, \tag{5}$$

where $\rho_1, \rho_2 \in (0, 1)$ and each reference value C_k is taken to be the convex combination of C_{k-1} and $F(x_k)$ defined by $C_k = (\eta Q_{k-1}C_{k-1} + F(x_k))/Q_k$, where $Q_k = \eta Q_{k-1} + 1$, $\eta \in [0, 1)$, $Q_0 = 1$ and $C_0 = F(x_0)$.

To guarantee the convergence of the line–search methods to stationary point, it is necessary that the search directions satisfy certain properties. In particular, in this work, we prove the global convergence of this class of methods, considering that the sequence of tangent directions $\{z_k\}$ satisfy the following properties:

1. Descent property:
$$\langle \nabla_{\mathcal{M}}F(x_k), z_k\rangle_{x_k} \leq -c_1||\nabla_{\mathcal{M}}F(x_k)||^2_{x_k}, \quad \forall k \in \mathbb{N}. \tag{6}$$
2. The search directions are bounded by the gradient norm:
$$||z_k||_{x_k} \leq c_2||\nabla_{\mathcal{M}}F(x_k)||_{x_k}, \quad \forall k \in \mathbb{N}. \tag{7}$$

where $c_1, c_2 > 0$ are two positive constants. When we have a sequence of directions $\{z_k\}$ verifying the inequalities (6)–(7), we will say that the *Direction Assumption* holds. Observe that these two properties directly implies that

$$\langle \nabla_{\mathcal{M}}F(x_k), z_k\rangle_{x_k} \leq -\frac{c_1}{c_2^2}||z_k||^2_{x_k}. \tag{8}$$

Remark 1 Note that the properties (6)–(7) are direct extensions of the conditions introduced in [44], where Zhang and Hager demonstrated the global convergence for line–search algorithms in the Euclidean setting.

The above description leads us to the following general Riemannian line–search algorithm (Algorithm 1).

Algorithm 1 Riemannian line–search method.

Require: $x_0 \in \mathcal{M}$, $z_0 \in T_{x_0}\mathcal{M}$ such that z_0 verifies the Direction assumptions (6)–(7), $0 < \alpha_m \leq \alpha_M < \infty$, $\eta \in [0, 1)$, $\rho_1, \rho_2, \delta \in (0, 1)$, $Q_0 = 1$, $C_0 = F(x_0)$, $k = 0$.

```
while ||∇_M F(x_k)||_{x_k} ≠ 0 do
    Select α_k ∈ [α_m, α_M].
    while F(R_{x_k}(α_k z_k)) > C_k + ρ_1 α_k ⟨∇_M F(x_k), z_k⟩_{x_k} − ρ_2 α_k^2 ||z_k||_{x_k}^2 do
        α_k ← δα_k,
    end while
    x_{k+1} = R_{x_k}(α_k z_k).
    Select a search direction z_{k+1} ∈ T_{x_{k+1}}M such that z_{k+1} verifies the inequalities (6) and (7).
    Q_{k+1} = ηQ_k + 1 and C_{k+1} = (ηQ_k C_k + F(x_{k+1}))/Q_{k+1}.
    k ← k + 1.
end while
```

4 Global convergence

In this section, we establish the global convergence of Algorithm 1. Firstly, observe that the pullback mapping $\hat{F} : \mathcal{TM} \to \mathbb{R}$ of the objective function through the retraction R on $\mathcal{M}$ is given by

$$\hat{F}(\xi) = F(R(\xi)), \quad \forall \xi \in T\mathcal{M}. \tag{9}$$

In addition, the restriction of $\hat{F}$ to the tangent space $T_x\mathcal{M}$ is denoted by $\hat{F}_x : T_x\mathcal{M} \to \mathbb{R}$, that is,

$$\hat{F}_x(\xi_x) = F(R_x(\xi_x)), \quad \forall \xi_x \in T_x\mathcal{M}. \tag{10}$$

Moreover, by differentiating the function $\hat{F}_x(\cdot)$ and using the second part of the Definition 1, we have

$$\nabla_{\mathcal{M}} F(x) = D\hat{F}_x(0_x), \tag{11}$$

where 0_x is the origin of $T_x\mathcal{M}$ and $D\hat{F}_x(\cdot)$ denotes the derivative of $\hat{F}_x(\cdot)$. To obtain the relation (11), it is necessary to use the canonical identification, $T_{0_x}T_x\mathcal{M} \simeq T_x\mathcal{M}$. In order to establish the global convergence of Algorithm 1, we need the following general assumption.

Assumption 1.

1. The objective function F is bounded below in the level set $\mathcal{L} = \{x \in \mathcal{M} : F(x) \leq F(x_0)\}$, which holds in particular when $\mathcal{M}$ itself is a compact manifold.
2. The derivative $D\hat{F}_x : T_x\mathcal{M} \to T_x\mathcal{M}$ is Lipchitz continuous at $0_x \in T_x\mathcal{M}$ uniformly in $x \in \mathcal{M}$ in the level set $\mathcal{L}$, that is, there exist two constants $\kappa > 0$ and $L > 0$ such that

$$||D\hat{F}_x(\xi_x) - D\hat{F}_x(0_x)||_x \leq L||\xi_x||_x, \tag{12}$$

for any $x \in \mathcal{L}$ and $\xi_x \in T_x\mathcal{M}$ verifying that $||\xi_x||_x \leq \kappa$.

Remark 2 The Assumption 1 was also considered to establish the global convergence of a Riemannian modified conjugate gradient method introduced in [43]. However, the analysis presented in this work differs from the theoretical results described in [43].

Lemma 1 *Let $\{x_k\}$ be an infinite sequence generated by Algorithm 1. Then*

$$F(x_k) \leq C_k, \quad \forall\, k \geq 0. \tag{13}$$

Proof Let us define $\psi : \mathbb{R} \to \mathbb{R}$ as

$$\psi(\tau) = \frac{\tau C_{k-1} + F(x_k)}{\tau + 1},$$

observe that the derivative of $\psi(\cdot)$ is

$$\dot{\psi}(\tau) = \frac{C_{k-1} - F(x_k)}{(\tau + 1)^2}.$$

By using the property (6) and the non–monotone Armijo–type condition (5), we obtain

$$F(x_k) \leq C_{k-1} + \rho_1 \alpha_{k-1} \langle \nabla_{\mathcal{M}} F(x_{k-1}), z_{k-1} \rangle_{x_{k-1}} - \rho_2 \alpha_{k-1}^2 ||z_{k-1}||_{x_{k-1}}^2 < C_{k-1}, \tag{14}$$

which implies that $\dot{\psi}(\tau) \geq 0$ for all $\tau \geq 0$. Hence, $\psi(\cdot)$ is a nondecreasing function, and $F(x_k) = \psi(0) \leq \psi(\tau)$ for all $\tau \geq 0$. Then, taking $\bar{\tau} = \eta Q_{k-1}$ we arrive at

$$F(x_k) = \psi(0) \leq \psi(\bar{\tau}) = C_k, \tag{15}$$

which completes the proof.

Lemma 2 *Let $\{x_k\}$ be an infinite sequence generated by Algorithm 1. Then the sequence $\{C_k\}$ is monotonically decreasing.*

Proof By the construction of Algorithm 1 and the property (6), we have

$$C_{k+1} = \frac{\eta Q_k C_k + F(x_{k+1})}{Q_{k+1}} < \frac{(\eta Q_k + 1) C_k}{Q_{k+1}} = C_k. \tag{16}$$

Therefore, $\{C_k\}$ is monotonically decreasing and converges to some limit $C^* \in \mathbb{R} \cup \{-\infty\}$.

The following lemma establishes a lower bound for the step–size α_k in Algorithm 1.

Lemma 3 *In Algorithm 1, for all k sufficiently large, there exists a constant $\nu > 0$ such that the step–size α_k satisfies*

$$\alpha_k \geq \nu. \tag{17}$$

Proof If $\alpha_k \geq \alpha_m$, we need no proof. Now, it follows form Lemma 1, Lemma 2 and Assumption 1 that $\{C_k\}$ is a convergent sequence. In addition, we have

$$\frac{-\rho_1 \alpha_k \langle \nabla_{\mathcal{M}} F(x_k), z_k \rangle_{x_k} + \rho_2 \alpha_k^2 ||z_k||_{x_k}^2}{Q_{k+1}} \leq C_k - C_{k+1}. \tag{18}$$

Merging this result with the fact that $Q_{k+1} = 1 + \eta Q_k = 1 + \eta + \eta^2 Q_{k-1} = \cdots = \sum_{i=0}^{k} \eta^i < (1-\eta)^{-1}$ and applying limits in both side of inequality (18), we arrive at

$$\lim_{k \to \infty} \alpha_k |\langle \nabla_{\mathcal{M}} F(x_k), z_k \rangle_{x_k}| = 0 \quad \text{and} \quad \lim_{k \to \infty} \alpha_k^2 ||z_k||_{x_k}^2 = 0. \tag{19}$$

Next, we assume that $\alpha_k < \alpha_m$ for all k sufficiently large. By the construction of the Algorithm 1, the Zhang–Hager–type inequality fails for $\delta^{-1}\alpha_k$, so for all k sufficiently large we have,

$$F(R_{x_k}(\delta^{-1}\alpha_k z_k)) - C_k > \rho_1 \delta^{-1} \alpha_k \langle \nabla_{\mathcal{M}} F(x_k), z_k \rangle_{x_k} - \rho_2 \delta^{-2} \alpha_k^2 ||z_k||_{x_k}^2, \tag{20}$$

which together with Lemma 1 implies that

$$F(R_{x_k}(\delta^{-1}\alpha_k z_k)) - F(x_k) > \rho_1 \delta^{-1} \alpha_k \langle \nabla_{\mathcal{M}} F(x_k), z_k \rangle_{x_k} - \rho_2 \delta^{-2} \alpha_k^2 ||z_k||_{x_k}^2. \tag{21}$$

From (19), we have, for all k sufficiently large,

$$\delta^{-1} \alpha_k ||z_k||_{x_k} \leq \kappa. \tag{22}$$

By using the mean–value theorem, the Cauchy–Schwarz inequality, (12), and (22), there exists a $t_k \in (0,1)$ such that, for all k sufficiently large,

$$\begin{aligned} F(R_{x_k}(\delta^{-1}\alpha_k z_k)) - F(x_k) &= F(R_{x_k}(\delta^{-1}\alpha_k z_k)) - F(R_{x_k}(0_{x_k})) \\ &= \hat{F}_{x_k}(\delta^{-1}\alpha_k z_k) - \hat{F}_{x_k}(0_{x_k}) \\ &= \delta^{-1}\alpha_k \langle D\hat{F}_{x_k}(t_k \delta^{-1} \alpha_k z_k), z_k \rangle_{x_k} \\ &= \delta^{-1}\alpha_k \langle D\hat{F}_{x_k}(0_{x_k}), z_k \rangle_{x_k} \\ &\quad + \delta^{-1}\alpha_k \langle D\hat{F}_{x_k}(t_k \delta^{-1}\alpha_k z_k) - D\hat{F}_{x_k}(0_{x_k}), z_k \rangle_{x_k} \\ &\leq \delta^{-1}\alpha_k \langle D\hat{F}_{x_k}(0_{x_k}), z_k \rangle_{x_k} + \delta^{-2}\alpha_k^2 t_k L ||z_k||_{x_k}^2 \\ &\leq \delta^{-1}\alpha_k \langle \nabla_{\mathcal{M}} F(x_k), z_k \rangle_{x_k} + \delta^{-2}\alpha_k^2 L ||z_k||_{x_k}^2. \end{aligned} \tag{23}$$

Combining (21) and (23), and using (8) yields for all k sufficiently large,

$$\begin{aligned}\alpha_k &\geq \frac{(\rho_1-1)\delta}{(L+\rho_2)}\frac{\langle\nabla_{\mathcal{M}}F(x_k),z_k\rangle_{x_k}}{||z_k||^2_{x_k}}\\ &= \frac{(1-\rho_1)\delta}{(L+\rho_2)}\frac{|\langle\nabla_{\mathcal{M}}F(x_k),z_k\rangle_{x_k}|}{||z_k||^2_{x_k}}\\ &\geq \frac{(1-\rho_1)\delta c_1}{(L+\rho_2)c_2^2}. \end{aligned}\tag{24}$$

Therefore, (17) holds for all k sufficiently large with $\nu = \min\left\{\alpha_m, \frac{(1-\rho_1)\delta c_1}{(L+\rho_2)c_2^2}\right\}$.

Now we prove the global convergence of Algorithm 1 to a stationary point of (1).

Theorem 1 *Suppose Assumption 1 is satisfied and Algorithm 1 generates an infinite sequence $\{x_k\}$ of points on the manifold. In addition, suppose that the sequence of the search direction $\{z_k\} \subset T\mathcal{M}$ verifies the properties* (6) *and* (7)*. Then we have*

$$\lim_{k\to\infty} ||\nabla_{\mathcal{M}}F(x_k)||_{x_k} = 0.$$

Proof By Lemma 3, the definition of C_{k+1}, and the descent property (6), we have, for all k sufficiently large, that

$$\begin{aligned}C_{k+1} &= \frac{\eta Q_k C_k + F(x_{k+1})}{Q_{k+1}}\\ &\leq \frac{\eta Q_k C_k + C_k + \rho_1\alpha_k\langle\nabla_{\mathcal{M}}F(x_k),z_k\rangle_{x_k} - \rho_2\alpha_k^2||z_k||^2_{x_k}}{Q_{k+1}}\\ &= C_k + \frac{\rho_1\alpha_k\langle\nabla_{\mathcal{M}}F(x_k),z_k\rangle_{x_k} - \rho_2\alpha_k^2||z_k||^2_{x_k}}{Q_{k+1}}\\ &\leq C_k + \frac{\rho_1\alpha_k\langle\nabla_{\mathcal{M}}F(x_k),z_k\rangle_{x_k}}{Q_{k+1}}\\ &\leq C_k - \frac{c_1\rho_1\alpha_k||\nabla_{\mathcal{M}}F(x_k)||^2_{x_k}}{Q_{k+1}}\\ &\leq C_k - \frac{c_1\rho_1\nu||\nabla_{\mathcal{M}}F(x_k)||^2_{x_k}}{Q_{k+1}}. \end{aligned}\tag{25}$$

Since $Q_{k+1} < (1-\eta)^{-1}$, form (25) we obtain

$$C_{k+1} < C_k - c_1\rho_1\nu(1-\eta)||\nabla_{\mathcal{M}}F(x_k)||^2_{x_k},$$

which implies that

$$0 \leq c_1\rho_1\nu(1-\eta)||\nabla_{\mathcal{M}}F(x_k)||^2_{x_k} < C_k - C_{k+1}. \tag{26}$$

Taking limits in both sides of (26) we arrive at

$$\lim_{k\to\infty} ||\nabla_{\mathcal{M}}F(x_k)||_{x_k} = 0,$$

which completes the proof.

5 A new Riemannian conjugate descent method

In this section, we introduce a new globally convergent Riemannian conjugate gradient algorithm to tackle the Riemannian optimization problem (1). Similar to the unconstrained optimization context, we have many possible choices to define the sequence of search directions $\{z_k\}$, that guarantee the properties (6)–(7), and therefore that generate a globally convergent method. For instance, the gradient descent, conjugate gradient, or quasi–Newton direction, can be adapted in order to verified the *Direction assumptions*. However, we focus on the conjugate gradient descent direction due to its simplicity. In particular, we consider the update scheme (4) and propose to build the $z_k's$ tangent search directions, recursively by

$$z_{k+1} = -\nabla_{\mathcal{M}} F(x_{k+1}) + \beta_k \mathcal{T}_{\alpha_k z_k}(z_k), \tag{27}$$

where $\mathcal{T}_x(\cdot)$ is any non–expansive vector transport and $\beta_k \in \mathbb{R}$ is given by

$$\beta_k = \mu_k \left(\frac{\langle \nabla_{\mathcal{M}} F(x_{k+1}), \mathcal{T}_{\alpha_k z_k}(z_k) \rangle_{x_{k+1}}}{-||z_k||^2_{x_k}} \right), \tag{28}$$

where $\{\mu_k\}$ is any bounded sequence of positive real numbers such that $0 < \mu_k \leq \mu_{\max}$, for all $k \in \mathbb{N}$, with $\mu_{\max} \in \mathbb{R}$.

The following lemma shows that the *Direction Assumption* holds, for the sequence of directions $\{z_k\}$ defined in (27)–(28).

Lemma 4 *The directions z_k's defined in* (27) *satisfies the directions properties* (6) *and* (7)*, for all $k \geq 0$.*

Proof Firstly, the equalities (27) and (28) yield

$$\begin{aligned} \langle \nabla_{\mathcal{M}} F(x_{k+1}), z_{k+1} \rangle_{x_{k+1}} &= -||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}} + \beta_k \langle \nabla_{\mathcal{M}} F(x_{k+1}), \mathcal{T}_{\alpha_k z_k}(z_k) \rangle_{x_{k+1}} \\ &= -||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}} - \frac{\beta_k^2}{\mu_k} ||z_k||^2_{x_k} \\ &\leq -||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}}. \end{aligned} \tag{29}$$

On the other hand, it follows from the Cahuchy–Schwarz inequality, the non–expansiveness of $\mathcal{T}_x(\cdot)$ and the definition of β_k that

$$\begin{aligned} ||z_{k+1}||^2_{x_{k+1}} &= ||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}} - 2\beta_k \langle \nabla_{\mathcal{M}} F(x_{k+1}), \mathcal{T}_{\alpha_k z_k}(z_k) \rangle_{x_{k+1}} + \beta_k^2 ||\mathcal{T}_{\alpha_k z_k}(z_k)||^2_{x_{k+1}} \\ &\leq ||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}} - 2\beta_k \langle \nabla_{\mathcal{M}} F(x_{k+1}), \mathcal{T}_{\alpha_k z_k}(z_k) \rangle_{x_{k+1}} + \beta_k^2 ||z_k||^2_{x_k} \\ &= ||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}} + \left(\frac{2+\mu_k}{\mu_k} \right) \beta_k^2 ||z_k||^2_{x_k} \\ &= ||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}} + (\mu_k^2 + 2\mu_k) \left(\frac{(\langle \nabla_{\mathcal{M}} F(x_{k+1}), \mathcal{T}_{\alpha_k z_k}(z_k) \rangle_{x_{k+1}})^2}{||z_k||^2_{x_k}} \right) \\ &\leq ||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}} + (\mu_k^2 + 2\mu_k) \left(\frac{||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}} ||\mathcal{T}_{\alpha_k z_k}(z_k)||^2_{x_{k+1}}}{||z_k||^2_{x_k}} \right) \\ &\leq ||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}} + (\mu_k^2 + 2\mu_k) \left(\frac{||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}} ||z_k||^2}{||z_k||^2_{x_k}} \right) \\ &= (\mu_k + 1)^2 \, ||\nabla_{\mathcal{M}} F(x_{k+1})||^2_{x_{k+1}}, \end{aligned}$$

In view of the inequality stated above and the fact that $\mu_k \leq \mu_{\max}$ for all $k \in \mathbb{N}$, we have

$$||z_{k+1}||_{x_{k+1}} \leq (\mu_{\max} + 1) ||\nabla_{\mathcal{M}} F(x_{k+1})||_{x_{k+1}},$$

which proves the lemma.

Now, we outline all the steps of our Riemannian conjugate descent method in Algorithm 2.

Algorithm 2 Riemannian conjugate descent method (RCD).

Require: $x_0 \in \mathcal{M}$, $\mu_{\max} > 0$, $\{\mu_k\}$ a sequence such that $0 < \mu_k \leq \mu_{\max}$ for all $k \in \mathbb{N}$, $0 < \alpha_m \leq \alpha_M < \infty$, $\eta \in [0, 1)$, $\rho_1, \rho_2, \epsilon, \delta \in (0, 1)$, $Q_0 = 1$, $C_0 = F(x_0)$, $k = 0$.

while $||\nabla_{\mathcal{M}} F(x_k)||_{x_k} > \epsilon$ **do**
  Select $\alpha_k \in [\alpha_m, \alpha_M]$.
  while $F(R_{x_k}(\alpha_k z_k)) > C_k + \rho_1 \alpha_k \langle \nabla_{\mathcal{M}} F(x_k), z_k \rangle_{x_k} - \rho_2 \alpha_k^2 ||z_k||_{x_k}^2$ **do**
    $\alpha_k \leftarrow \delta \alpha_k$,
  end while
  $x_{k+1} = R_{x_k}(\alpha_k z_k)$.
  $\beta_k = \left(\mu_k \langle \nabla_{\mathcal{M}} F(x_{k+1}), \mathcal{T}_{\alpha z_k}(z_k) \rangle_{x_{k+1}}\right) / (-||z_k||_{x_k}^2)$.
  $z_{k+1} = -\nabla_{\mathcal{M}} F(x_{k+1}) + \beta_k \mathcal{T}_{x_{k+1}}(z_k)$.
  $Q_{k+1} = \eta Q_k + 1$ and $C_{k+1} = (\eta Q_k C_k + F(x_{k+1}))/Q_{k+1}$.
  $k \leftarrow k + 1$.
end while

6 Numerical experiments

In this section, we present some numerical experiments in order to give further insight into the Riemannian conjugate descent method (RCD). We test our Algorithm 2 on two classes Riemannian optimization problems, namely, linear eigenvalue problem (LEP) and joint diagonalization problem (JDP). All experiments have been performed on a intel(R) CORE(TM) i7–4770, CPU 3.40 GHz with 500GB HD and 16GB RAM. The algorithm was coded in Matlab (version 2017b) with double precision. The running times are always given in CPU seconds. The implementation of our algorithm is available in `http://www.optimization-online.org/DB_HTML/2021/03/8297.html`.

In this section, we consider the following Riemannian manifolds together with their associated retractions and vector transports. Firstly, we consider *the Stiefel manifold*, which is defined by

$$St(n,p) = \{X \in \mathbb{R}^{n \times p} : X^\top X = I\}.$$

For our numerical experiments concerning the Stiefel manifold we will use the following retractions, introduced in [2], the retraction based on the QR decomposition, that is,

$$R_X(\xi_X) = \mathtt{qf}(X + \xi_X), \quad \forall \xi_X \in T_X St(n,p), \tag{30}$$

where $X \in St(n,p)$ and $\mathtt{qf}(\mathrm{M})$ denotes the orthogonal factor Q obtained form the QR decomposition of M, such that $M = QR$; where $Q \in St(n,p)$ and $R \in \mathbb{R}^{p \times p}$ is an upper triangular matrix with strictly positive diagonal elements. Additionally, we consider the retraction based on the polar decomposition

$$R_X(\xi_X) = (X + \xi)\left((X + \xi_X)^\top (X + \xi_X)\right)^{-1/2}, \quad \forall \xi_X \in T_X St(n,p). \tag{31}$$

In addition, for the Stiefel manifold, we will use the vector transport based on the orthogonal projection mapping over the tangent space of $St(n,p)$, i.e.

$$\mathcal{T}_{\eta_X}(\xi_X) = \xi_X - \frac{1}{2} R_X(\eta_X)\left(R_X(\eta_X)^\top \xi_X + \xi_X^\top R_X(\eta_X)\right),$$

where $\eta_X, \xi_X \in T_X St(n,p)$, $X \in St(n,p)$ and $R_X(\cdot)$ is any of the two retractions presented in (30)–(31).

On the other hand, we also consider *the oblique manifold*,

$$\mathcal{OB}(n,p) = \{X \in \mathbb{R}^{n \times p} : \mathtt{ddiag}(X^\top X) = I\},$$

where $\mathtt{ddiag}(M)$ denotes the matrix M with all its off–diagonal entries assigned to zero. For this specific manifold, we will use the retraction presented in [1], which is given by

$$R_X(\xi_X) = (X + \xi)\mathtt{ddiag}\left((X + \xi_X)^\top (X + \xi_X)\right)^{-1/2}, \quad \forall \xi_X \in T_X St(n,p). \tag{32}$$

Additionally, we consider the following vector transport associated to the oblique manifold,

$$\mathcal{T}_{\eta_X}(\xi_X) = \xi_X - \frac{1}{2}R_X(\eta_X)\texttt{ddiag}\left(R_X(\eta_X)^\top \xi_X\right),$$

where $\eta_X, \xi_X \in T_X\mathcal{OB}(n,p)$, and $X \in \mathcal{OB}(n,p)$.

6.1 Stopping criteria and implementation details

We let all the algorithms run up to K iterations, and stop them at iteration $k < K$ as soon as $||\nabla_{\mathcal{M}}F(X_k)||_F < \epsilon$, where $||M||_F$ denotes the Frobenius norm of the matrix M and $\nabla_{\mathcal{M}}F(\cdot)$ is the Riemannian gradient of F obtained under the standard inner product $\langle A, B\rangle = tr[A^\top B]$, where $tr(M)$ denotes the trace of the matrix M. Also, in case

$$\frac{||X_{k+1} - X_k||_F}{||X_k||_F} < \epsilon_X \quad \text{and} \quad \frac{|F(X_{k+1}) - F(X_k)|}{|F(X_k)| + 1} < \epsilon_F,$$

the iterations are interrupted, as well as when the maximum number of iterations ($K = 2000$) is allowed for the algorithms. The defaults values of ϵ; ϵ_X; ϵ_F are 1e-5, 1e-18, 1e-18, respectively. In addition, in Algorithm 2 we use $\delta = 0.2$, $\mu_{\max} = 1e10$, $\alpha_m = 1e\text{-}10$, $\alpha_M = 1e10$, $\eta = 0.85$, $\rho_1 = \rho_2 = 1e\text{-}4$, and $\mu_k = \alpha_k$ as defaults values.

In the rest of this section, we will use the following notation: *Time*, *Iter*, *Grad* and *Fval* will denote the averaged total computing time in seconds, the averaged number of iterations, the averaged residual $||\nabla_{\mathcal{M}}F(\hat{X})||_F$ where $\hat{X}$ is the estimated optimum by the method, respectively. In all experiment presented below, we solve ten independent instances for each pair (n, p) and then we report all these averages.

6.2 The linear eigenvalue problem

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with associate eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$. The linear eigenvalue problem is mathematically formulated as

$$\max_{X\in\mathbb{R}^{n\times p}} tr[X^\top AX] \quad \text{s.t.} \quad X^\top X = I. \tag{33}$$

It is well–known that any global solution of the optimization problem (33) correspond to a matrix $X^* \in \mathbb{R}^{n\times p}$, whose columns span the eigenspace of A associated with the p–largest eigenvalues of A, and the optimal value is $F(X^*) = tr[(X^*)^\top AX^*] = \sum_{i=1}^{p} \lambda_i$.

In this subsection, we compare the numerical behavior of the Algorithm 2 with the algorithm proposed by Wen and Yin [40][1], denotes by *OptSt*, and with the Riemannian conjugate gradient methods *Algor.1a* and *Algor.1b* developed in [46][2], on two sets of symmetric and positive definite matrices. Both test sets includes a few randomly generated dense matrices assembled as $A = M^\top M$, where $M \in \mathbb{R}^{n\times n}$ is a matrix whose elements are sampled from the standard Gaussian distribution. The first set correspond to n varying from 500 through 5000 and a fixed $p = 6$; while the second set involves randomly generated problems with a fixed $n = 1000$ and varying $p \in \{1, 5, 10, 50, 100, 200\}$. The numerical results associated with these two experiments are presented in Table 1 and Table 2, respectively.

From Table 1 we observe that our RCD algorithm (with any of the two retractions) converges to a stationary point of problem (33) in fewer iterations than the rest of the methods. Moreover, we notice that the RCD with polar retraction and the RCD with QR retraction solvers show almost the same performance. Additionally, we see that when n grows, our method converges faster than all other methods in terms of CPU–time and number of iterations. A similar behavior can be observed in Table 2. In fact in this last table, we clearly note that the two versions of our Riemannian conjugate gradient method are superior are to the rest of the procedures when p approaches n.

[1] The OptSt Matlab code is available in `https://github.com/wenstone/OptM`

[2] The Riemannian conjugate gradient methods Algor.1a and Algor.1b can be downloaded from `http://www.optimization-online.org/DB_HTML/2016/09/5617.html`

Table 1 Numerical results related to the linear eigenvalue problem on randomly generated dense matrices for fixed $p = 6$.

n	500	1000	2000	3000	4000	5000
			OptSt			
Iter	128	124	307	297	369	440
Time	0.033	0.081	1.176	2.497	5.181	11.301
Grad	6.36e-6	7.81e-6	8.16e-6	8.01e-6	5.48e-6	7.42e-6
Fval	1.14e+4	2.32e+4	4.70e+4	7.08e+4	9.48e+4	1.19e+5
			Algor.1a (CG)			
Iter	135	140	336	339	443	436
Time	0.051	0.161	1.562	3.565	8.259	12.309
Grad	8.21e-6	8.14e-6	8.06e-6	8.39e-6	8.47e-6	8.66e-6
Fval	1.14e+4	2.32e+4	4.70e+4	7.08e+4	9.48e+4	1.19e+5
			Algor.1b (CG)			
Iter	132	139	337	363	450	446
Time	0.046	0.159	1.555	3.966	8.298	12.866
Grad	7.83e-6	9.10e-6	9.11e-6	8.18e-6	7.71e-6	7.38e-6
Fval	1.14e+4	2.32e+4	4.70e+4	7.08e+4	9.48e+4	1.19e+5
			RCD with polar retraction			
Iter	127	123	288	257	373	370
Time	0.031	0.078	0.809	1.566	4.463	6.804
Grad	7.92e-6	6.37e-6	8.79e-6	7.40e-6	7.68e-6	7.40e-6
Fval	1.14e+4	2.32e+4	4.70e+4	7.08e+e4	9.48e+4	1.19e+e5
			RCD with QR retraction			
Iter	121	124	278	275	348	351
Time	0.028	0.074	0.771	1.792	3.816	6.477
Grad	7.67e-6	6.44e-6	8.60e-6	7.33e-6	8.66e-6	8.19e-6
Fval	1.14e+4	2.32e+4	4.70e+4	7.08e+4	9.48e+4	1.19e+5

Table 2 Numerical results associated with the linear eigenvalue problem of randomly generated 1000–dimensional dense matrices and varying p.

p	1	5	10	50	100	200
			OptSt			
Iter	107	153	241	297	460	519
Time	0.030	0.108	0.349	1.214	4.632	15.651
Grad	7.10e-6	6.67e-6	7.98e-6	7.36e-6	6.18e-6	7.48e-6
Fval	3.97e+3	1.94e+4	3.82e+4	1.72e+5	3.13e+5	5.36e+5
			Algor.1a (CG)			
Iter	130	177	231	307	464	593
Time	0.059	0.236	0.447	1.695	6.501	27.192
Grad	7.88e-6	8.69e-6	7.92e-6	7.92e-6	8.16e-6	8.42e-6
Fval	3.97e+3	1.94e+4	3.82e+4	1.72e+5	3.13e+5	5.36e+5
			Algor.1b (CG)			
Iter	130	172	243	326	483	558
Time	0.068	0.223	0.461	1.889	6.294	23.729
Grad	8.02e-6	8.32e-6	8.42e-6	9.08e-6	9.26e-6	8.83e-6
Fval	3.97e+3	1.94e+4	3.82e+4	1.72e+5	3.13e+5	5.36e+5
			RCD with polar retraction			
Iter	118	150	219	268	378	482
Time	0.040	0.109	0.261	1.011	2.098	8.965
Grad	8.19e-6	8.35e-6	7.30e-6	7.57e-6	7.60e-6	6.80e-6
Fval	3.97e+3	1.94e+4	3.82e+4	1.72e+5	3.13e+5	5.36e+5
			RCD with QR retraction			
Iter	119	146	228	263	380	463
Time	0.036	0.104	0.252	0.707	1.966	7.916
Grad	7.98e-6	6.38e-6	7.79e-6	5.92e-6	7.43e-6	8.12e-6
Fval	3.97e+3	1.94e+e4	3.82e+4	1.72e+5	3.13e+5	5.36e+5

Table 3 Numerical results on the joint diagonalization problem.

Method	Iter	Time	Grad	Iter	Time	Grad
	$n = 500$, $p = 100$, $N = 5$			$n = 1000$, $p = 10$, $N = 5$		
RCG	136	4.394	6.78e-7	40	2.441	1.51e-7
RBFGS	128	5.700	9.48e-7	29	1.844	1.17e-7
Trust–Reg	18	10.592	1.78e-8	9	1.392	2.47e-9
RCD	158	2.350	8.83e-7	29	0.763	4.78e-7
	$n = 5000$, $p = 10$, $N = 5$			$n = 5000$, $p = 10$, $N = 5$		
RCG	67	6.341	2.18e-7	37	48.350	6.63e-7
RBFGS	64	6.504	9.71e-7	25	34.678	2.12e-7
Trust–Reg	15	15.008	2.77e-7	8	20.822	2.27e-7
RCD	90	4.133	3.52e-7	21	13.066	3.57e-7

6.3 The joint diagonalization problem

The joint diagonalization problem was introduced by Absil and Gallivan in [1], where they considered the following oblique manifold constrained optimization problem

$$\min_{X\in\mathbb{R}^{n\times p}} \sum_{i=1}^{N} ||\texttt{off}(X^\top A_i X)||_F^2 \quad \text{s.t.} \quad \texttt{ddiag}(X^\top X) = I, \tag{34}$$

where the matrix function $\texttt{off}(\cdot)$ is given by $\texttt{off}(M) := M - \texttt{ddiag}(M)$, for every square matrix $M \in \mathbb{R}^{m\times m}$. This kind of optimization model can be used to perform independent component analysis, see [1]. In this subsection we consider the optimization problem (34), in order to evaluate the numerical performance of our Algorithm 2.

For benchmarking, we compare our RCD method (Algorithm 2) with the *manopt* toolbox [6][3]. In fact, the *manopt* a Matlab toolbox to conduct optimization on several available Riemannian manifolds, including the oblique manifold, and it contains many iterative solvers based on Riemannian trust regions (Trust–Reg), Riemannian steepest descent (SD), Riemannian Barzilai–Borwein method (RBB), Riemannian conjugate gradient (RCG), Riemannian quasi–Newton methods (RBFGS), among others. For the comparisons presented in this subsection, we choose the following solvers: Trust–Reg, RCG and RBFGS because these procedures tend to converge in fewer iterations than the SD and RBB methods.

In all the experiments the data matrices A_i were generated by

$$A_i = \texttt{diag}(\sqrt{n+1}, \sqrt{n+2}, \cdots, \sqrt{2n}) + B_i^\top + B_i, \quad \forall i \in \{1, 2, \cdots, N\},$$

where the B_i matrices were randomly generated by the Matlab command $B_i = \texttt{randn}(n)$. This specific experiment design was taken from [46]. The numerical results concerning this experiment are reported in Table 3. From Table 3, we can see that our proposal is more efficient than the three methods contained in the manopt toolbox, because our RCD approach converges to a stationary point of (34) in less computational time than the manopt methods, in all the tested instances.

7 Concluding remarks

In this work, we have established the global convergence of the Riemannian line–search methods based on retractions mappings. Specifically, we demonstrated that if we adopt a modified version of the non–monotone line–search technique of Zhang and Hager [44] as the globalization tool to determine the step–size, in the general update scheme of the Riemannian line–search approaches, then the sequence $\{||\nabla_{\mathcal{M}} F(x_k)||_{x_k}\}$ tends to zero, under two practical assumptions related to the tangent search directions. This new theoretical analysis can be seen as a generalization of the convergence study developed by Zhang and Hager in [44], where these authors focused on analyzing the convergence of this type of methods in the Euclidean setting.

[3] The manopt tool–box is available in `http://www.manopt.org/`

In addition, we have proposed a new globally convergent Riemannian conjugate gradient algorithm (RCD) to deal with optimization problems on Riemannian manifolds. The search directions computed by our new algorithm were proved to satisfy the two direction assumptions introduced in this manuscript, so that the global convergence result presented in this work apply. Several numerical experiments on set covering linear eigenvalue problems defined on the Stiefel manifold and joint diagonalization problems constrained to the oblique manifold was performed to illustrate the effectiveness and efficiency of our novel procedure. Our numerical results showed that our RCD reaches better results than some Riemannian iterative solvers existing in the literature.

Acknowledgements

The author was financially supported by FGV (Fundação Getulio Vargas) through the excellence post–doctoral fellowship program.

References

1. P-A Absil and Kyle A Gallivan. Joint diagonalization on the oblique manifold for independent component analysis. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V. IEEE, 2006.
2. P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
3. Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128. PMLR, 2016.
4. Marcio Antonio de Andrade Bortoloti, Teles Araujo Fernandes, and Orizon Pereira Ferreira. On the globalization of riemannian newton method. *arXiv preprint arXiv:2008.06557*, 2020.
5. Nicolas Boumal and Pierre-antoine Absil. Rtrmc: A riemannian trust-region method for low-rank matrix completion. *Advances in neural information processing systems*, 24:406–414, 2011.
6. Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.
7. Oscar Dalmau and Harry Oviedo. A projection method for optimization problems on the stiefel manifold. In *Mexican Conference on Pattern Recognition*, pages 84–93. Springer, 2017.
8. Oscar Dalmau and Harry Oviedo. Projected nonmonotone search methods for optimization with orthogonality constraints. *Computational and Applied Mathematics*, 37(3):3118–3144, 2018.
9. Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.
10. Daniel Gabay. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219, 1982.
11. Bin Gao, Nguyen Thanh Son, P-A Absil, and Tatjana Stykel. Riemannian optimization on the symplectic stiefel manifold. *arXiv preprint arXiv:2006.15226*, 2020.
12. Luigi Grippo, Francesco Lampariello, and Stephano Lucidi. A nonmonotone line search technique for newton's method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.
13. Igor Grubišić and Raoul Pietersz. Efficient rank reduction of correlation matrices. *Linear algebra and its applications*, 422(2-3):629–653, 2007.
14. Jiang Hu, Bo Jiang, Lin Lin, Zaiwen Wen, and Ya-xiang Yuan. Structured quasi-newton methods for optimization with orthogonality constraints. *SIAM Journal on Scientific Computing*, 41(4):A2239–A2269, 2019.
15. Jiang Hu, Xin Liu, Zai-Wen Wen, and Ya-Xiang Yuan. A brief introduction to manifold optimization. *Journal of the Operations Research Society of China*, 8(2):199–248, 2020.
16. Jiang Hu, Andre Milzarek, Zaiwen Wen, and Yaxiang Yuan. Adaptive quadratically regularized newton method for riemannian optimization. *SIAM Journal on Matrix Analysis and Applications*, 39(3):1181–1207, 2018.
17. Wen Huang, Kyle A Gallivan, and P-A Absil. A broyden class of quasi-newton methods for riemannian optimization. *SIAM Journal on Optimization*, 25(3):1660–1685, 2015.
18. Bruno Iannazzo and Margherita Porcelli. The riemannian barzilai–borwein method with nonmonotone line search and the matrix geometric mean computation. *IMA Journal of Numerical Analysis*, 38(1):495–517, 2018.
19. Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11(2), 2010.
20. Effrosini Kokiopoulou, Jie Chen, and Yousef Saad. Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications*, 18(3):565–602, 2011.
21. Rongjie Lai and Stanley Osher. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58(2):431–449, 2014.
22. Hugo Lara and Harry Oviedo. Solving joint diagonalization problems via a riemannian conjugate gradient method in stiefel manifold. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 6(2), 2018.
23. Hugo Lara, Harry Oviedo, and Jinyun Yuan. Matrix completion via a low rank factorization model and an augmented lagrangean succesive overrelaxation algorithm. *Bulletin of Computational Applied Mathematics*, 2(2), 2014.

24. Jun Li, Li Fuxin, and Sinisa Todorovic. Efficient riemannian optimization on the stiefel manifold via the cayley transform. *arXiv preprint arXiv:2002.01113*, 2020.
25. Xiao-bo Li, Nan-jing Huang, Qamrul Hasan Ansari, and Jen-Chih Yao. Convergence rate of descent method with new inexact line-search on riemannian manifolds. *Journal of Optimization Theory and Applications*, 180(3):830–854, 2019.
26. Zhizhong Li, Deli Zhao, Zhouchen Lin, and Edward Y Chang. A new retraction for accelerating the riemannian three-factor low-rank matrix completion algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4530–4538, 2015.
27. David G Luenberger. The gradient projection method along geodesics. *Management Science*, 18(11):620–631, 1972.
28. Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
29. Harry Oviedo. Implicit steepest descent algorithm for optimization with orthogonality constraints.
30. Harry Oviedo and Oscar Dalmau. A scaled gradient projection method for minimization over the stiefel manifold. In *Mexican International Conference on Artificial Intelligence*, pages 239–250. Springer, 2019.
31. Harry Oviedo, Oscar Dalmau, and Hugo Lara. Two adaptive scaled gradient projection methods for stiefel manifold constrained optimization. *Numerical Algorithms*, 87:1107–1127, 2020.
32. Harry Oviedo and Hugo Lara. A riemannian conjugate gradient algorithm with implicit vector transport for optimization in the stiefel manifold. Technical report, Technical report, Technical report. UFSC-Blumenau, CIMAT, 2018.
33. Harry Oviedo, Hugo Lara, and Oscar Dalmau. A non-monotone linear search algorithm with mixed direction on stiefel manifold. *Optimization Methods and Software*, 34(2):437–457, 2019.
34. Houduo Qi and Defeng Sun. A quadratically convergent newton method for computing the nearest correlation matrix. *SIAM journal on matrix analysis and applications*, 28(2):360–385, 2006.
35. Wolfgang Ring and Benedikt Wirth. Optimization methods on riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627, 2012.
36. Hiroyuki Sakai and Hideaki Iiduka. Hybrid riemannian conjugate gradient methods with global convergence properties. *Computational Optimization and Applications*, 77(3):811–830, 2020.
37. Hiroyuki Sato and Toshihiro Iwai. A new, globally convergent riemannian conjugate gradient method. *Optimization*, 64(4):1011–1031, 2015.
38. Matthias Seibert, Martin Kleinsteuber, and Knut Hüper. Properties of the bfgs method on riemannian manifolds. *Mathematical System Theory C Festschrift in Honor of Uwe Helmke on the Occasion of his Sixtieth Birthday*, pages 395–412, 2013.
39. Constantin Udriste. *Convex functions and optimization methods on Riemannian manifolds*, volume 297. Springer Science & Business Media, 2013.
40. Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1):397–434, 2013.
41. Chao Yang, Juan C Meza, Byounghak Lee, and Lin-Wang Wang. Kssolv—a matlab toolbox for solving the kohn-sham equations. *ACM Transactions on Mathematical Software (TOMS)*, 36(2):1–35, 2009.
42. Yaguang Yang. Globally convergent optimization algorithms on riemannian manifolds: Uniform framework for unconstrained and constrained optimization. *Journal of Optimization Theory and Applications*, 132(2):245–265, 2007.
43. Teng-Teng Yao, Zheng-Jian Bai, and Zhi Zhao. A riemannian variant of the fletcher–reeves conjugate gradient method for stochastic inverse eigenvalue problems with partial eigendata. *Numerical Linear Algebra with Applications*, 26(2):e2221, 2019.
44. Hongchao Zhang and William W Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM journal on Optimization*, 14(4):1043–1056, 2004.
45. Li Zhang, Weijun Zhou, and Donghui Li. Global convergence of a modified fletcher–reeves conjugate gradient method with armijo-type line search. *Numerische mathematik*, 104(4):561–572, 2006.
46. Xiaojing Zhu. A riemannian conjugate gradient method for optimization on the stiefel manifold. *Computational optimization and Applications*, 67(1):73–110, 2017.