SPECTRAL RESIDUAL METHOD FOR NONLINEAR EQUATIONS ON RIEMANNIAN MANIFOLDS

HARRY OVIEDO* AND HUGO LARA†

Abstract. In this paper, the spectral algorithm for nonlinear equations (SANE) is adapted to the problem of finding a zero of a given tangent vector field on a Riemannian manifold. The generalized version of SANE uses, in a systematic way, the tangent vector field as a search direction and a continuous real–valued function that adapts this direction and ensures that it verifies a descent condition for an associated merit function. In order to speed up the convergence of the proposed method, we incorporate a Riemannian adaptive spectral parameter in combination with a non–monotone globalization technique. The global convergence of the proposed procedure is established under some standard assumptions. Numerical results indicate that our algorithm is very effective and efficient solving tangent vector field on different Riemannian manifolds and competes favorably with a Polak–Ribiére–Polyak Method recently published and other methods existing in the literature.

Key words. Tangent vector field, Riemannian manifold, Nonlinear system of equations, Spectral residual method, non–monotone line search.

AMS subject classifications. 65K05, 90C30, 90C56, 53C21.

1. Introduction.

In this work, we consider the problem of finding a zero of a tangent vector field $F(\cdot)$ over a Riemannian manifold $\mathcal{M}$, with an associated Riemannian metric $\langle\cdot,\cdot\rangle$ (a local inner product which induces a corresponding local metric). The problem can be mathematically formulated as the solution of the following nonlinear equation

$$F(X) = 0_X, \tag{1.1}$$

where $F : \mathcal{M} \to \mathcal{TM}$ is a continuously differentiable tangent vector field, and $\mathcal{TM} := \cup_{X\in\mathcal{M}} T_X\mathcal{M}$ denotes the tangent bundle of $\mathcal{M}$, i.e., $\mathcal{TM}$ is the union of all tangent spaces at points in the manifold. Here, 0_X denotes the zero vector of the tangent space $T_X\mathcal{M}$. This kind of problem appears frequently in several applications, for example: statistical principal component analysis [17], where the Oja's flow induces the associated vector field, total energy minimization in electronic structure calculations [16, 25, 33], linear eigenvalue problems [15, 29], dimension reduction techniques in pattern recognition [13, 28, 32], Riemannian optimization problems, where the Riemannian gradient flow leads to the associated tangent vector field [2, 10, 24], among others.

Problem (1.1) is closely related to the problem of minimizing a differentiable function over the manifold $\mathcal{M}$,

$$\min \mathcal{F}(X) \quad \text{s.t.} \quad X \in \mathcal{M}, \tag{1.2}$$

where $\mathcal{F} : \mathcal{M} \to \mathbb{R}$ is a smooth function. Different iterative methods have been developed for solving (1.2). Some popular schemes are based on gradient method [6, 12, 15, 18, 19, 20, 21], conjugate gradient methods [2, 10, 34], Newton's method [2, 26], or quasi–Newton methods [2]. All these numerical methods can be used to

*Mathematics Research Center, CIMAT A.C. Guanajuato, Mexico. (harry.oviedo@cimat.mx)

†Universidade Federal de Santa Catarina, Campus Blumenau, Blumenau, Brazil. (hugo.lara.urdaneta@ufsc.br)

find a zero of the following tangent vector field equation,

$$\nabla_{\mathcal{M}}\mathcal{F}(X) = 0_X, \tag{1.3}$$

where $\nabla_{\mathcal{M}}\mathcal{F}(\cdot)$ denotes the Riemannian gradient of $\mathcal{F}(\cdot)$, which is a particular case of problem (1.1). The Riemannian line–search methods, designed to solve the optimization problem (1.2) construct a sequence of points using the following recursive formula

$$X_{k+1} = R_{X_k}[\xi_{X_k}], \tag{1.4}$$

where $R_X : T_X\mathcal{M} \to \mathcal{M}$ is a retraction (see Definition 2.1), and $\xi_{X_k} \in T_{X_k}\mathcal{M}$ is a descent direction, i.e, ξ_{X_k} verifies the inequality $\langle \nabla_{\mathcal{M}}\mathcal{F}(X_k), \xi_{X_k}\rangle < 0$ for all $k \geq 0$. Among the Riemannian line–search methods, the Riemannian gradient approach exhibits the lowest cost per iteration. This method uses the gradient vector field $\nabla_{\mathcal{M}}\mathcal{F}(\cdot)$ to define the search direction by $\xi_{X_k} = -\nabla_{\mathcal{M}}\mathcal{F}(X_k)$, at each iteration.

In the literature, there are some iterative algorithms addressing the problem (1.1). In [3, 9, 14] were developed several Riemannian Newton methods for the solution of tangent vector fields on general Riemannian manifolds. Among the features of Newton's method, the requirement of using second–order information and geodesics (that involves the computation of exponential mapping) to ensure keeping into the corresponding manifold, leads to the growth of computational cost. In addition, the authors in [5] proposed a Riemannian Gauss–Newton method to address the solution of (1.1) through the optimization problem (1.2). Recently, in [30] was introduced a Riemannian conjugate gradient to deal with the numerical solution of (1.1), that does not need derivative computation (it does not use the Jacobian of $F(\cdot)$), and that incorporates the use of retractions (see Definition 2.1), which is a mapping that generalizes the definition of geodesics, and that was introduced by Absil in [2] to deal with optimization problems on matrix manifolds.

In the Euclidean case $\mathcal{M} = \mathbb{R}^n$ in (1.1), i.e., for the solution of standard nonlinear system of equations, the authors in [7] introduced a method called SANE, which uses the residual $\pm F(X_k)$ as a search direction. Then the trial point, at each iteration, is computed by $X_{k+1} = X_k \pm \tau_k F(X_k)$, where τ_k is a spectral coefficient based on the Barzilai–Borwein step–size [4, 22]. This iterative process uses precisely the functional $F(\cdot)$, in order to define the search direction. It's feature of been a derivative–free procedure, is highly attractive, lowering the storage requirements and the computational cost per iteration.

Motivated by the Riemannian gradient and SANE methods, in this paper, we introduce RSANE, which is a generalization of SANE to tackle the numerical solution of nonlinear equations on Riemannian manifolds. In particular, we modified the update formula of SANE by incorporating a retraction, in order to guarantee that each point X_k belongs to the desired manifold. By following the ideas of the Riemannian Barzilai–Borwein method developed by Iannazzo et.al in [12], we propose an extension of the spectral parameter τ_k to the case of Riemannian manifolds, using mappings so–called *scaled vector transport*. In addition, we present the convergence analysis of the proposed method obtained under the Zhang–Hager globalization strategy [31]. Finally, some numerical experiments are reported to illustrate the efficiency and effectiveness of our proposal.

The rest of this paper is organized as follow. To do this article self–contained, we briefly review, in Section 2, some concepts and tools from Riemannian geometry that it can be founded in [2]. In Section 3, we present our proposed Riemannian spectral residual method (RSANE) for solving (1.1). Section 4 is devoted to the convergence analysis concerning our proposed method. In Section 5 numerical tests are carried out, in order to illustrate the good performance of our approach considering the computing of eigenspaces associated to given symmetric matrices using both simulated data and real data. Finally, conclusions and perspectives are provided in Section 7.

2. Preliminaries on Riemannian Geometry. In this section, we briefly review some notions and tools of Riemannian geometry crucial for understanding this paper, by summarizing [2].

Let $\mathcal{M}$ be a Riemannian manifold with an associated Riemannian metric $\langle\cdot,\cdot\rangle$, and let $T_X\mathcal{M}$ be its tangent vector space at a given point $X \in \mathcal{M}$. In addition, let $\mathcal{F} : \mathcal{M} \to \mathbb{R}$ a smooth scalar function defined on the Riemannian manifold $\mathcal{M}$, the Riemannian gradient of $\mathcal{F}(\cdot)$ at X, denoted by $\nabla_{\mathcal{M}}\mathcal{F}(X)$, is defined as the unique element of $T_X\mathcal{M}$ that verifies

$$\langle \nabla_{\mathcal{M}}\mathcal{F}(X), \xi_X\rangle = \mathcal{D}\mathcal{F}(X)[\xi_X], \quad \forall \xi_X \in T_X\mathcal{M},$$

where $\mathcal{D}\mathcal{F}(X)[\xi_X]$ is the function that takes any point $X \in \mathcal{M}$ to the directional derivative of $\mathcal{F}(\cdot)$ in the direction ξ_X, evaluated at $X \in \mathcal{M}$. In the particular case that $\mathcal{M}$ is a Riemannian submanifold of an Euclidean space $\mathcal{E}$, we have an explicit evaluation of the gradient: let $\bar{\mathcal{F}} : \mathcal{E} \to \mathbb{R}$ a smooth function defined on $\mathcal{E}$ and let $\mathcal{F} : \mathcal{M} \subset \mathcal{E} \to \mathbb{R}$, then the Riemannian gradient of $\mathcal{F}(\cdot)$ evaluated at $X \in \mathcal{M}$ is equal to the orthogonal projection of the standard gradient of $\bar{\mathcal{F}}(\cdot)$ onto $T_X\mathcal{M}$, that is,

$$\nabla_{\mathcal{M}}\mathcal{F}(X) = P_X[\nabla\bar{\mathcal{F}}(X)]. \tag{2.1}$$

This result provides us an important tool to compute the Riemannian gradient, which will be useful in the experiments section.

Another fundamental concept for this work is *retraction*. This can be seen as a smooth function that pragmatically approximates the notion of geodesics [10]. Now we present its rigorous definition.

DEFINITION 2.1 ([2]). *A retraction on a manifold $\mathcal{M}$ is a smooth mapping $R[\cdot]$ from the tangent bundle $\mathcal{T}\mathcal{M}$ onto $\mathcal{M}$ with the following properties. Let $R_X[\cdot]$ denote the restriction of $R[\cdot]$ to $T_X\mathcal{M}$.*

a. *$R_X[0_X] = X$, where 0_X denotes the zero element of $T_X\mathcal{M}$*
b. *With the canonical identification, $T_{0_X}T_X\mathcal{M} \simeq T_X\mathcal{M}$, $R_X[\cdot]$ satisfies*

$$\mathcal{D}R_X[0_X] = id_{T_X\mathcal{M}},$$

where $id_{T_X\mathcal{M}}$ denotes the identity mapping on $T_X\mathcal{M}$.

The second condition in Definition 2.1 is known as *local rigidity condition*.

The concept of vector transport, which appears in [2], provides us a tool to perform operations between two or more vectors that belong to different tangent spaces of $\mathcal{M}$, and can be seen as a relaxation of the purely Riemannian concept of *parallel transport* [10].

DEFINITION 2.2 ([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,

$$\mathcal{TM} \oplus \mathcal{TM} = \{(\eta, \xi) : \eta, \xi \in T_X\mathcal{M}, X \in \mathcal{M}\}.$$

a. *There exists a retraction $R[\cdot]$, called the retraction associated with $\mathcal{T}[\cdot]$, such that*

$$\pi(\mathcal{T}_\eta[\xi]) = R_X(\eta), \quad \eta, \xi \in T_X\mathcal{M},$$

where $\pi(\mathcal{T}_\eta[\xi])$ denotes the foot of the tangent vector $\mathcal{T}_\eta[\xi]$.

b. *$\mathcal{T}_{0_X}[\xi] = \xi$ for all $\xi \in T_X\mathcal{M}$.*

c. *$\mathcal{T}_\eta[a\xi + b\zeta] = a\mathcal{T}_\eta[\xi] + b\mathcal{T}_\eta[\zeta]$, for all $a, b \in \mathbb{R}$ and $\eta, \xi, \zeta \in T_X\mathcal{M}$.*

Next, the concept of isometry [34] is established, which is a property satisfied by some vector transports.

DEFINITION 2.3 ([34]). *A vector transport $\mathcal{T}[\cdot]$ on a manifold $\mathcal{M}$ is called isometric if it satisfies*

$$\langle \mathcal{T}_\eta[\xi], \mathcal{T}_\eta[\xi] \rangle = \langle \xi, \xi \rangle, \tag{2.2}$$

for all $\eta, \xi \in T_X\mathcal{M}$, where $R_X[\cdot]$ is the retraction associated with $\mathcal{T}[\cdot]$.

3. Spectral Approach for Tangent Vector Field on Riemannian Manifolds. In this section, we shall establish our proposal RSANE. An intuitive way to solve (1.1) is to promote the reduction of the residual $||F(\cdot)||$, which we can achieve by solving the following auxiliar manifold constrained optimization problem

$$\min_{X \in \mathcal{M}} \mathcal{F}(X) = \frac{1}{2}||F(X)||^2. \tag{3.1}$$

We can deal with this optimization model using some Riemannian optimization method. Nevertheless, we are interested in directly solving the Riemannian nonlinear equation (1.1). For this purpose, we consider the following iterative method, based on the SANE method,

$$X_{k+1} = X_k - \tau_k F(X_k). \tag{3.2}$$

Firstly, the vector $-F(X_k)$ is not necessarily a descent direction for the merit function $\mathcal{F}(\cdot)$ and secondly, that X_{k+1} does not necessarily belong to the manifold $\mathcal{M}$. We can overcome the first one, by modifying this vector with the sign of $\langle \nabla_{\mathcal{M}}\mathcal{F}(X_k), F(X_k) \rangle$, following the same idea used in SANE method, in order to force $\pm F(X_k)$ satisfies the descent condition. Observe that the Riemannian gradient method of $\mathcal{F}(\cdot)$ can be computed by

$$\nabla_{\mathcal{M}}\mathcal{F}(X) = (JF(X))^*[F(X)], \quad \forall X \in \mathcal{M}, \tag{3.3}$$

that is, to compute the Riemannian gradient of $\mathcal{F}(\cdot)$ at X, we need to calculate the adjoint of the Jacobian of $F(\cdot)$ evaluated at X.

On the other hand, the second disadvantage can be easily remedied by incorporating a retraction, similarly to the scheme (1.4). Keeping in mid all theses considerations, we now propose our Riemannian spectral residual method, which computes the iterates recursively by

$$X_{k+1} = R_{X_k}[\tau_k Z_k], \tag{3.4}$$

where $\tau_k > 0$ represents the step–size, $R_X[\cdot]$ is a retraction and the search direction is determined by

$$Z_k = -s_\theta(\langle \nabla_{\mathcal{M}} \mathcal{F}(X_k), F(X_k)\rangle)F(X_k), \tag{3.5}$$

where $s_\theta(\cdot) : \mathbb{R} - \{0\} \to \{-1, 1\}$ is defined by $s(x) = \frac{x}{|x|}$. Observe that $s(\cdot)$ is a continuous function for all $x \neq 0$, a crucial property to our convergence analysis.

3.1. A Nonmonotone Line Search with a Riemannian Spectral Parameter. In the scenario of the solution of nonlinear systems of equations over $\mathbb{R}^n$, the SANE method uses a spectral parameter τ_k inspired by the Barzilai–Borwein step–size, originally introduced in [4] to speed up the convergence of gradient–type methods, in the context of optimization. SANE computes this spectral parameter as follow

$$\tau_{k+1}^{BB} = \mathrm{sgn}_k \frac{S_k^\top S_k}{S_k^\top Y_k}, \tag{3.6}$$

where $S_k = X_{k+1} - X_k$, $Y_k = F(X_{k+1}) - F(X_k)$, $\mathrm{sgn}_k = s(F(X_k)^\top J(X_k)F(X_k))$ and $J(\cdot)$ denotes the Jacobian matrix of the vector–valued function $F : \mathbb{R}^n \to \mathbb{R}^n$.

In the framework of Riemannian manifolds, the vectors $F(X_{k+1}) \in T_{X_{k+1}}\mathcal{M}$ and $F(X_k) \in T_{X_k}\mathcal{M}$ lie in different tangent spaces, then the difference between these vectors may not be well defined (this is only well defined over linear manifolds). The same drawback occurs with the difference between the points X_{k+1} and X_k. Therefore, we cannot directly use the parameter (3.6) to address the numerical solution of (1.1). In [12] Iannazzo et. al. was extended the Barzilai–Borwein step–sizes in the context of optimization on Riemannian manifolds, through the use of a vector transport (see Definition 2.2). This strategy transports the calculated directions to the correct tangent space, providing a way to overcome the drawback. Guided by the descriptions contained in [12], we propose the following generalization of the spectral parameter (3.6),

$$\tau_{k+1}^{RBB1} = s(\sigma_k)\frac{\langle \hat{S}_k, \hat{S}_k\rangle}{\langle \hat{S}_k, \hat{Y}_k\rangle}, \tag{3.7}$$

where $\sigma_k = \langle \nabla_{\mathcal{M}} \mathcal{F}(X_k), F(X_k)\rangle$,

$$\hat{S}_k = \mathcal{T}_{\tau_k Z_k}[\tau_k Z_k] = -\tau_k s(\sigma_k)\mathcal{T}_{\tau_k Z_k}[F(X_k)] \tag{3.8}$$

and

$$\hat{Y}_k = F(X_{k+1}) - \mathcal{T}_{\tau_k Z_k}[F(X_k)] = F(X_{k+1}) + \frac{1}{\tau_k s(\sigma_k)}\hat{S}_k, \tag{3.9}$$

where $\mathcal{T}[\cdot]$ is any vector transport satisfying the Ring–Wirth non–expansive condition,

$$||\mathcal{T}_{\eta_X}[\xi_X]|| \leq ||\xi_X||, \quad \forall \xi_X, \eta_X \in T_X\mathcal{M}. \tag{3.10}$$

Another alternative for the spectral parameter is

$$\tau_{k+1}^{RBB2} = s(\sigma_k)\frac{\langle \hat{S}_k, \hat{Y}_k\rangle}{\langle \hat{Y}_k, \hat{Y}_k\rangle}. \tag{3.11}$$

In order to take advantage of both spectral parameters τ_{k+1}^{RBB1} and τ_{k+1}^{RBB2}, we adopt the following adaptive strategy

$$\tau_{k+1}^{RBB} = \begin{cases} \tau_{k+1}^{RBB1} & \text{for even k;} \\ \tau_{k+1}^{RBB2} & \text{for odd k.} \end{cases} \tag{3.12}$$

Note that it is always possible to define a transporter (a function that sends vectors from a tangent space to another tangent space) that satisfies the condition (3.10), by scaling,

$$\mathcal{T}_{\eta_X}^{\text{scaled}}[\xi_X] = \begin{cases} \mathcal{T}_\eta[\xi] & \text{if } ||\mathcal{T}_{\eta_X}[\xi_X]|| \leq ||\xi_X||; \\ \frac{||\xi||}{||\mathcal{T}_\eta[\xi]||}\mathcal{T}_\eta[\xi] & \text{otherwise.} \end{cases} \tag{3.13}$$

This function, introduced by Sato and Iwai in [27], is referred as *scaled vector transport*. Observe that (3.13) is not necessarily a vector transport. However for the extension of the spectral parameter to the setting of Riemannian manifolds, it is not strictly mandatory. In fact, it is enough having a non–expansive transporter available. Therefore, we will use the scaled vector transport (3.13), in the construction of the vectors $\hat{S}_k$ and $\hat{Y}_k$ in equations (3.8)–(3.9).

Since the spectral parameter τ_k^{RBB} does not necessarily reduces the value of the merit function $\mathcal{F}(\cdot)$ at each iteration, the convergence result could be invalid. We can overcome this drawback by incorporating a globalization strategy, which guarantees convergence by regulating the step–size τ_k only occasionally (see [7, 23, 31]). In the seminal paper [7], the authors consider the globalization technique proposed by Grippo et.al. in [11], in the definition of SANE method.

We could define our Riemannian generalization of SANE incorporating this non–monotone technique, and so analyze the convergence following the ideas described in [7, 12]. Instead of that, in this work, to define RSANE we adopt a more elegant globalization strategy proposed by Zhang and Hager in [31]. Specifically, we compute $\tau_k = \delta^h \tau_k^{RBB}$ where $h \in \mathbb{N}$ is the smallest integer satisfying

$$\mathcal{F}(R_{X_k}[\tau_k Z_k]) \leq C_k + \rho_1 \tau_k \langle \nabla_{\mathcal{M}} \mathcal{F}(X_k), Z_k\rangle, \tag{3.14}$$

where each value C_{k+1} is given by a convex combination of $\mathcal{F}(X_{k+1})$ and the previous C_k as

$$C_{k+1} = \frac{\eta Q_k C_k + \mathcal{F}(X_{k+1})}{Q_{k+1}},$$

for $Q_{k+1} = \eta Q_k + 1$, starting at $Q_0 = 1$ and $C_0 = \mathcal{F}(X_0)$. In the sequel our generalization RSANE will be described in detail.

Algorithm 3.1 Spectral Residual Method for tangent vector field (RSANE)

Require: Let $X_0 \in \mathcal{M}$ be the initial guess, $\tau > 0$, $0 < \tau_m \leq \tau_M < \infty$, $\eta \in [0, 1)$, $\rho_1, \epsilon, \epsilon_1, \delta \in (0, 1)$, $Q_0 = 1$, $C_0 = \mathcal{F}(X_0)$, $k = 0$.

1: **while** $||F(X_k)|| > \epsilon$ **do**
2: $\sigma_k = \langle \nabla_{\mathcal{M}} \mathcal{F}(X_k), F(X_k) \rangle$,
3: **if** $|\sigma_k| < \epsilon_1 ||F(X_k)||^2$ **then**
4: stop the process.
5: **end if**
6: $Z_k = -s(\sigma_k) F(X_k)$,
7: **while** $\mathcal{F}(R_{X_k}[\tau Z_k]) > C_k - \rho_1 \epsilon_1 \tau ||F(X_k)||^2$ **do**
8: $\tau \leftarrow \delta \tau$,
9: **end while**
10: $\tau_k = \tau$,
11: $X_{k+1} = R_{X_k}[\tau_k Z_k]$,
12: $Q_{k+1} = \eta Q_k + 1$ and $C_{k+1} = (\eta Q_k C_k + \mathcal{F}(X_{k+1}))/Q_{k+1}$,
13: $\hat{S}_k = -\tau_k s(\sigma_k) \mathcal{T}_{\tau_k Z_k}[F(X_k)]$ and $\hat{Y}_k = F(X_{k-1}) + \frac{1}{\tau_k s(\sigma_k)} \hat{S}_k$,
14: $\tau_{k+1}^{RBB} = s(\sigma_k) \frac{\langle \hat{S}_k, \hat{S}_k \rangle}{\langle \hat{S}_k, \hat{Y}_k \rangle}$,
15: $\tau = \max(\min(\tau_{k+1}^{RBB}, \tau_M), \tau_m)$.
16: $k \leftarrow k + 1$.
17: **end while**

Remark 3.1. In Algorithm 3.1, we replace the nonmonotone condition (3.14) by

$$\mathcal{F}(R_{X_k}[\tau Z_k]) \leq C_k - \rho_1 \epsilon_1 \tau ||F(X_k)||^2. \tag{3.15}$$

We remark that with this relaxed condition, Algorithm 3.1 is well defined. In fact, if at iteration k the procedure does not stop at Step 4, then Z_k is a descent direction (see Lemma 3.4), and for all $\rho_1 \in (0, 1)$ there exists $t > 0$ such that the non–monotone Zhang–Hager condition (3.14) holds by continuity, for $\tau > 0$ sufficiently small(a proof of this fact appears in [31]). In addition, it follows form Step 4, (3.14) and Lemma 3.4 that

$$\begin{aligned} \mathcal{F}(R_{X_k}[\tau Z_k]) &\leq C_k + \rho_1 \tau \langle \nabla_{\mathcal{M}} \mathcal{F}(X_k), Z_k \rangle \\ &\leq C_k - \rho_1 \epsilon_1 \tau ||F(X_k)||^2, \end{aligned}$$

which implies that the relaxed condition (3.15) is also verified for all $\tau > 0$ that satisfy (3.14).

Remark 3.2. The bottleneck of Algorithm 3.1 appears in step 2, since to calculate σ_k, we must compute the Riemannian gradient of $\mathcal{F}(\cdot)$, which implies evaluating the Jacobian of $F(\cdot)$ (see equation (3.3)). However, given a retraction $R_X[\cdot]$, σ_k can be approximated using finite differences as follow

$$\begin{aligned} \sigma_k &= \langle \nabla_{\mathcal{M}} \mathcal{F}(X_k), F(X_k) \rangle \\ &= \mathcal{D}\mathcal{F}(X_k)[F(X_k)] \\ &= \lim_{t \to 0} \frac{\mathcal{F}(R_{X_k}[tF(X_k)]) - \mathcal{F}(X_k)}{t} \\ &\approx \frac{\mathcal{F}(R_{X_k}[hF(X_k)]) - \mathcal{F}(X_k)}{h}, \end{aligned} \tag{3.16}$$

where $0 < h \ll 1$ is a small real number. The fact that this approximation does not need the explicit knowledge of the Jacobian operator is useful for large–scale problems.

The following lemma establishes that in most cases τ_k^{RBB1} is positive.

LEMMA 3.3. *Let τ_{k+1}^{RBB1} be computed as in (3.7), then $\tau_{k+1}^{RBB1} > 0$ when one of the following cases holds*

a. $\langle \mathcal{T}_{\tau_k Z_k}[F(X_k)], F(X_{k+1})\rangle < 0$;
b. $\langle \mathcal{T}_{\tau_k Z_k}[F(X_k)], F(X_{k+1})\rangle > 0$ *and* $||F(X_{k+1})|| < ||\mathcal{T}_{\tau_k Z_k}[F(X_k)]||$.

Proof. Since $\tau_{k+1}^{RBB1} = s(\sigma_k)\frac{\langle \hat{S}_k, \hat{S}_k\rangle}{\langle \hat{S}_k, \hat{Y}_k\rangle} = \frac{-1}{\tau_k}\frac{||\hat{S}_k||^2}{\langle \mathcal{T}_{\tau_k Z_k}[F(X_k)], \hat{Y}_k\rangle}$ then $\texttt{sign}(\tau_{k+1}^{RBB1}) = \texttt{sign}(-\langle \mathcal{T}_{\tau_k Z_k}[F(X_k)], \hat{Y}_k\rangle)$. Suppose that (a) holds, then

$$\langle \mathcal{T}_{\tau_k Z_k}[F(X_k)], \hat{Y}_k\rangle = \langle \mathcal{T}_{\tau_k Z_k}[F(X_k)], F(X_{k+1})\rangle - ||\mathcal{T}_{\tau_k Z_k}[F(X_k)]||^2 < 0,$$

which implies that $\tau_{k+1}^{RBB} > 0$.

On the other hand, if (b) holds, from the Cauchy–Schwarz inequality, we find that

$$0 < \langle \mathcal{T}_{\tau_k Z_k}[F(X_k)], F(X_{k+1})\rangle \leq ||\mathcal{T}_{\tau_k Z_k}[F(X_k)]||\,||F(X_{k+1})|| < ||\mathcal{T}_{\tau_k Z_k}[F(X_k)]||^2,$$

and so

$$\langle \mathcal{T}_{\tau_k Z_k}[F(X_k)], \hat{Y}_k\rangle = \langle \mathcal{T}_{\tau_k Z_k}[F(X_k)], F(X_{k+1})\rangle - ||\mathcal{T}_{\tau_k Z_k}[F(X_k)]||^2 < 0,$$

and hence $\tau_{k+1}^{RBB1} > 0$, which proves the lemma. ∎

The same theoretical result is verified for spectral parameter τ_{k+1}^{RBB2}, and therefore it is also valid for the adaptive parameter τ_{k+1}^{RBB}.

Notice that if the transporter $\mathcal{T}[\cdot]$ is isometric (see Definition 2.3), then the second condition of item (b) in Lemma 3.3 is reduced to $||F(X_{k+1})|| < ||F(X_k)||$. In addition, observe that if we set $\eta = 0$ in Algorithm 3.1, then we have that $\mathcal{F}(X_{k+1}) < \mathcal{F}(X_k)$ for all $k \in \mathbb{N}$. Therefore this condition would always be verified under these choices of η and $\mathcal{T}[\cdot]$.

LEMMA 3.4. *Assume that Algorithm 3.1 does not terminate. Let $\{Z_k\}$ be an infinite sequence of tangent direction generated by Algorithm 3.1. Then Z_k is a descent direction for the merit function $\mathcal{F}(\cdot)$ at X_k, for all $k \geq 0$.*

Proof. From Step 6 in Algorithm 3.1 we have $Z_k = -s(\sigma_k)F(X_k)$, where $\sigma_k = \langle \nabla_{\mathcal{M}}\mathcal{F}(X_k), F(X_k)\rangle$. Since Algorithm 3.1 does not terminate, form Step 3 we obtain $|\sigma_k| \geq \epsilon_1 ||F(X_k)||^2 > 0$. Now, observe that

$$\begin{aligned}
\langle \nabla_{\mathcal{M}}\mathcal{F}(X_k), Z_k\rangle &= -s(\sigma_k)\langle \nabla_{\mathcal{M}}\mathcal{F}(X_k), F(X_k)\rangle \\
&= -s(\sigma_k)\sigma_k \\
&= -\frac{\sigma_k^2}{|\sigma_k|} \\
&= -|\sigma_k| \\
&< 0.
\end{aligned}$$

Therefore, we conclude that Z_k is a descent direction for $\mathcal{F}(\cdot)$ at X_k, for all $k \geq 0$. ∎

4. Convergence Analysis. In this section, we analyse the global convergence for our Algorithm 3.1 under mild assumptions. Our analysis consists on a generalization of the global convergence of line–search methods for unconstrained optimization, presented in [31] and an adaptation of Theorem 4.3.1 in [2].

The following lemma establishes that C_k is bounded below by the sequence $\{\mathcal{X}_{\|}\}$.

LEMMA 4.1. *Let $\{X_k\}$ be an infinite sequence generated by Algorithm 3.1. Then for the iterates generated by Algorithm 3.1.*

$$\mathcal{F}(X_k) \leq C_k, \quad \forall k. \tag{4.1}$$

Proof. Firstly, we define $\psi_k : \mathbb{R} \to \mathbb{R}$ by

$$\psi(\alpha) = \frac{\alpha C_{k-1} + \mathcal{F}(X_k)}{\alpha + 1},$$

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

$$\dot{\psi}(\alpha) = \frac{C_{k-1} - \mathcal{F}(X_k)}{(\alpha+1)^2}.$$

it follows from the non–monotone condition (3.15) that

$$\mathcal{F}(X_k) \leq C_{k-1} - \rho_1 \epsilon_1 \tau ||F(X_k)||^2 < C_{k-1}, \tag{4.2}$$

which implies that $\dot{\psi}(\alpha) \geq 0$ for all $\alpha \geq 0$. Hence, the function $\psi(\cdot)$ is nondecreasing, and $\mathcal{F}(X_k) = \psi(0) \leq \psi(\alpha)$ for all $\alpha \geq 0$. Then, taking $\bar{\alpha} = \eta Q_{k-1}$ we obtain

$$\mathcal{F}(X_k) = \psi(0) \leq \psi(\bar{\alpha}) = C_k, \tag{4.3}$$

which completes the proof. ∎

In order to prove the global convergence of our proposed algorithm, we need the following asymptotic property.

LEMMA 4.2. *Any infinite sequence $\{X_k\}$ generated by Algorithm 3.1 verifies the following property*

$$\lim_{k\to\infty} \tau_k ||F(X_k)||^2 = 0. \tag{4.4}$$

Proof. By the construction of Algorithm 3.1 and using Lemma 3.4, we have

$$C_{k+1} = \frac{\eta Q_k C_k + \mathcal{F}(X_{k+1})}{Q_{k+1}} < \frac{(\eta Q_k + 1) C_k}{Q_{k+1}} = C_k. \tag{4.5}$$

Hence, $\{C_k\}$ is monotonically decreasing and bounded below by zero, therefore it converges to some limit $C^* \geq 0$. It follows from Step 7 and Step 12 in Algorithm 3.1 that

$$\sum_{k=0}^{\infty} \frac{\rho_1 \epsilon_1 \tau_k ||F(X_k)||^2}{Q_{k+1}} \leq \sum_{k=0}^{\infty} C_k - C_{k+1} = C_0 - C_* < \infty. \tag{4.6}$$

Merging this result with the fact that $Q_{k+1} = 1 + \eta Q_k = 1 + \eta + \eta^2 Q_{k-1} = \cdots = \sum_{i=0}^{k} \eta^i < (1-\eta)^{-1}$, we have

$$\lim_{k\to\infty} \tau_k ||F(X_k)||^2 = 0, \tag{4.7}$$

which proves the lemma. ∎

The theorem below establishes a global convergence property concerning Algorithm 3.1. The proof can be seen as a modification to that of Theorem 3.4 in [30], and to that of Theorem 4.1 in [18].

THEOREM 4.3. *Algorithm 3.1 either terminates at a finite iteration $j \in \mathbb{N}$ where $|\langle \nabla_{\mathcal{M}} \mathcal{F}(X_j), F(X_j)\rangle| < \epsilon ||F(X_j)||^2$, or it generates an infinite sequence $\{X_k\}$ such that*

$$\liminf_{k\to\infty} ||F(X_k)|| = 0.$$

Proof. Let us assume that Algorithm 3.1 does not terminate, and let X_* be any accumulation point of the sequence $\{X_k\}$. We may assume that $\lim_{k\to\infty} X_k = X_*$, taking a subsequence if necessary. By contradiction, suppose that there exists $\epsilon_0 > 0$ such that

$$||F(X_k)||^2 > \epsilon_0, \quad \forall k \geq 0. \tag{4.8}$$

In view of (4.8) and Lemma 4.2 we have

$$\liminf_{k\to\infty} \tau_k = 0. \tag{4.9}$$

Firstly, we define the curve $Y_k(\tau) = R_{X_k}[-\tau Z_k]$ for all $k \in \mathbb{N}$, which is smooth due to the differentiability of the retraction $R_X[\cdot]$. Since the parameter $\tau_k > 0$ is chosen by carrying out a backtracking process, then $\tau_k = \delta^{m_k}\tau_k^{RBB}$, for all k greater than some $\bar{k}$, where $m_k \in \mathbb{N}$ is the smallest positive integer number such that the relaxed nonmonotone condition (3.15) is fulfilled. Thus, the scalar $\bar{\tau} = \frac{\tau_k}{\delta}$ violates the condition (3.15), i.e., it holds

$$-\rho_1\epsilon_1\bar{\tau}||F(X_k)||^2 < \mathcal{F}(Y_k(\bar{\tau})) - C_k \leq \mathcal{F}(Y_k(\bar{\tau})) - \mathcal{F}(X_k), \quad \forall k \geq \bar{k}, \tag{4.10}$$

where the last inequality is obtained using Lemma 4.1.

Let us set $\psi_k(\tau) := \mathcal{F} \circ Y_k(\tau)$, then (4.10) is equivalent to

$$-\frac{\psi_k(\bar{\tau}) - \psi_k(0)}{\bar{\tau} - 0} < \rho_1\epsilon_1||F(X_k)||^2, \quad \forall k \geq \bar{k}. \tag{4.11}$$

It follows from the mean value theorem, that there exists $t \in (0, \bar{\tau})$ such that $-\dot{\psi}_k(t) < \rho_1\epsilon_1||F(X_k)||^2$ for all $k \geq \bar{k}$, or equivalently

$$-\langle \nabla_{\mathcal{M}} \mathcal{F}(Y_k(t)), \dot{Y}_k(t)\rangle < \rho_1\epsilon_1||F(X_k)||^2, \quad \forall k \geq \bar{k}. \tag{4.12}$$

In view of the continuity of functions $s(\cdot)$, $\nabla_{\mathcal{M}}\mathcal{F}(\cdot)$, $F(\cdot)$, $Y_k(\cdot)$, the smoothness and local rigidity condition of the retraction $R_X[\cdot]$, and taking limit in (4.12), we arrive at

$$-\langle \nabla_{\mathcal{M}} \mathcal{F}(X_*), Z_*\rangle \leq \rho_1\epsilon_1||F(X_*)||^2,$$

or equivalently,

$$|\sigma_*| \leq \rho_1\epsilon_1||F(X_*)||^2, \tag{4.13}$$

where $|\sigma_*| = |\langle \nabla_{\mathcal{M}} \mathcal{F}(X_*), F(X_*)\rangle|$. Since Algorithm 3.1 does not terminate, form Step 3 we have

$$|\sigma_k| \geq \epsilon_1||F(X_k)||^2. \tag{4.14}$$

Applying limits in (4.14) we find that $|\sigma_*| \geq \epsilon_1 ||F(X_*)||^2$. Merging this last result with (4.13), we arrive at

$$||F(X_*)||^2 \leq \rho_1 ||F(X_*)||^2.$$

Since $0 < \rho_1 < 1$ then we have $||F(X_*)|| = 0$, this last result contradicts (4.8), which completes the proof. ∎

By Theorem 4.3, we obtain the following theoretical consequence under compactness assumptions.

Corollary 4.4. Let $\{X_k\}$ be an infinite sequence generated by Algorithm 3.1. Suppose that the level set $\mathcal{L} = \{X \in \mathcal{M} : \mathcal{F}(X) \leq \mathcal{F}(X_0)\}$ is compact (which holds in particular when the Riemannian manifold $\mathcal{M}$ is compact). Then

$$\lim_{k\to\infty} ||F(X_k)|| = 0.$$

Proof. Let $\{X_k\}$ be an infinite sequence generated by Algorithm 3.1. It follows from Lemma 4.1, the strict inequality (4.5) and the construction of Algorithm 3.1 that

$$\mathcal{F}(X_{k+1}) \leq C_{k+1} < C_k < \cdots < C_1 < C_0 = \mathcal{F}(X_0),$$

which implies that $X_k \in \mathcal{L}$, for all $k \geq 0$.

Now, by contradiction let us suppose that there is a subsequence $\{X_k\}_{k\in\mathcal{K}}$ and $\epsilon_0 > 0$ such that

$$||F(X_k)|| \geq \epsilon_0, \tag{4.15}$$

for any $k \in \mathcal{K}$. Since $X_k \in \mathcal{L}$, for all $k \geq 0$ and L is a compact set, we have that $\{X_k\}_{k\in\mathcal{K}}$ must have an accumulation point $X_* \in \mathcal{L}$. Taking limit in (4.15) and using the continuity of $||F(\cdot)||$, we obtain $||F(X_*)|| \geq \epsilon_0$, which contradicts Theorem 4.3. ∎

Remark 4.5. The main drawback of Algorithm 3.1 is that it can prematurely terminate with a *bad breakdown* ($|\sigma_k| < \epsilon_1 ||F(X_k)||^2$). One way to remedy this problem is to use $Z_k = -\nabla\mathcal{F}(X_k)$ as the search direction, if a *bad breakdown* occurs at k–th iteration; and then in the next iteration, we can retry using the steps of Algorithm 3.1. We may even use any tangent direction $Z_k \in T_{X_k}\mathcal{M}$ such that $\langle \nabla\mathcal{F}(X_k), Z_k\rangle < 0$, as long as a *bad breakdown* happens, in order to overcome this difficulty.

5. Numerical Experiments. In order to give further insight into the RSANE method we present the results of some numerical experiments. We test our algorithm on some randomly generated gradient tangent vector fields on three different Riemannian manifold, involving the unit sphere, the Stiefel manifold and oblique manifold. All experiments have been performed on a intel(R) CORE(TM) i7–4770, CPU 3.40 GHz with 1TB HD and 16GB RAM memory. The algorithm was coded in Matlab with double precision. The running times are always given in CPU seconds. For numerical comparisons, we consider the SANE method proposed in [7], and the recently published Riemannian Derivative–Free Conjugate gradient Polak–Ribiére–Polyak method (CGPR) for the numerical solution of tangent vectors field [30]. The Matlab codes of our RSANE, SANE and CGPR are available in: http://www.optimization-online.org/DB_HTML/2020/09/8028.html

6. Implementation details. In our implementation, in addition to monitoring the residual norm $||F(X_k)||_F$, we also check the relative changes of the two consecutive iterates and their corresponding residual values

$$\text{rel}_k^X := \frac{||X_{k+1} - X_k||_F}{||X_k||_F} \quad \text{and} \quad \text{rel}_k^F := \frac{|\mathcal{F}(X_{k+1}) - \mathcal{F}(X_k)|}{\mathcal{F}(X_k) + 1}. \tag{6.1}$$

Here $||M||_F$ denotes the Frobenius norm of the matrix M. In the case when M is a vector, this norm is reduced to the standard norm on $\mathbb{R}^n$. Although the residual $||X_{k+1} - X_k||_F$ is meaningless in the Riemannian context, for our numerical experiments, we will only consider Riemannian manifolds embedded in the Euclidean space $\mathbb{R}^{n\times p}$, and the residual rel_k^X is well defined for these types of manifolds.

We let all the algorithms run up to K iterations and stop them at iteration $k < K$ if $||F(X_k)||_F < \epsilon$, or $\text{rel}_k^X < \epsilon_X$ and $\text{rel}_k^F < \epsilon_F$, or

$$\text{mean}([\text{rel}^X_{k-\min k,T+1}, \ldots, \text{rel}^X_k]) \leq 10\epsilon_X \text{ and } \text{mean}([\text{rel}^F_{k-\min k,T+1}, \ldots, \text{rel}^F_k]) \leq 10\epsilon_F.$$

Here, the defaults values of $\epsilon, \epsilon_X, \epsilon_F$ and T are 1e-5, 1e-15, 1e-15 and 5, respectively. In addition, in Algorithm 3.1 we use $\eta = 0.6$, $\tau =$ 1e-3 (the initial step–size τ), $\tau_m =$ 1e-10, $\tau_m =$ 1e+10, $\delta = 0.2$, $\epsilon_1 = 1e-8$ and $\rho_1 =$ 1e-4 as defaults values.

6.1. Considered manifolds and their geometric tools. In this subsection, we present three Riemannian manifolds that we will use for the numerical experiments in the remainder subsections, as well as some tools necessary for the algorithms, associated with each manifold, such as vectors transports and retractions.

Firstly we considere the *unit sphere* given by

$$S^{n-1} = \{x \in \mathbb{R}^n : ||x||_2 = 1\}. \tag{6.2}$$

It is well–known that the tangent space of the unit sphere at $x \in S^{n-1}$ is given by $T_xS^{n-1} = \{z \in \mathbb{R}^n : z^\top x = 0\}$. Let S^{n-1} be endowed with the inner product inherited from the classical inner product on $\mathbb{R}^n$,

$$\langle \xi_x, \eta_x \rangle_x := \xi_x^\top \eta_X, \quad \forall \xi_x, \eta_x \in T_xS^{n-1},\ x \in S^{n-1}.$$

Then, S^{n-1} with this inner product defines an $n-1$ dimensional Riemannian sub–manifold of $\mathbb{R}^n$. The retraction $R_x[\cdot]$ on S^{n-1} is chosen as in [2],

$$R_x[\xi_x] = \frac{x + \xi_x}{||x + \xi_x||_2},$$

for all $\xi_x \in T_xS^{n-1}$ and $x \in S^{n-1}$. In addition, for this particular manifold, we consider the vector transport based on orthogonal projection

$$\mathcal{T}_{\eta_x}[\xi_x] = \xi_x - R_x[\eta_x]R_x[\eta_x]^\top \xi_x.$$

Notice that this vector transport verifies the Ring–Wirth non-expansive condition (3.10).

The second Riemannian manifold considered in this section is the *Stiefel manifold*, which is defined as

$$St(n,p) = \{X \in \mathbb{R}^{n\times p} : X^\top X = I_p\}, \tag{6.3}$$

where $I_p \in \mathbb{R}^{p\times p}$ denotes the identity matrix of size p–by–p. By differentiating both sides of $X(t)^\top X(t) = I_p$, we obtain the tangent space of $St(n,p)$ at X, given by $T_X St(n,p) = \{Z \in \mathbb{R}^{n\times p} : Z^\top X + X^\top Z = 0\}$. Let $St(n,p)$ be endowed with induced Riemannian metric from $\mathbb{R}^{n\times p}$, i.e.,

$$\langle \xi_X, \eta_X\rangle_X := tr(\xi_X^\top \eta_X), \quad \forall \xi_x, \eta_x \in T_X St(n,p),\ X \in St(n,p). \tag{6.4}$$

The pair $(St(n,p), \langle\cdot,\cdot\rangle)$, where $\langle\cdot,\cdot\rangle$ is the inner product given in (6.4), forms a Riemannian sub–manifold of the Euclidean space $\mathbb{R}^{n\times p}$, and its dimension is equal to $np - \frac{1}{2}p(p+1)$, [2]. For our numerical experiments concerning the Stiefel manifold we will use the retractions, introduced in [2], given by

$$R_X[\xi_X] = \mathtt{qf}(X + \xi_X), \tag{6.5}$$

and the retraction based on the matrix polar decomposition

$$R_X[\xi_X] = (X + \xi_X)\left((X + \xi_X)^\top (X + \xi_X)\right)^{-1/2}, \tag{6.6}$$

for all $\xi_X \in T_X St(n,p)$ and $X \in St(n,p)$. In equation (6.5), $\mathtt{qf}(W)$ denotes the orthogonal factor Q obtained form the QR–factorization of W, such that $W = QR$, where Q belongs $St(n,p)$ and $R \in \mathbb{R}^{p\times p}$ is the upper triangular matrix with strictly positive diagonal elements. Additionally, we consider the following vector transport

$$\mathcal{T}_{\eta_X}[\xi_X] = \xi_X - R_X[\eta_X]\mathtt{sym}(R_X[\eta_X]^\top \xi_X), \tag{6.7}$$

where $R_X[\cdot]$ is any of the two retractions defined in (6.5)–(6.6), and $\mathtt{sym}(W) = \frac{1}{2}(W^\top + W)$ is the function that assigns to the matrix W its symmetric part. This vector transport is inspired by the orthogonal projection on the tangent space of the Stiefel manifold. Moreover, the function (6.7) verifies the Ring–Wirth non-expansive condition (3.10).

Our third example, the *oblique manifold*, is defied as

$$\mathcal{OB}(n,p) = \{X \in \mathbb{R}^{n\times p} : \mathtt{ddiag}(X^\top X) = I_p\}, \tag{6.8}$$

where $\mathtt{ddiag}(W)$ denotes the matrix W with all its off–diagonal entries assigned to zero. The tangent space associated to $\mathcal{OB}(n,p)$ at X is given by $T_X\mathcal{OB}(n,p) = \{Z \in \mathbb{R}^{n\times p} : \mathtt{ddiag}(X^\top Z) = 0\}$. Again, if we endow $OB(n,p)$ with the inner product inherited from the standard inner product on $\mathbb{R}^{n\times p}$, given by (6.4), then the $\mathcal{OB}(n,p)$ becomes an embedded Riemannian manifold on $\mathbb{R}^{n\times p}$. For this particular manifold, we consider the retraction, which appears in [1], defined by

$$R_X[\xi_X] = (X + \xi_X)\mathtt{ddiag}\left((X + \xi_X)^\top (X + \xi_X)\right)^{-1/2}, \tag{6.9}$$

for all $\xi_X \in T_X\mathcal{OB}(n,p)$ and $X \in \mathcal{OB}(n,p)$. We will use another vector transport based on the orthogonal projection operator on $T_X\mathcal{OB}(n,p)$,

$$\mathcal{T}_{\eta_X}[\xi_X] = \xi_X - R_X[\eta_X]\mathtt{ddiag}(R_X[\eta_X]^\top \xi_X), \tag{6.10}$$

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

6.2. Eigenvalues computation on the sphere. For the first test problem, we consider the standard eigenvalue problem

$$Ax = \lambda x, \tag{6.11}$$

where $A \in \mathbb{R}^{n\times n}$ is a given symmetric matrix. The values of λ that verify (6.11) are known as eigenvalues of A and the corresponding vectors $x \in \mathbb{R}^n$ are the eigenvectors. A simple way to compute extreme eigenvalues of A is by minimizing (or maximizing) the associated Rayleigh quotient

$$r(x) = \frac{x^\top Ax}{x^\top x}. \tag{6.12}$$

This function is a continuously differentiable map $r : \mathbb{R}^n - \{0\} \to \mathbb{R}$, whose gradient is given by $\nabla r(x) = \frac{2}{||x||_2^2}(Ax - r(x)x)$. It is clear that any eigenvector x and its corresponding eigenvalue λ satisfy that $r(x) = \lambda$, and thus in that case x, is a critical point of $r(\cdot)$, i.e., $\nabla r(x) = 0$. By noting that $\nabla r(x) = 0$ if, and only if x is a solution of the following nonlinear system of equations

$$G(x) \equiv Ax - r(x)x = 0, \tag{6.13}$$

we can address directly this eigenvalue problem by using SANE method. On the other hand, by introducing the constraint $||x||_2 = 1$, the nonlinear system $G(x) = 0$ can be cast to the following tangent vector field, $F : S^{n-1} \to \mathcal{T}S^{n-1}$ defined on a Riemannian manifold (the unit sphere)

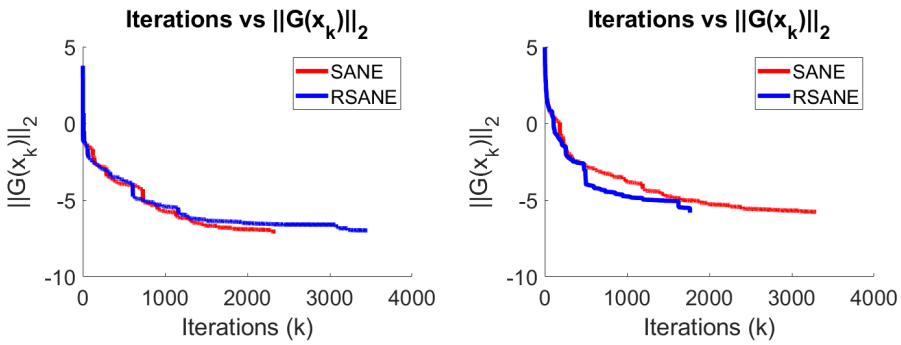
$$F(x) \equiv Ax - (x^\top Ax)x = 0_x. \tag{6.14}$$

Note that for all $x \in S^{n-1}$, we have $x^\top F(x) = 0$. Therefore $F(\cdot)$ effectively maps point on the sphere to its corresponding tangent space, i.e., $F(\cdot)$ defines a tangent vector field.

For illustrative purposes, in this subsection we compare the numerical performance of the SANE method solving (6.13), versus its generalized Riemannian version RSANE solving (6.14). To do this, we consider 40 instances of symmetric and positive definite sparse matrices taken from the UF sparse Matrix collection [8][1], which contains several matrices that arise in real applications. For this experiment, we stop both algorithms when is founded a vector x_k that satisfies the inequality $||G(x_k)||_2 < TOL$, where $TOL = \max\{\epsilon, \epsilon||G(x_0)||_2\}$, with $\epsilon =$ 2e-5. Note that for our proposed algorithm $F(x_k) = G(x_k)$, for all x_k generated by RSANE, since our proposal preserves the constraint $||x||_2 = 1$, hence this stop criterion is well defined for our algorithm. The starting vector $x_0 \in \mathbb{R}^n$ was generated by $x_0 = v/\sqrt{n}$, where $v = (1, 1, \ldots, 1)^\top$.

The numerical results associated to this experiments are summarized in Table 1. In this table, we report the number of iteration (Nitr); the CPU–time in seconds (Time); the residual value $\min\left\{\frac{||G(x_*)||_2}{||x_0||_2}, ||G(x_*)||_2\right\}$ (NrmF), where x_* denotes the estimated solution obtained by each algorithm; and the number of function evaluations (Nfe). In the case of SANE this is the number of times that SANE evaluates the $G(\cdot)$ map, while for our RSANE, Nfe denotes the number of times that it evaluates

[1] The SuiteSparse Matrix Collection tool–box is available in https://sparse.tamu.edu/



(a) 1138_bus (b) s3rmt3m1

FIG. 1. *Convergence history of SANE and RSANE for the instances "1138_bus" and "s3rmt3m1". The y–axis is in logarithmic scale.*

the functional $F(\cdot)$.

We observe, from Table 1, that the RSANE algorithm is a robust option to solve linear eigenvalue problems. The performance of both SANE and RSANE varied over different matrices. We note that our RSANE is very competitive for large–scale problems. Indeed, the RSANE algorithm outperforms its Euclidean version (SANE) in general terms, since RSANE took on average a total of 3644.1 iterations and 31.167 seconds to solve all the 40 problems, while SANE took 34.079 seconds and 3864.3 iterations to solve all the instances on average. It is worth noting that although RSANE failed on problem *bcsstk13*, it can become successful on *bcsstk16*, while both methods take the maximum number of iterations on problems *bcsstk27*, *bcsstk28* and *nasa4704*.

In Figure 1, we plot the convergence history of SANE and RSANE methods for the particular instances "1138_bus" and "s3rmt3m1", respectively. From this figure, we note that RSANE can converge faster than SANE, which is illustrated in Figure 1 (b). However the opposite conclusion is obtained from Figure 1(a).

6.3. A nonlinear eigenvalue problem on the Stiefel manifold. As a second experiment, we investigate the performance our RSANE method applied to deal with the following Stiefel manifold constrained nonlinear eigenvalue problem

$$H(X)X - X\Lambda = 0, \tag{6.15a}$$

$$X^\top X - I_p = 0. \tag{6.15b}$$

where $H(X) = L + \mu\,\mathtt{ddiag}(L^\dagger\rho(X))$, where $L^\dagger \in \mathbb{R}^{n\times n}$ is the pseudo–inverse matrix of the discrete Laplacian operator L, $\mu > 0$ and $\rho(X) = \mathtt{diag}(XX^\top)$, here $\mathtt{diag}(M) \in \mathbb{R}^n$ denotes the vector containing the diagonal elements of the matrix $M \in \mathbb{R}^{n\times n}$. Observe that $\Lambda \in \mathbb{R}^{p\times p}$ can be seen as a block of eigenvalues of the nonlinear matrix $H(X)$. In fact, in the special case that the operator $H(X)$ is constant and $p = 1$, this problem is reduced to the linear eigenvalue problem (6.11). This kind of nonlinear eigenvalue problem appear frequently in total energy minimization in electronic structure calculations [6, 25]. Pre–multiplying by $X^\top$ both sides of (6.15a) and using (6.15b) we obtain $\Lambda = X^\top H(X)X$, so substituting this result in (6.15a),

TABLE 1
Numerical results for eigenvalues computation.

		SANE				RSANE			
Name	n	Nfe	NrmF	Nitr	Time	Nfe	NrmF	Nitr	Time
1138_bus	1138	10621	9.70e-6	2791	0.284	14778	7.21e-6	3781	0.449
af_0_k101	503625	2269	1.95e-5	692	51.875	2074	1.99e-5	644	49.273
af_1_k101	503625	1798	2.00e-5	560	39.915	1702	1.99e-5	451	33.397
af_2_k101	503625	2683	2.00e-5	796	59.179	1421	1.93e-5	451	34.259
af_3_k101	503625	2207	1.93e-5	664	49.175	1757	1.92e-5	541	42.234
af_4_k101	503625	2281	1.98e-5	694	51.156	1758	1.85e-5	545	43.613
af_5_k101	503625	2651	1.76e-5	789	58.730	1975	1.79e-5	607	47.731
af_shell3	504855	305	1.99e-5	119	6.989	328	1.79e-5	125	8.026
af_shell7	504855	452	1.14e-5	165	10.204	363	1.53e-5	133	8.851
apache1	80800	15377	1.88e-5	3992	14.466	34224	1.54e-5	8434	34.905
apache2	715176	58685	1.93e-5	13970	801.886	52821	6.89e-6	12516	785.019
bcsstk10	1086	2450	1.30e-5	737	0.063	1954	1.87e-5	603	0.054
bcsstk11	1473	883	1.76e-5	302	0.034	5421	2.00e-5	1501	0.199
bcsstk13	2003	19254	1.98e-5	3293	1.372	88608	3.08e-4	15000	6.523
bcsstk14	1806	76	7.15e-6	35	0.006	4674	1.94e-5	1341	0.283
bcsstk16	4884	105001	7.49e+0	15000	24.274	1430	1.41e-5	458	0.386
bcsstk27	1224	64534	3.43e-4	15000	2.937	63935	2.11e-4	15000	3.064
bcsstk28	4410	62724	2.32e-4	15000	13.300	63527	1.83e-4	15000	14.293
bcsstk36	23052	54355	2.00e-5	12861	76.108	26406	1.99e-5	6561	38.742
bcsstm12	1473	11960	1.94e-5	3017	0.295	8632	1.99e-5	2268	0.240
bcsstm23	3134	14536	1.98e-5	3670	0.500	14820	2.00e-5	3735	0.652
bcsstm24	3562	8255	2.00e-5	2228	0.312	6285	1.85e-5	1721	0.267
cfd1	70656	2693	2.00e-5	801	6.265	2201	1.83e-5	659	5.238
ex15	6867	197	7.92e-6	76	0.027	9578	1.97e-5	2606	1.334
fv1	9604	371	1.81e-5	136	0.061	474	1.85e-5	170	0.0949
fv2	9801	369	1.54e-5	138	0.066	334	1.51e-5	127	0.0637
fv3	9801	506	1.82e-5	183	0.077	371	1.52e-5	136	0.0664
Kuu	7102	5050	3.28e-6	1479	1.533	2674	4.82e-6	819	0.9281
mhd4800b	4800	1959	1.98e-5	605	0.146	1784	1.96e-5	561	0.1363
msc23052	23052	43296	1.99e-5	10410	62.815	37779	1.99e-5	9217	56.9276
Muu	7102	29	1.78e-5	13	0.006	28	1.70e-5	12	0.0052
nasa4704	4704	63663	8.21e-5	15000	7.418	63440	5.79e-5	15000	8.264
s1rmq4m1	5489	9000	2.00e-5	2411	2.394	11052	1.99e-5	2933	3.2589
s1rmt3m1	5489	23019	1.98e-5	5895	4.566	13688	1.93e-5	3592	3.1067
s2rmq4m1	5489	9839	2.00e-5	2564	2.384	10292	1.96e-5	2683	2.774
s2rmt3m1	5489	17422	2.00e-5	4437	3.504	21669	1.93e-5	5385	4.9953
s3rmq4m1	5489	6063	1.85e-5	1710	1.500	5422	1.20e-5	1517	1.4757
s3rmt3m1	5489	12250	1.76e-5	3295	2.467	6331	1.52e-5	1764	1.3742
s3rmt3m3	5357	13090	1.89e-5	3461	2.569	9911	1.83e-5	2669	2.1663
sts4098	4098	22153	1.89e-5	5583	2.335	17664	1.93e-5	4499	2.0278

we obtain the following tangent vector field defined on the Stiefel manifold

$$F(X) \equiv H(X)X - XX^\top H(X)X = 0_X, \tag{6.16}$$

where $F : St(n,p) \to \mathcal{T}St(n,p)$.

In Table 2 we report the results obtained by running the RSANE and CGPR methods on the problem of finding a zero of (6.16), with six possible choices for the pair (n,p), obtained by varying n and p in $\{100, 500, 1000\}$ and $\{10, 50\}$, respectively. For comparison purposes, we repeat our experiments over 30 different random generated

TABLE 2
Numerical results for nonlinear eigenvalue problems.

Method	Nitr	Time	NrmF	Nfe	Nitr	Time	NrmF	Nfe
	$(n,p) = (100,10)$				$(n,p) = (100,50)$			
RSANE-polar	70.0	0.022	7.85e-6	167.2	632.9	0.898	2.10e-4	2170.1
RSANE-qr	67.8	0.016	6.87e-6	161.9	620.6	0.505	4.10e-4	2126.9
CGPR-polar	79.0	0.029	8.30e-5	231.3	1209.4	2.747	1.46e-3	6586.4
CGPR-qr	78.2	0.024	8.05e-5	229.3	996.8	1.296	3.30e-4	5451.3
	$(n,p) = (500,10)$				$(n,p) = (500,50)$			
RSANE-polar	65.0	0.200	8.10e-6	151.6	311.6	2.048	7.79e-6	955.4
RSANE-qr	68.0	0.211	7.43e-6	160.0	311.5	1.928	8.79e-6	954.6
CGPR-polar	74.3	0.284	8.14e-5	209.3	737.5	7.757	8.86e-5	3601.4
CGPR-qr	73.6	0.280	8.17e-5	210.2	767.1	7.583	9.12e-5	3731.1
	$(n,p) = (1000,10)$				$(n,p) = (1000,50)$			
RSANE-polar	68.9	0.628	7.55e-6	159.8	316.3	5.272	8.12e-6	968.9
RSANE-qr	66.8	0.604	6.86e-6	153.4	321.3	5.294	8.03e-6	987.6
CGPR-polar	73.5	0.820	7.57e-5	208.5	790.3	21.224	9.36e-5	3880.0
CGPR-qr	77.4	0.860	6.87e-5	219.6	881.2	23.505	9.50e-5	4347.7

starting points $X_0 \in St(n,p)$ for each pair of (n,p) and report the averaged number of iteration (Nitr), the averaged number of the evaluation of the functional $F(\cdot)$ (Nfe), the averaged total computing time in seconds (Time) and the averaged residual (NrmF) given by $\frac{1}{30}\sum_{i=1}^{30} ||F(x_*^i)||_F$, where x_*^i denotes the estimated solution obtained by the respective algorithm for the i–th starting point, for all $i \in \{1,2,\ldots,30\}$. In addition, for all the experiments we fix $\mu = 1$, $\epsilon =$ 1e-4 as the tolerance for the termination rule based on residual norm $||F(X_k)||_F < \epsilon$. In this table, RSANE_polar and RSANE_qr denotes our Algorithm 3.1 using the retractions defined in (6.6) and (6.5), respectively. Similarly, CGPR_polar and CGPR_qr denotes the Riemannian Derivative–Free conjugate gradient Polak–Ribiére–Polyak method developed in [30] using the retractions (6.6) and (6.5), respectively.

As shown in Table 2, RSANE is superior to the Riemannian conjugate gradient method solving nonlinear eigenvalues problems for different choices of (n,p). In particular, we note that for problems with $p = 50$, our proposal basically converges in half of the iterations that the CGPR takes to reach the desired precision.

6.4. Joint diagonalization on the oblique manifold. In this subsection we analyze the numerical behaviour of our RSANE method solving the nonlinear equation based on the tangent vector field obtained from the Riemannian gradient of the scalar function $\mathcal{F} : \mathcal{OB}(n,p) \to \mathbb{R}$, defined by

$$\mathcal{F}(X) = \sum_{i=1}^{N} ||\texttt{off}(X^\top C_i X)||_F^2, \tag{6.17}$$

where $\texttt{off}(W) = W - \texttt{ddiag}(W)$ and $C_i \in \mathbb{R}^{n\times n}$ are given symmetric matrices. The minimization of this function on $\mathcal{OB}(n,p)$ is frequently used to perform independent component analysis, see [1]. It is well–known that the problem of minimizing (6.17) is closely related to the problem of finding a zero to the Riemannian gradient of the

function $\mathcal{F}(\cdot)$, which is given by

$$\nabla_{\mathcal{OB}(n,p)}\mathcal{F}(X) = P_{T_X\mathcal{OB}(n,p)}\left[\sum_{i=1}^{N} 4C_iX\,\mathtt{off}(X^\top C_iX)\right], \tag{6.18}$$

where $P_{T_X\mathcal{OB}(n,p)}[Z] = Z - X\,\mathtt{ddiag}(X^\top Z)$, for all $Z \in \mathbb{R}^{n\times p}$, for details see Section 3 in [1]. In order to study the numerical performance of our RSANE compared with the CGPR method, we consider the following tangent vector field

$$F(X) \equiv P_{T_X\mathcal{OB}(n,p)}\left[\sum_{i=1}^{N} 4C_iX\,\mathtt{off}(X^\top C_iX)\right] = 0_X, \tag{6.19}$$

which is obtained form the Riemannian gradient of $\mathcal{F}(\cdot)$.

Now, we present a computational comparison considering the RSANE and CGPR [30] methods on the solution of the tangent vector field (6.19). To do this, we set $N = 5$, and study the numerical behavior of both methods for the pairs of values $(n,p) = (500,100)$ y $(n,p) = (1000,100)$. For each pair (n,p), we repeat 30 independent runs of RSANE and CGPR, generating the symmetric matrices $C_i \in \mathbb{R}^{n\times n}$ as follows

$$C_i = D + B_i + B_i^\top,$$

where $D \in \mathbb{R}^{n\times n}$ was a diagonal matrix, whose diagonal elements are given by $d_{ii} = \sqrt{n+i}$, for all $i \in \{1,2,\ldots,n\}$; and the $B_i's \in \mathbb{R}^{n\times n}$ were randomly generated matrices, whose entries follow a standard normal distribution. This particular structure of the C_i's matrices was taken from [34]. The starting point $X_0 \in \mathcal{OB}(n,p)$ was randomly generated using the following Matlab commands

$$X_0 = \mathtt{zeros}(n,p); \quad M = \mathtt{randn}(n,p) \quad \text{and} \quad X_0(:,i) = M(:,i)/\mathtt{norm}(M(:,i)),$$

for all $i = 1,...,p$. For this comparison, we use $\epsilon =$ 1e-5 as the tolerance for the stopping rule based on the residual norm $||F(X_k)||_F < \epsilon$.

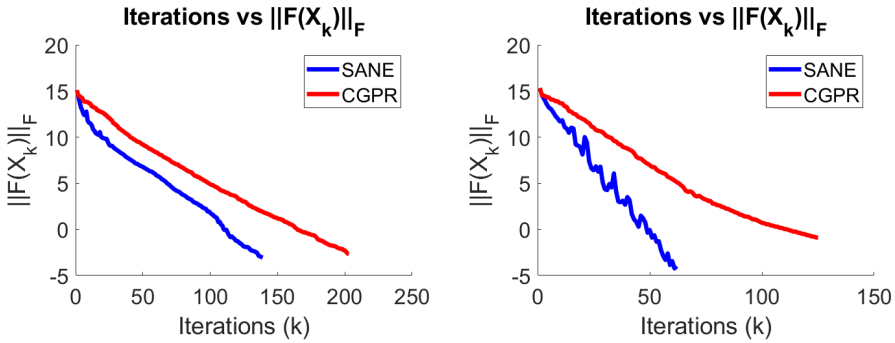
The mean of number of iteration, number of evaluation of $F(\cdot)$, total computational time in seconds, and the residual "NrmF" defined as in the previous subsection, are reported in Table 3. In addition, in Figure 2 we plot the numerical behavior associated to the average residual curve $||F(X_k)||_F$ throughout the iterations, for all the methods and for each pairs (n,p).

From Table 3 and Figure 2, we can see that both RSANE and CGPR can find an approximation of the solution of problem (6.19) with the pre–established precision for the residual norm, in the two cases of (n,p) considered. In addition, we clearly observe that RSANE performs better than CGPR in terms of the mean value of iterations and CPU–time.

7. Concluding Remarks. In this paper, a Riemannian residual approach for finding a zero of a tangent vector field is proposed. The new approach can be seen as an extended version of the SANE method developed in [7], for the solution of large–scale nonlinear systems of equations. Since the proposed method systematically uses the tangent vector field for building the search direction, RSANE is very easy to implement and has low storage requirements, which is suitable for solving large–scale

TABLE 3
Numerical results for joint diagonalization Riemannian gradient vector field.

Method	Nitr	Time	NrmF	Nfe
$(n,p) = (500, 100)$				
RSANE	157.8	3.950	6.85e-6	390.0
CGPR	238.8	7.589	7.84e-6	741.8
$(n,p) = (1000, 100)$				
RSANE	68.1	4.894	6.50e-6	142.5
CGPR	153.4	14.342	7.59e-6	416.1



(a) Iterations vs $||F(X_k)||_F$ (b) Iterations vs $||F(X_k)||_F$

FIG. 2. *Averaged convergence history of RSANE and CGPR, from the same initial point, for the joint diagonalization Riemannian gradient vector field. On the left,* $(n, p, N) = (500, 100, 5)$*, and on the right* $(n, p, N) = (1000, 100, 5)$*. The y–axis is in logarithmic scale.*

problems. In addition, our proposal uses a modification of the Riemannian Barzilai–Borwein step–sizes introduced in [12], combined with the Zhang–Hager globalization strategy [31], in order to guarantee the convergence of the associated residual sequence.

The preliminary numerical results show that RSANE performs efficiently, dealing with several tangent vector fields considering both real and simulated data, and different Riemannian manifolds. In particular, RSANE is competitive against its Euclidean version (SANE), solving large–scale eigenvalue problems. Additionally, RSANE outperforms the derivative-free conjugate gradient algorithm recently published in [30], on two considered matrix manifolds.

REFERENCES

[1] P.-A. ABSIL AND K. A. GALLIVAN, *Joint diagonalization on the oblique manifold for independent component analysis*, in 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 5, IEEE, 2006, pp. V–V.
[2] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2009.
[3] R. L. ADLER, J.-P. DEDIEU, J. Y. MARGULIES, M. MARTENS, AND M. SHUB, *Newton's method on riemannian manifolds and a geometric model for the human spine*, IMA Journal of Numerical Analysis, 22 (2002), pp. 359–390.
[4] J. BARZILAI AND J. M. BORWEIN, *Two-point step size gradient methods*, IMA journal of numerical analysis, 8 (1988), pp. 141–148.
[5] P. BREIDING AND N. VANNIEUWENHOVEN, *Convergence analysis of riemannian gauss–newton*

methods and its connection with the geometric condition number, Applied Mathematics Letters, 78 (2018), pp. 42–50.
[6] O. S. D. Cedeno and H. F. O. Leon, *Projected nonmonotone search methods for optimization with orthogonality constraints*, Computational and Applied Mathematics, 37 (2018), pp. 3118–3144.
[7] W. L. Cruz and M. Raydan, *Nonmonotone spectral methods for large-scale nonlinear systems*, Optimization Methods and Software, 18 (2003), pp. 583–599.
[8] T. A. Davis and Y. Hu, *The university of florida sparse matrix collection*, ACM Transactions on Mathematical Software (TOMS), 38 (2011), p. 1.
[9] J.-P. Dedieu, P. Priouret, and G. Malajovich, *Newton's method on riemannian manifolds: covariant alpha theory*, IMA Journal of Numerical Analysis, 23 (2003), pp. 395–419.
[10] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM journal on Matrix Analysis and Applications, 20 (1998), pp. 303–353.
[11] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for newton's method*, SIAM Journal on Numerical Analysis, 23 (1986), pp. 707–716.
[12] B. Iannazzo and M. Porcelli, *The riemannian barzilai–borwein method with nonmonotone line search and the matrix geometric mean computation*, IMA Journal of Numerical Analysis, 38 (2018), pp. 495–517.
[13] E. Kokiopoulou, J. Chen, and Y. Saad, *Trace optimization and eigenproblems in dimension reduction methods*, Numerical Linear Algebra with Applications, 18 (2011), pp. 565–602.
[14] C. Li and J. Wang, *Convergence of the newton method and uniqueness of zeros of vector fields on riemannian manifolds*, Science in China Series A: Mathematics, 48 (2005), pp. 1465–1478.
[15] J. H. Manton, *Optimization algorithms exploiting unitary constraints*, IEEE Transactions on Signal Processing, 50 (2002), pp. 635–650.
[16] R. M. Martin, *Electronic structure: basic theory and practical methods*, Cambridge university press, 2020.
[17] E. Oja, *Neural networks, principal components, and subspaces*, International journal of neural systems, 1 (1989), pp. 61–68.
[18] H. Oviedo, *Implicit steepest descent algorithm for optimization with orthogonality constraints*.
[19] H. Oviedo and O. Dalmau, *A scaled gradient projection method for minimization over the stiefel manifold*, in Mexican International Conference on Artificial Intelligence, Springer, 2019, pp. 239–250.
[20] H. Oviedo, O. Dalmau, and H. Lara, *Two adaptive scaled gradient projection methods for stiefel manifold constrained optimization*, Numerical Algorithms, (2020), pp. 1–21.
[21] H. Oviedo, H. Lara, and O. Dalmau, *A non-monotone linear search algorithm with mixed direction on stiefel manifold*, Optimization Methods and Software, 34 (2019), pp. 437–457.
[22] M. Raydan, *On the barzilai and borwein choice of steplength for the gradient method*, IMA Journal of Numerical Analysis, 13 (1993), pp. 321–326.
[23] M. Raydan, *The barzilai and borwein gradient method for the large scale unconstrained minimization problem*, SIAM Journal on Optimization, 7 (1997), pp. 26–33.
[24] W. Ring and B. Wirth, *Optimization methods on riemannian manifolds and their application to shape space*, SIAM Journal on Optimization, 22 (2012), pp. 596–627.
[25] Y. Saad, J. R. Chelikowsky, and S. M. Shontz, *Numerical methods for electronic structure calculations of materials*, SIAM review, 52 (2010), pp. 3–54.
[26] H. Sato, *Riemannian newton's method for joint diagonalization on the stiefel manifold with application to ica*, arXiv preprint arXiv:1403.8064, (2014).
[27] H. Sato and T. Iwai, *A new, globally convergent riemannian conjugate gradient method*, Optimization, 64 (2015), pp. 1011–1031.
[28] P. Turaga, A. Veeraraghavan, and R. Chellappa, *Statistical analysis on stiefel and grassmann manifolds with applications in computer vision*, in 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[29] Z. Wen, C. Yang, X. Liu, and Y. Zhang, *Trace-penalty minimization for large-scale eigenspace computation*, Journal of Scientific Computing, 66 (2016), pp. 1175–1203.
[30] T.-T. Yao, Z. Zhao, Z.-J. Bai, and X.-Q. Jin, *A riemannian derivative-free polak–ribiére–polyak method for tangent vector field*, Numerical Algorithms, (2020), pp. 1–31.
[31] H. Zhang and W. W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM journal on Optimization, 14 (2004), pp. 1043–1056.
[32] T. Zhang and Y. Yang, *Robust pca by manifold optimization*, The Journal of Machine Learning Research, 19 (2018), pp. 3101–3139.
[33] X. Zhang, J. Zhu, Z. Wen, and A. Zhou, *Gradient type optimization methods for electronic structure calculations*, SIAM Journal on Scientific Computing, 36 (2014), pp. C265–C289.

[34] X. Zhu, *A riemannian conjugate gradient method for optimization on the stiefel manifold*, Computational optimization and Applications, 67 (2017), pp. 73–110.