A Riemannian Conjugate Gradient Algorithm with Implicit Vector Transport for Optimization on the Stiefel Manifold

H. Oviedo\textsuperscript{a} and H. Lara\textsuperscript{b}
\textsuperscript{a}Mathematics Research Center, CIMAT A.C. Guanajuato, Mexico; \textsuperscript{b}Universidade Federal de Santa Catarina. Campus Blumenau, Brazil

ARTICLE HISTORY
Compiled February 24, 2018

ABSTRACT
In this paper, a reliable curvilinear search algorithm for solving optimization problems over the Stiefel manifold is presented. This method is inspired by the conjugate gradient method, with the purpose of obtain a new direction search that guarantees descent of the objective function in each iteration. The merit of this algorithm lies in the fact that is not necessary extra calculations associated to vector transport. To guarantee the feasibility of each iteration, a retraction based on the QR factorization is considered. In addition, this algorithm enjoys global convergence. Finally, two numerical experiments are given to confirm the effectiveness and efficiency of presented method with respect to some other state of the art algorithms.

KEYWORDS
optimization on manifolds, Stiefel manifold, conjugate gradient method, vector transport.

AMS CLASSIFICATION
49K99; 49Q99; 49M30; 49M37; 90C30; 68W01

1. Introduction

In this paper, we consider the following manifold-constrained optimization problem

$$\min_{X} \mathcal{F}(X) \quad s.t. \quad X \in St(n, p),$$

where $\mathcal{F} : \mathbb{R}^{n \times p} \to \mathbb{R}$ is a continuously differentiable real-valued function and $St(n, p) = \{X \in \mathbb{R}^{n \times p} : X^\top X = I\}$ is known as Stiefel manifold, where $I$ denote the identity matrix.

Nowadays, manifold-constrained optimization is an active area of research due to the wide range of applications that this involves, for example, in machine learning [1], recommendation systems (matrix completion problem) [2–4], electronic structure [5], joint diagonalization [6], multi-view clustering [7], maxcut problems [8]. For the specific case of the Stiefel manifold, the problem (1) arises in applications...
such as, 1-bit compressed sensing [9], linear and nonlinear eigenvalue problems [5,10,11], orthogonal procrustes problems and weighted orthogonal procrustes [12–14], image segmentation [15], sensor localization [16], manifold learning [17], p-harmonic flow [18], among others. In addition, many techniques for reducing the dimensionality of data in the area of pattern recognition can be formulated as the problem (1) [19].

It have been developed works that address the general optimization problem on Riemannian manifold since the decade of the 70’s [20]. However, the first purpose algorithms specially designed to deal with the problem (1) appeared in the 1990’s, [21–23]. These seminal works exploit the geometry of the feasible set $St(n,p)$, and the underlined algorithms perform a serie of descent steps along a geodesic (the curve of shortest length between two points on a manifold). Such search is intractable in practice. In 2002, Manton [24] presents two algorithms that break with the paradigm of descending along a geodesic. Since that date, it has emerged different pragmatic algorithms that approximate descent search along other smooth curves on the manifold. These curves are defined by mappings that convert any displacement in the tangent space to a point on the manifold, these smooth mappings are called retractions. Some gradient-type methods based on retractions to solve the problem (1) have been proposed in [25–30]. Other first-order algorithms such as the Riemannian conjugate gradient method (RCG) and Quasi-Newton method are the subject in [23,27,31–33]. Furthermore some Newton’s type algorithms, which use second-order information, appear in [23,27,34].

All the conjugated gradients algorithms introduced in [23,27,31,32] need to use some “vector transport” (the definition of this concept is in section 2), which entails a greater computational effort associated to a projection onto the manifold. This paper introduces a new RCG algorithm based on retraction that avoids extra calculations related to vector transport, which emerge implicitly from the previous computations.

In this article, an efficient monotone algorithm of linear search on manifold is proposed, which is inspired by the standard conjugate gradient method, in order to obtain a novel version of the Riemannian Conjugate Gradient Method that does not require to use any vector transport or parallel transport explicitly. In this way, we designed a feasible Conjugated Gradient Method to address the problem (1) and we also provide a new vector transport for the Stiefel manifold. This algorithm preserve the feasibility of each iterate by using QR re-orthogonalization procedure. Therefore, we get a descent algorithm SVD-free and also free of matrix inversion, unlike many other methods of the state of the art.

The remainder of this paper is organized as follows. In the next section, we introduce some notations and definitions about Riemannian geometry, which may not be familiar to some readers, furthermore, in subsection 2.1 we provide a self-contained introduction to geometry of the Stiefel manifold and we introduce a new vector transport for the Stiefel manifold. In section 3, we review the Riemannian conjugate gradient methods, which are described in [21,27]. In section 4, we describe a new feasible update scheme to address the problem (1) and propose our algorithm. Section 5 is dedicated to present some numerical results. Finally, the conclusions and discussions are commented in the last section.
2. Notation and background

In the remainder of this work, we will use the following concepts and notations. Let $A$ be an $n$-by-$n$ matrix with real entries. We say that $A$ is skew-symmetric if $A^\top = -A$. The trace of $A$, denoted by $\text{Tr}[A]$, is defined as the sum of the diagonal entries. The Euclidean inner product of two matrices $A, B \in \mathbb{R}^{m \times n}$ is defined as $\langle A, B \rangle := \sum_{i,j} a_{ij} b_{ij} = \text{Tr}[A^\top B]$, where $a_{ij}$ and $b_{ij}$ denote the elements $(i,j)$ of the matrices $A$ and $B$ respectively. Furthermore, the canonical inner product associated with a matrix $X \in \mathbb{R}^{m \times n}$ is defined as $\langle A, B \rangle_c := \text{Tr}[A^\top (I - \frac{1}{2}XX^\top) B]$.

The Frobenius norm is defined as the metric induced by the Euclidean inner product, that is, $||A||_F = \sqrt{\langle A, A \rangle_e}$. Let $\mathbf{F} : \mathbb{R}^{n \times p} \to \mathbb{R}$ be a differentiable function, and denote by $G := \mathbf{D}\mathbf{F}(X) := \left( \frac{\partial \mathbf{F}(X)}{\partial x_{ij}} \right)$ the matrix of partial derivatives of $\mathbf{F}$ with respect to $X$ (that is, the Euclidean gradient of $\mathbf{F}$). The directional derivative of $\mathbf{F}$ along a given matrix $Z \in \mathbb{R}^{n \times p}$ at a given point $X$ is defined by

$$
\mathbf{D}\mathbf{F}(X)[Z] := \lim_{t \to 0} \frac{\mathbf{F}(X + tZ) - \mathbf{F}(X)}{t} = \langle G, Z \rangle_e.
$$

A Riemannian manifold $\mathcal{M}$ is a manifold whose tangent spaces $T_x\mathcal{M}$ at a given $x \in \mathcal{M}$ are endowed with a smooth local inner product $g(\eta_x, \xi_x) = \langle \eta_x, \xi_x \rangle_x$, where $\eta_x, \xi_x \in T_x\mathcal{M}$. This smoothly varying inner product is called the Riemannian metric. Let $f : \mathcal{M} \to \mathbb{R}$ be a differentiable scalar field on a Riemannian manifold $\mathcal{M}$. The Riemannian gradient of $f$ at $x$, denoted by $\nabla f(x)$, is defined as the unique element of $T_x\mathcal{M}$ that satisfies

$$
\langle \nabla f(x), \xi \rangle_x = \mathbf{D}f(x)[\xi], \quad \forall \xi \in T_x\mathcal{M}.
$$

Now, let $f : E \to \mathbb{R}$ be a differentiable objective function that we want to minimize on a Riemannian submanifold $\mathcal{M}$ of a Euclidean space $E$, and let $\nabla f(x)$ be the Euclidean gradient of $f$ at $x \in E$. Then the Riemannian gradient of $f$ at $x \in \mathcal{M}$ is equal to the orthogonal projection of the Euclidean gradient $\nabla f(x)$ onto $T_x\mathcal{M}$, that is

$$
\nabla f(x) = P_{T_x\mathcal{M}}[\nabla f(x)],
$$

where $P_{T_x\mathcal{M}}[\cdot]$ denote the orthogonal projection onto $T_x\mathcal{M}$.

Other concepts of interest are retraction and vector transport. A retraction is a smooth mapping that transforms any displacement in the tangent space to a point on the manifold that satisfies some technical conditions. Below we present a rigorous definition of a retraction.

**Definition 4.1.1 in [27]**. A retraction on a manifold $\mathcal{M}$ is a smooth mapping $R$ from the tangent bundle $T\mathcal{M} := \bigcup_{x \in \mathcal{M}} T_x\mathcal{M}$ onto $\mathcal{M}$ with the following properties. Let $R_x$ denote the restriction of $R$ 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, $T_0 T_x M \simeq T_x M$, $R_x$ satisfies
\[ DR_x(0_x) = \text{id}_{T_x M}, \]
where $\text{id}_{T_x M}$ denotes the identity mapping on $T_x M$.

This concept is widely used in optimization methods on manifold, both those of the first order (Line Search Methods, Steepest Descent, Conjugated Gradient, Quasi-Newton, among others) as well as those of second order (for example Newton method). The choice of an appropriate retraction when designing an optimization method is essential to obtain an efficient algorithm.

When developing algorithms on manifolds based on retractions, operations involving vectors in different tangent spaces can appear. To overcome this technical difficulty, the concept of vector transport have been used. This concept is fundamental to establish the Riemannian Conjugate Gradient methods.

Definition 2.2 (Definition 8.1.1 in [27]). A vector transport $T$ on a manifold $M$ is a smooth mapping
\[ TM \oplus TM \to TM : (\eta, \xi) \mapsto T_\eta(\xi) \in TM, \]
satisfying the following properties for all $x \in M$ where $\oplus$ denote the Whitney sum, that is,
\[ TM \oplus TM = \{ (\eta, \xi) : \eta, \xi \in T_x M, x \in M \}. \]

(1) There exists a retraction $R$, called the retraction associated with $T$, such that
\[ \pi(T_\eta(\xi)) = R_x(\eta), \quad \eta, \xi \in T_x M, \]
where $\pi(T_\eta(\xi))$ denotes the foot of the tangent vector $T_\eta(\xi)$.

(2) $T_{0_x}(\xi) = \xi$ for all $\xi \in T_x M$.

(3) $T_\eta(a\xi + b\zeta) = aT_\eta(\xi) + bT_\eta(\zeta)$, for all $a, b \in \mathbb{R}$ and $\eta, \xi, \zeta \in T_x M$.

2.1. The geometry of the Stiefel manifold and the gradient of the objective function

In this subsection, we review the geometry of the Stiefel manifold $St(n, p)$, as discussed in [23,27].

It is well known that the Stiefel manifold $St(n, p)$ is an embedded submanifold of $\mathbb{R}^{n \times p}$. The tangent space of $St(n, p)$ at $X \in St(n, p)$ is
\[ T_X St(n, p) = \{ Z \in \mathbb{R}^{n \times p} : Z^\top X + X^\top Z = 0 \}. \]

The following proposition provides us with an alternative characterization of $T_X St(n, p)$.
Proposition 2.3. Let $X$ be a matrix in $St(n,p)$ and $\Omega_X$ the set defined by,

$$\Omega_X = \{ Z \in \mathbb{R}^{n \times p} : Z = WX, \text{ for some skew-symmetric matrix } W \in \mathbb{R}^{p \times p} \}.$$ 

Then $T_X St(n,p) = \Omega_X$.

**Proof.** Let $Z \in T_X St(n,p)$ be an arbitrary matrix, and define by

$$W = P_X Z - X Z^\top P_X,$$

where $P_X = I - \frac{1}{2}XX^\top$. Clearly, $W$ is a skew-symmetric matrix. Using the fact that $Z^\top X = -X^\top Z$ and $X \in St(n,p)$ we have,

$$WZ = Z - \frac{1}{2}XX^\top Z - \frac{1}{2}XZ^\top X = Z,$$

therefore, we conclude that $T_X St(n,p) \subset \Omega_X$.

On the other hand, consider $Z \in \Omega_X$, then there exists a skew-symmetric matrix $W \in \mathbb{R}^{n \times n}$ such that $Z = WX$. So, note that,

$$Z^\top X + X^\top Z = (WX)^\top X + X^\top WX = -X^\top WX + X^\top WX = 0,$$

thus, we have that $\Omega_X \subset T_X St(n,p)$, which completes the proof.

If we endorse the Stiefel manifold with the Riemannian metric $\langle \cdot, \cdot \rangle_e$ inherited from the embedding space $\mathbb{R}^{n \times p}$, then the normal space to $T_X St(n,p)$ is

$$\left( T_X St(n,p) \right) ^\perp = \{ XS : S^\top = S, S \in \mathbb{R}^{p \times p} \}.$$ 

The orthogonal projection onto $T_X St(n,p)$ is

$$P_{T_X St(n,p)}[\xi] = (I - XX^\top)\xi + X\text{skew}(X^\top \xi),$$

(2)

where $\text{skew}(W) := \frac{1}{2}(W - W^\top)$ denote the skew-symmetric part of the square matrix $W$. The Riemannian gradient of objective function $\mathcal{F}$ of the equation (1) at $X$ under the Euclidean inner product $\langle \cdot, \cdot \rangle_e$ is

$$\nabla_e \mathcal{F}(X) := \text{grad}\mathcal{F}(X) = P_{T_X St(n,p)}[G] = (I - XX^\top)G + X\text{skew}(X^\top G),$$

where $G$ is the Euclidean gradient of $\mathcal{F}$ at $X$. And the Riemannian gradient of $\mathcal{F}$ at $X$ under canonical inner product $\langle \cdot, \cdot \rangle_c$ is

$$\nabla_c \mathcal{F}(X) := G - XG^\top X,$$

for more details about this two Riemannian gradient see [25,27]. Observe that $\nabla_c \mathcal{F}(X)$ can be rewritten as
\[ \nabla_c F(X) := A(X)X, \]

where \( A(X) = GX^\top - XG^\top \). Clearly, \( \nabla_c F(X) \) and \( \nabla_c F(X) \) belong to the tangent space \( T_XSt(n,p) \). Furthermore, it is follows from proposition 2.3 that \( \eta = WX \) also belongs to \( T_XSt(n,p) \). In section 4 we will exploit this result in combination with the Riemannian gradient \( \nabla_c F(X) \), using the function \( A(X) \), to design a new Riemannian conjugate gradient method for the Stiefel manifold.

3. The Riemannian conjugate gradient methods

In this section, we give a brief review of the Riemannian conjugate gradient methods, a complete description of these methods appear in [23,27]. Manifold-constrained optimization refers to a class of problems of the form,

\[ \min f(x) \quad \text{s.t.} \quad x \in M, \]

where \( f : M \to \mathbb{R} \) is a given smooth real-valued function, and \( M \) is a Riemannian manifold. A general iteration of a Riemannian conjugate gradient method update the iterated using the following scheme starting in \( x_0 \in M \) with \( \eta_0 = -\nabla f(x_0) \),

\[ x_{k+1} = R_{x_k}(\tau_k \eta_k), \]

where \( R_{x_k} : T_{x_k}M \to M \) is a retraction, \( \tau_k > 0 \) is the step-size and \( \eta_k \in T_{x_k}M \) is given by the following recursive formula,

\[ \eta_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1}T_{\tau_k \eta_k}(\eta_k), \quad (3) \]

where \( T \) is a vector transport and \( \nabla f(\cdot) \) is the Riemannian gradient. There are several expressions to update the parameter \( \beta_{k+1} \) of the equation (3), some of the most popular are, the \( \beta \) of Fletcher-Reeves

\[ \beta_{k+1}^{FR} = \frac{\langle \nabla f(x_{k+1}), \nabla f(x_{k+1}) \rangle_{x_{k+1}}}{\langle \nabla f(x_k), \nabla f(x_k) \rangle_{x_k}}, \]

and the \( \beta \) of Polak-Ribière,

\[ \beta_{k+1}^{PR} = \frac{\langle \nabla f(x_{k+1}), \nabla f(x_{k+1}) - T_{\tau_k \eta_k}(\nabla f(x_k)) \rangle_{x_{k+1}}}{\langle \nabla f(x_k), \nabla f(x_k) \rangle_{x_k}}. \]

Some Riemannian conjugate gradient algorithms to solve the problem (1) have been proposed in [23,27,31,32]. As we can see, this method requires the use of vector transport to update the direction search \( \eta_k \) and in some cases, it is also necessary for the calculation of \( \beta \). If \( M \) is an embedded submanifold of a Euclidean space \( \mathcal{E} \), with an associated retraction \( R_x(\cdot) \), then we have the inclusion \( T_yM \subset \mathcal{E} \) for all \( y \in M \),
and so, we can define the vector transport by
\[ T_{\eta_x}(\xi_x) = P_{R_x(\eta_x)}(\xi_x), \]
where \( P_x \) denotes the orthogonal projector onto \( T_xM \). For the case when \( M \) is the Stiefel manifold, this vector transport uses the orthogonal projector defined in (2), and makes the Riemannian conjugate gradient algorithm perform more matrix multiplications per iteration than the Riemannian steepest descent. In [32], Zhu introduce two novel vector transports associated with the Cayley transform retraction for Stiefel-constrained optimization. However its two vector transports require to invert a matrix, which can be computationally expensive. In the next section, we present a novel Riemannian conjugate gradient algorithm, which does not require to calculate a vector transport.

4. A novel Riemannian Conjugate Gradient Algorithm

In this section, we formulate a concrete algorithm to address optimization problems on the Stiefel manifold with general objective functions. Specifically, we introduce an update formula that preserves the restrictions using a retraction map based on QR re-orthogonalization. The proposal consists of a novel Riemannian Conjugate Gradient Method (RCG), which does not need to use parallel transport or vector transport explicitly, to update the direction search. This novelty is what differentiates our proposal from the standard RCG algorithm discussed in section 3.

Let \( X_k \) be the point obtained in the \( k \)-th iteration, then we compute the new iterate \( Y_k(\tau) \) as a point on the curve,
\[ Y_k(\tau) = Qf(X_k + \tau Z_k), \]
where \( \tau > 0 \) is the step-size, \( Z_k = W_k X_k \), and \( Qf(\cdot) \) denote the mapping that sends a matrix to the \( Q \) factor of its QR decomposition such that the \( R \) factor is an upper triangular \( p \times p \) matrix with strictly positive diagonal elements. It is well known that the \( Qf(\cdot) \) is a retraction on the Stiefel manifold. Our contribution arises in the definition of the matrix \( W_k \), which is given by the following recursive formula starting in \( W_0 = -A_0 \),
\[ W_k = -A_k + \beta_k W_{k-1}, \]
where \( A_k := A(X_k) = G_kX_k^\top - X_kG_k^\top \), this operator was defined at the end of section 2 and this is related to the Riemannian gradient under the canonical inner product, and \( \beta_k \) is a scalar that can be selected in several ways. For example, this parameter can be chosen in such a way that descent is guaranteed, that is,
\[ \beta_k = \begin{cases} \geq 0 & \text{if } D\mathcal{F}(X_k)[W_{k-1}X_k] \leq 0 \\ < \frac{1}{2} \frac{||A_k||_2}{D\mathcal{F}(X_k)[W_{k-1}X_k]} & \text{in other case.} \end{cases} \]
However, in this paper we will use the parameters $\beta_k$ given by the well-known formulas of Fletcher-Reeves and Polak-Ribiére [35],

$$\beta^\text{FR}_k = \frac{||A_k||_F^2}{||A_{k-1}||_F^2} \quad \text{and} \quad \beta^\text{PR}_k = \frac{Tr[A_k^T(A_k - A_{k-1})]}{||A_{k-1}||_F^2}.$$ 

In particular, we adopt a hybrid strategy, which is very used in the case of non-linear unconstrained optimization, that updates $\beta_k$ as follow,

$$\beta^\text{FR-PR}_k = \begin{cases} -\beta^\text{PR}_k & \text{if } \beta^\text{PR}_k < -\beta^\text{FR}_k \\ \beta^\text{PR}_k & \text{if } |\beta^\text{PR}_k| < \beta^\text{FR}_k \\ \beta^\text{FR}_k & \text{if } \beta^\text{PR}_k > \beta^\text{FR}_k, \end{cases}$$

which has performed well on some applications.

Now, observe that $W_k$ is a skew-symmetric matrix because this matrix is obtained as the sum of two skew-symmetric matrices. This result implies that the proposed direction search $Z_k = W_kX_k$ belongs to the tangent space of the Stiefel manifold at $X_k$. On the other hand, by taking the inner product of $Z_k$ with the gradient matrix $G_k$, using (5) and trace properties, we obtain

$$DF(X_k)[Z_k] = Tr[G_k^TW_kX_k] = -Tr[G_k^TA_kX_k] + \beta_kTr[G_k^TW_{k-1}X_k] = -\frac{1}{2}||A_k||_F^2 + \beta_kTr[G_k^TW_{k-1}X_k].$$

Then we have from (6) that $Z_k$ may not be to descent direction at $X_k$, because the second term in (6) may dominate the first term and in this case, we obtain $DF(X_k)[Z_k] > 0$. Nevertheless, this issue can be solved by resetting the direction search, that is, selecting $\beta_k$ as follows,

$$\beta_k = \begin{cases} \beta^\text{FR-PR}_k & \text{if } DF(X_k)[Z_k] < 0 \\ 0 & \text{in other case.} \end{cases}$$

Therefore, with this selection of the parameter $\beta_k$ we obtain an line search method on matrix manifold that uses a search direction that satisfies the descent condition and that is also in the tangent space of the manifold.

Now we introduce a new vector transport for the Stiefel manifold. Consider $X \in St(n, p)$, and $\xi_X, \eta_X \in T_X St(n, p)$, since $\Omega_X = T_X St(n, p)$, then there exists skew-symmetric matrices $W, \hat{W} \in \mathbb{R}^{n \times n}$ such that, $\xi_X = WX$ and $\eta_X = \hat{W}X$. Given a retraction $R_X(\cdot)$ on the Stiefel manifold, we define our vector transport as,

$$T_{\eta_X}(\xi_X) := WR_X(\eta_X).$$
Proposition 4.1. Let \( R_X(\cdot) : TSt(n,p) \to St(n,p) \) be an arbitrary retraction defined for the Stiefel manifold. Then the mapping defined in (8) is a vector transport for \( St(n,p) \).

**Proof.** Let \( \xi_X = WX \) and \( \eta_X = \hat{W}X \) be two tangent vectors of the Stiefel manifold at \( X \), with \( X \in St(n,p) \). Observe that from the proposition 2.3 we have \( T_{\eta_X}(\xi_X) \in T_{R_X(\eta_X)}St(n,p) \subset TSt(n,p) \), thus \( \pi(T_{\eta_X}(\xi_X)) = R_X(\eta_X) \), for all \( \eta_X, \xi_X \in T_X St(n,p) \), where \( \pi(T_{\eta_X}(\xi_X)) \) denotes the foot of the tangent vector defined in (8). On the other hand, using (8) and the fact that \( R_X(\cdot) \) is a retraction we arrive to,

\[
T_{0_X}(\xi_X) = WR_X(0_X) = WX = \xi_X, \quad \forall \xi_X = WX \in T_X St(n,p).
\]

In addition, let \( \xi_X = WX, \eta_X = \hat{W}X \) and \( \zeta_X = WX \) be three vectors belonging to \( T_X St(n,p) \), where the matrices \( W, \hat{W}, \bar{W} \) are skew-symmetric, \( X \in St(n,p) \), and let \( a, b \in \mathbb{R} \) two arbitrary scalars. It is follows from the fact that the set of skew-symmetric matrices is a linear vector space, that

\[
T_{\eta_X}(a\xi_X + b\zeta_X) = T_{\eta_X}((aW + b\hat{W})X) = (aW + b\hat{W})R_X(\eta_X) = aWR_X(\eta_X) + b\hat{WR}_X(\eta_X) = aT_{\eta_X}(\xi_X) + bT_{\eta_X}(\zeta_X),
\]

which implies that the mapping defined in is linear. Therefore, we conclude that the mapping (8) satisfies the three conditions of the definition 2.2, i.e. this mapping is a vector transport for the Stiefel manifold.

\[\square\]

Observe that the calculation of our vector transport (8), is much simpler than those existing in the literature. This only requires compute a matrix multiplication, hence this vector transport can generate more efficient Riemannian conjugate gradient algorithms. Using this particular transport vector, we prove below that the direction search given by \( Z_k \) is indeed a Riemannian conjugate gradient direction. This connects our proposal with the standard Riemannian conjugate methods presented in section 3. In fact,

\[
Z_{k+1} = W_{k+1}X_{k+1} = (-A_{k+1} + \beta_{k+1}W_k)X_{k+1} = -A_{k+1}X_{k+1} + \beta_{k+1}W_kX_{k+1} = -\nabla_cF(X_{k+1}) + \beta_{k+1}W_kX_{k+1} = -\nabla_cF(X_{k+1}) + \beta_{k+1}T_{x_k}Z_k(Z_k), \quad \text{(using (8))}
\]

therefore, the direction search that we propose, corresponds to a Riemannian conjugate gradient direction, for which, the calculation of the vector transport is free. That is, in the update of the direction search \( Z_{k+1} \) of our conjugate gradient algorithm, the vector transport of \( Z_k \) to the tangent space of the Stiefel manifold at \( X_{k+1} \) is implicit.
This allows us to save the calculations associated to the projection in the usual vector transport.

4.1. The step-size selection.

In this subsection, we focus on discussing the strategy for the selection of the step size $\tau$ in the equation (4) and we also present the proposed algorithm.

Typically, monotone line search algorithms construct a sequence $\{X_k\}$ such that the sequence $\{F(X_k)\}$ is monotone decreasing. Generally these methods calculate the step size $\tau$ as the solution of the following optimization problem,

$$\min_{\tau > 0} \phi(\tau) = F(Y_k(\tau)), \quad (10)$$

In most cases the optimization problem (10) does not have a closed solution, which makes it necessary to estimate the step size satisfying conditions that relax this optimization problem, but which in turn guarantees descent of the objective function. One of the most popular descent conditions is the so-called condition of sufficient descent or also called the Armijo rule [35]. Using this condition, the step-size of the iteration $k$-th $\tau_k$ is chosen as the largest positive real number that satisfies,

$$F(Y_k(\tau_k)) \leq F(X_k) + \rho \tau_k D F(X_k)[Z_k], \quad (11)$$

where $0 < \rho < 1$. For our algorithm, we will use the Armijo rule combined with the classic backtracking heuristic [35] to estimate the step size. Taking in mind all these considerations, we arrive to our algorithm.

**Algorithm 1** Quasi Conjugated Gradient Method with Armijo’s Inexact Line Search

**Require:** $X_0 \in St(n,p)$, $\rho, \epsilon, \delta \in (0, 1)$, $G_0 = DF(X_0)$ $A_0 = G_0 X_0^T - X_0 G_0^T$, $W_0 = -A_0$, $Z_0 = W_0 X_0$, $k = 0$.

**Ensure:** $X^*$ an $\epsilon$-stationary point.

1: while $||A_k||_F > \epsilon$ do
2: \hspace{1em} Step size selection: take an initial step-size $\tau = \mu_0$ where $\mu_0 > 0$.
3: \hspace{1em} while $F(Y_k(\tau)) > F(X_k) + \rho \tau D F(X_k)[Z_k]$ do
4: \hspace{2em} $\tau = \delta \tau$,
5: \hspace{1em} end while
6: \hspace{1em} $X_{k+1} = Y_k(\tau) := Qf(X_k + \tau Z_k)$,
7: \hspace{1em} Calculate, $G_{k+1} = DF(X_{k+1})$ and $A_{k+1} = G_{k+1} X_{k+1}^T - X_{k+1} G_{k+1}^T$,
8: \hspace{1em} Update $W_{k+1} = -A_{k+1} + \beta_{k+1} W_k$ with $\beta_{k+1}$ as in (25),
9: \hspace{1em} $Z_{k+1} = W_{k+1} X_{k+1}$,
10: \hspace{1em} $k = k + 1$,
11: end while
12: $X^* = X_k$.

The bottleneck of the previous algorithm is found in step 3, since in this step, several QR factorization must be calculated which can require a lot of computation, if it is calculated inefficiently, for example using Gram-Schmidt process. With the purpose
of reducing the computational cost of algorithm 1, we consider to compute the QR factorization of the matrix $M = X_k + \tau Z_k$ via Cholesky factorization. For this end, we use the following procedure,

**Algorithm 2 QR factorization via Cholesky**

**Require:** $M \in \mathbb{R}^{n \times p}$ a matrix such that $M^\top M$ is positive definite.

**Ensure:** $Q \in \text{St}(n, p)$ and $R \in \mathbb{R}^{p \times p}$ upper-triangular matrix.

1. Compute the Cholesky factorization $L^\top L$ of $M^\top M$ where $L \in \mathbb{R}^{p \times p}$ is a upper-triangular matrix.
2. $Q = ML^{-1}$ and $R = L$

Note that, since $W_k$ is skew-symmetric and $X_k \in \text{St}(n, p)$, we have $M^\top M = I_p + \tau^2 X_k^\top W_k W_k X_k$, so, let $v \in \mathbb{R}^p$ be an arbitrary non-zero vector, then $v^\top M_k^\top M_k v = ||v||^2 + \tau^2 ||W_k X_k v||^2 > 0$, thus we conclude that $M^\top M$ is a positive defined matrix. Therefore, the existence of the Cholesky factorization of such matrix is guaranteed.

Typically, the Riemannian conjugate gradient methods require that the step size $\tau$ satisfy the strong Wolfe conditions [35], (whose Riemannian version appear in [32,36]) in each iteration, and furthermore, the vector transport needs to satisfy the non-expansive Ring-Wirth condition [36], to guarantee global convergence. By means of a direct calculation, it can be verified that the vector transport given in (8) satisfies this condition for the case of $n = p$, nevertheless, when $p < n$, this condition is not necessarily fulfilled. There are ways to rescale any vector transport to obtain another one that satisfies the Ring-Wirth condition [37].

However, if we allow the algorithm to periodically restart the direction by setting $\beta_k = 0$, then we can guarantee that all these $Z_k$ are descent directions, and then the procedure becomes a line search method on matrix manifold. Observe that in our algorithm, we use this restart and in this way, we have that $\{Z_k\}$ gradient-related sequence that is contained in the bundle of the Stiefel manifold. In addition, the mapping $R_{X_k}(\eta_k) := Qf(X_k + \eta_k)$ is a retraction, a proof of this fact is found in [27]. Therefore, the results of global convergence that appear in [27] regarding line-search methods on manifolds using retractions apply directly to Algorithm 1, which allows us to establish the following theorem.

**Theorem 4.2.** Let $\{X_k\}$ be an infinite sequence of iterates generated by Algorithm 1, then

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

Another important observation, is that our proposal also works with other retractions different to that based on QR decomposition. For example, we can use our proposal in combination with the Cayley transform to obtain the following update formula,

$$X_{k+1} = Y_k^\text{Cayley}(\tau) := (I_n + \frac{\tau}{2} W_k)^{-1} (I_n - \frac{\tau}{2} W_k) X_k,$$  

this update scheme (12) would correspond to a modified version of the algorithm
proposed in [25]. We can also consider retractions based on exponential mapping and polar decomposition in conjunction with our proposal, that is,

\[ X_{k+1} = Y_k^{Exp}(\tau) := \exp(\tau W_k)X_k, \]  
(13)

\[ X_{k+1} = Y_k^{Polar}(\tau) := (X_k + \tau W_k X_k)(I_p - X_k^T W_k^2 X_k)^{-1/2}, \]  
(14)

respectively. Nevertheless, note that these update schemes (12)-(13)-(14) require to compute matrix inversions of size \( n \times n \), calculate exponentials matrix or estimate matrix square roots, which become expensive calculations when the dimension of the matrix \( X_k \) is large. For this reason, we adopt the retraction based on the QR decomposition, because this can be calculated efficiently using the Algorithm 2.

5. Numerical Experiments

In this section, we perform some numerical experiments to investigate the efficiency of the proposed method. All algorithms were implemented using MATLAB 7.10 on an Intel(R) CORE(TM) i7-4770, CPU 3.40 GHz with 500 Gb of HD and 16 Gb of Ram. We present a comparative study of our algorithm versus other state-of-the-art algorithms for several instances of the problems Joint diagonalization problem and Total energy minimization generating synthetic data.

5.1. Terminating the Algorithm

The algorithms are terminated if we reach a point \( X_k \) on the Stiefel manifold that satisfies any of the following stopping criteria,

- \( k \geq K_{max} \) where \( K_{max} \) is the maximum number of iterations allowed by the user.
- \( \|\nabla_c F(X_k)\|_F \leq \epsilon \), where \( \epsilon \in (0, 1) \) is a given scalar.
- \( \text{Err}_{Rel}^x_k < tolx \) and \( \text{Err}_{Rel}^f_k < tolf \), where \( tolx, tolf \in (0, 1) \).
- mean(Err_{Rel}^x_{1+k-min(k,T)} ,..., Err_{Rel}^x_k) \leq 10 tolx and mean((Err_{Rel}^f_{1+k-min(k,T)},..., Err_{Rel}^f_k)) \leq 10 tolf

where, the values \( \text{Err}_{Rel}^x_k \) and \( \text{Err}_{Rel}^f_k \) are defined by,

\[ \text{Err}_{Rel}^x_k := \frac{\|X_k - X_{k-1}\|_F}{\sqrt{n}}, \quad \text{Err}_{Rel}^f_k := \frac{|F(X_k) - F(X_{k-1})|}{|F(X_{k-1})| + 1}. \]

The default values for the parameters in our algorithms are: \( \epsilon = 1e-5 \), \( tolx = 1e-12 \), \( tolf = 1e-12 \), \( T = 5 \), \( K_{max} = 1000 \), \( \delta = 0.2 \), \( \rho_1 = 1e-4 \), \( \tau_m = 1e-20 \), \( \tau_M = 1e+20 \).

In the rest of this section we will denote by: “Nitr” to the number of iterations, “Time” to CPU time in seconds, “Fval” to the value of the evaluated objective function in the optimum estimated, “NrmG” to the norm of Gradient of the Lagrangean
function evaluated in the optimum estimate ($\|\nabla_c F(\hat{X})\|_F$) and “Feasi” to the feasibility error ($\|\hat{X}^T \hat{X} - I\|_F$), obtained by the algorithms.

5.2. Joint diagonalization problem

The Joint diagonalization problem (JDP) for $N$ matrices $A_1, A_2, \ldots, A_N$ symmetric of size $n \times n$, consists of finding an orthogonal matrix of size $n \times p$, which minimizes the sum of the squares of the diagonal entries of the matrices $X^T A_l X$, $l = 1, 2, \ldots, N$, or equivalently, maximize the sum of the squares of the entries of the diagonals of the matrices $X^T A_l X$, $l = 1, 2, \ldots, N$, see [38]. Getting a solution for this problem (JDP) is of great value for applications such as Independent Component Analysis (ICA) and also for The Blind Source Separation Problem among others (see [39]).

In [34,40] different methods on manifolds have been studied to solve the problem JDP on the Stiefel manifold, more specifically, consider the following problem:

$$\min \mathcal{F}(X) = - \sum_{l=1}^{N} \|\text{diag}(X^T A_l X)\|_F^2, \quad \text{s.a. } X \in \text{St}(n,p),$$

where $A_1, A_2, \ldots, A_N$ are real symmetric matrices of size $n \times n$ and where $\text{diag}(Y)$ is the matrix whose main diagonal matches the main diagonal of the matrix $Y$ and all other entries are zero.

In this subsection, we tested the efficiency of our algorithms when solving the problem (15), and compared our procedure with two of the state-of-the-art methods OptStiefel [25] and Grad_retrac [30]. In the following tables, we show the average of each values to be compared in a total of 100 executions. In addition, the maximum number of iterations is set at 8000, we use a tolerance for the gradient norm of $\epsilon = 1e-5$ and we take the following values, $\text{xtol} = 1e-14$ and $\text{ftol} = 1e-14$ as tolerances for the other stop criteria.

First, we perform an experiment varying $p$ and $n$ (this experiment was taken from [34]). In this case, we build the matrices $A_1, A_2, \ldots, A_N$ as follows. We generate $N = 10$ randomly $n \times n$ diagonal matrices $\Lambda_1, \Lambda_2, \ldots, \Lambda_N$, and a randomly chosen $n \times n$ orthogonal matrix $P$, where the diagonal entries $\lambda_{1}^{(l)}, \lambda_{2}^{(l)}, \ldots, \lambda_{n}^{(l)}$ of each $\Lambda_l$ are positive and in descending order. We then put $A_1, A_2, \ldots, A_N$ as $A_l = P \Lambda_l P^T$ for all $l = 1, 2, \ldots, N$. Observe that $X^* = P I_{n,p}$ is an optimal solution to the problem (15). As a starting point, we compute an approximate solution $X_0 = Q f(X^* + X^{rand})$, where $X^{rand}$ is a randomly chosen $n \times p$ matrix such that $\max_{i,j} |x_{ij}| \leq 0.01$, where $x_{ij}$ denotes the entry $(i, j)$ of matrix $X^{rand}$.

The numerical results, associated with this experiment for several values of $n$ and $p$, are contained in Table 1. From these table we can observe that, all the methods compared show a similar performance in terms of CPU time, but our method performs more iterations. However, the three algorithms get good solutions. In addition, in Figure 1, we illustrate the behavior of our proposed algorithm for a particular execution with $n = 50$, $p = 30$ and $N = 5$, building the problem as explained at the beginning of this paragraph.
Table 1. Numerical results for Experiment 1 with $N = 10$

<table>
<thead>
<tr>
<th>Methods</th>
<th>Nitr</th>
<th>Time</th>
<th>NrmG</th>
<th>Fval</th>
<th>Feasi</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>QR-CG</td>
<td>151.3</td>
<td>0.038</td>
<td>9.14e-6</td>
<td>2.2e-16</td>
<td>0.0203</td>
<td>9.25e-6</td>
</tr>
<tr>
<td>OptStiefel</td>
<td>97.6</td>
<td>0.025</td>
<td>8.84e-6</td>
<td>7.96e-16</td>
<td>0.0203</td>
<td>9.09e-6</td>
</tr>
<tr>
<td>Grad_retrac</td>
<td>97</td>
<td>0.031</td>
<td>8.1e-6</td>
<td>1.84e-14</td>
<td>0.0202</td>
<td>9.02e-6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Nitr</th>
<th>Time</th>
<th>NrmG</th>
<th>Fval</th>
<th>Feasi</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>QR-CG</td>
<td>258.5</td>
<td>0.25</td>
<td>9.29e-6</td>
<td>1.67e+2</td>
<td>2.23e-15</td>
<td>1.56e+2</td>
</tr>
<tr>
<td>OptStiefel</td>
<td>145.6</td>
<td>0.141</td>
<td>9.09e-6</td>
<td>9.25e-6</td>
<td>4.95e-14</td>
<td>2.48e-15</td>
</tr>
<tr>
<td>Grad_retrac</td>
<td>14.6e+2</td>
<td>0.187</td>
<td>9.32e-6</td>
<td>2.6e+2</td>
<td>2.27e-14</td>
<td>1.046</td>
</tr>
</tbody>
</table>

Figure 1. Behavior of QR-CG algorithm for Problem 2 with $n = 50$, $p = 30$ and $N = 5$. The y-axis is on logarithmic scale for the case of the gradient norm.

Next, we carry out another experiment, in order to test the methods starting from an initial point randomly generated and not necessarily close to a solution. We set $N = 10$ and build 100 JDPs generating each of the $N$ matrices $A_1, A_2, \ldots, A_N$ in the following way, first we generate randomly a matrix $\bar{A} \in \mathbb{R}^{n \times n}$ with all its entries following a Gaussian distribution, then we build $A_l$ for $A_l = \bar{A}^\top \bar{A}$ and thus ensure that every $A_l$ is symmetric. On the other hand, we generate at the initial point $X_0$ randomly on the Stiefel manifold. Table 2 shows the results obtained by the methods for different problem sizes. Based on the results summarized in Table 2, we observe that OptStiefel and our QR-CG are much more efficient methods than Grad_retrac in terms of CPU time, furthermore, our method presents competitive results compared to those obtained by the other two methods.

Table 2. Numerical results for Experiment 2 with $N = 10$

<table>
<thead>
<tr>
<th>Methods</th>
<th>Nitr</th>
<th>Time</th>
<th>NrmG</th>
<th>Fval</th>
<th>Feasi</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>QR-CG</td>
<td>208.7</td>
<td>0.032</td>
<td>3.8e-6</td>
<td>-2.1e-5</td>
<td>8.0e-16</td>
<td>2.05e-15</td>
</tr>
<tr>
<td>OptStiefel</td>
<td>161.1</td>
<td>0.036</td>
<td>1.2e-5</td>
<td>-2.1e-5</td>
<td>8.0e-16</td>
<td>2.05e-15</td>
</tr>
<tr>
<td>Grad_retrac</td>
<td>190.9</td>
<td>0.076</td>
<td>1.5e-5</td>
<td>-2.1e-5</td>
<td>2.13e-14</td>
<td>694.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Nitr</th>
<th>Time</th>
<th>NrmG</th>
<th>Fval</th>
<th>Feasi</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>QR-CG</td>
<td>765.9</td>
<td>0.61</td>
<td>5.0e-4</td>
<td>-1.7e+6</td>
<td>2.05e-15</td>
<td>842.6</td>
</tr>
<tr>
<td>OptStiefel</td>
<td>597.5</td>
<td>0.456</td>
<td>8.1e-6</td>
<td>-1.7e+6</td>
<td>6.11e-14</td>
<td>852.2</td>
</tr>
<tr>
<td>Grad_retrac</td>
<td>2443.5</td>
<td>5.75</td>
<td>5.0e-4</td>
<td>-1.7e+6</td>
<td>9.4e-15</td>
<td>1593.9</td>
</tr>
</tbody>
</table>
5.3. Total energy minimization

In this subsection, we consider a version of the total energy minimization problem:

\[
\min_{X \in \mathbb{R}^{n \times k}} E_{\text{total}}(X) = \frac{1}{2} \text{Tr}[X^\top LX] + \frac{\mu}{4} \rho(X)^\top L^\dagger \rho(X) \quad \text{s.t.} \quad X^\top X = I
\]  

(16)

where \( L \) is a discrete Laplacian operator, \( \mu > 0 \) is a constant, \( L^\dagger \) denote the Moore-Penrose generalized inverse of \( L \) and \( \rho(X) := \text{diag}(XX^\top) \) is the vector containing the diagonal elements of the matrix \( XX^\top \). The problem (16) is a simplified version of the Hartree-Fock (HF) total energy minimization problem and the Kohn-Sham (KS) total energy minimization problem in electronic structure calculations (see for details [41–44]). The first order necessary conditions for the total energy minimization problem (16) are given by:

\[
H(X)X - XL = 0 \quad \text{and} \quad X^\top X = I,
\]

(17)

(18)

where \( H(X) := L + \mu \text{Diag}(L^\dagger \rho(X)) \) and \( \Lambda \) is the Lagrange multipliers matrix. Here, the symbol \( \text{Diag}(x) \) is a diagonal matrix with a vector \( x \) on its diagonal. Observe that the equations (17)-(18) can be seen as a nonlinear eigenvalue problem.

The experiments 1.1-1.2 described below were taken from [42], we replay the experiments over 100 different starting points, moreover, we use a maximum number of iterations \( K = 4000 \) and tolerance of \( \epsilon = 1e-5 \). To show the efficacy of our methods solving the problem (16), we present the numerical results for problems 1.1-1.2 with different choices of \( n, k, \) and \( \mu \). In this subsection, we compare the Algorithm 1 with the Steepest Descent method (Steep-Dest), Trust-Region method (Trust-Reg) and Conjugate Gradient method (Conj-Grad) from manopt toolbox, see Ref. [45]1.

Experiment 1.1 [42]: We consider the nonlinear eigenvalue problem for \( k = 10; \mu = 1 \), and varying \( n = 200, 400, 800, 1000 \).

Experiment 1.2 [42]: We consider the nonlinear eigenvalue problem for \( n = 100 \) and \( k = 20 \) and varying \( \mu \).

In all these testing, the \( L \) matrix that appears in the problem (16) is built as the one-dimensional discrete Laplacian with 2 on the diagonal and 1 on the sub- and sup-diagonals.

The results corresponding to the experiment 1.1 are presented in Table 3. We see from this table that our method (\texttt{QR_CG}) is more efficient than the rest of the algorithms when \( n \) is small, but if \( n \) is moderately large then \texttt{Conj-Grad} becomes the best algorithm for this specific application. However, all methods obtain similar results in terms of the gradient norm (NrmG) and the optimal objective value (Fval).

Table 4 lists numerical results for Example 1.2. In this table, we note that our procedure gets better performance than the others methods, due to our \texttt{QR_CG}

---

1The tool-box manopt is available in http://www.manopt.org/
Table 3. Numerical results for Experiment 1.1

<table>
<thead>
<tr>
<th>Methods</th>
<th>Nitr</th>
<th>Time</th>
<th>NrmG</th>
<th>Fval</th>
<th>Feasi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust-Reg</td>
<td>30.3</td>
<td>0.158</td>
<td>3.48e-6</td>
<td>35.7086</td>
<td>3.27e-15</td>
</tr>
<tr>
<td>Steep-Dest</td>
<td>249.3</td>
<td>0.821</td>
<td>3.40e-6</td>
<td>35.7086</td>
<td>1.99e-15</td>
</tr>
<tr>
<td>Conj-Grad</td>
<td>68.2</td>
<td>0.107</td>
<td>8.14e-6</td>
<td>35.7086</td>
<td>2.20e-15</td>
</tr>
<tr>
<td>QR_CG</td>
<td>101.9</td>
<td>0.079</td>
<td>8.02e-6</td>
<td>35.7086</td>
<td>7.62e-16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Nitr</th>
<th>Time</th>
<th>NrmG</th>
<th>Fval</th>
<th>Feasi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust-Reg</td>
<td>31.9</td>
<td>0.987</td>
<td>5.38e-6</td>
<td>35.7086</td>
<td>5.69e-15</td>
</tr>
<tr>
<td>Steep-Dest</td>
<td>252.8</td>
<td>1.766</td>
<td>2.97e-6</td>
<td>35.7086</td>
<td>4.16e-15</td>
</tr>
<tr>
<td>Conj-Grad</td>
<td>69.8</td>
<td>0.265</td>
<td>7.93e-6</td>
<td>35.7086</td>
<td>5.61e-15</td>
</tr>
<tr>
<td>QR_CG</td>
<td>102.5</td>
<td>1.415</td>
<td>7.96e-6</td>
<td>35.7086</td>
<td>7.72e-16</td>
</tr>
</tbody>
</table>

Table 4. Numerical results for Experiment 1.2

<table>
<thead>
<tr>
<th>Methods</th>
<th>Nitr</th>
<th>Time</th>
<th>NrmG</th>
<th>Fval</th>
<th>Feasi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust-Reg</td>
<td>13.4</td>
<td>0.085</td>
<td>1.37e-6</td>
<td>1.4484</td>
<td>5.71e-15</td>
</tr>
<tr>
<td>Steep-Dest</td>
<td>410.8</td>
<td>1.618</td>
<td>3.53e-6</td>
<td>1.4484</td>
<td>2.45e-15</td>
</tr>
<tr>
<td>Conj-Grad</td>
<td>78.5</td>
<td>0.134</td>
<td>8.48e-6</td>
<td>1.4484</td>
<td>4.12e-15</td>
</tr>
<tr>
<td>QR_CG</td>
<td>125.5</td>
<td>0.042</td>
<td>8.14e-6</td>
<td>1.4484</td>
<td>4.01e-15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Nitr</th>
<th>Time</th>
<th>NrmG</th>
<th>Fval</th>
<th>Feasi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust-Reg</td>
<td>24.1</td>
<td>0.176</td>
<td>4.00e-6</td>
<td>7.8706</td>
<td>5.20e-15</td>
</tr>
<tr>
<td>Steep-Dest</td>
<td>567.9</td>
<td>2.186</td>
<td>3.35e-6</td>
<td>7.8706</td>
<td>2.33e-15</td>
</tr>
<tr>
<td>Conj-Grad</td>
<td>87.7</td>
<td>0.148</td>
<td>8.70e-6</td>
<td>7.8706</td>
<td>4.75e-15</td>
</tr>
<tr>
<td>QR_CG</td>
<td>146.9</td>
<td>0.058</td>
<td>7.87e-6</td>
<td>7.8706</td>
<td>1.07e-15</td>
</tr>
</tbody>
</table>

converge in less CPU time in all of the experiments listed in this table, we also observe that all the methods get good results.

6. Conclusions.

In this paper, we propose a numerically reliable iterative algorithm for optimization problems on the Stiefel manifold. The algorithm can be seen as a novel Riemannian conjugate gradient that does not employ a vector transport explicitly. In addition, we introduce a new vector transport for the Stiefel manifold, that is used only for theoretical purposes, to demonstrate the connection of our algorithm with the standard Riemannian conjugate gradient methods. In order to reduce the complexity of each iteration, the QR decomposition via Cholesky factorization is considered to preserve the feasibility of each point. In addition, we perform two computational experiments on the joint diagonalization problem and on a simplified version of the total energy minimization problem. Experimental results showed that the proposed procedure obtains a competitive performance against some algorithms of the state of the art.
Acknowledgements

This work was supported in part by CONACYT (Mexico).

References

[18] Donald Goldfarb, Zaiwen Wen, and Wotao Yin. A curvilinear search method for p-


[42] Zhi Zhao, Zheng-Jian Bai, and Xiao-Qing Jin. A riemannian newton algorithm for nonlin-
774, 2015.

total energy minimization in electronic structure calculations. *Journal of Computational

[44] Chao Yang, Juan C Meza, and Lin-Wang Wang. A trust region direct constrained mini-

Manopt, a matlab toolbox for optimization on manifolds. *Journal of Machine Learn-