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

Implicit steepest descent algorithm for optimization with orthogonality constraints

Harry Oviedo

Received: date / Accepted: date

Abstract Optimization problems with orthogonality constraints appear widely in applications from science and engineering. We address these types of problems from a numerical approach. Our new framework combines the steepest gradient descent, using implicit information, with a projection operator in order to construct a feasible sequence of points. In addition, we adopt an adaptive Barzilai–Borwein steplength mixed with a globalization technique in order to speed–up the convergence of our procedure. The global convergence, and some theoretical related to our algorithm are proved. The effectiveness of our proposed algorithm is demonstrated on a variety of problems including Rayleigh quotient maximization, heterogeneous quadratics minimization, weighted orthogonal procrustes problems and total energy minimization. Numerical results show that the new procedure can outperform some state–of–the–art solvers on some practically problems.

Keywords Orthogonality constrained optimization · Riemannian optimization · gradient–type methods · eigenvalue problem · total energy minimization.

1 Introduction.

A large range of problems in dimension reduction and machine learning [27], data mining and pattern recognition [16,27], compressive sensing [7], p–harmonic flow on spheres [21], joint diagonalization [3,38,42], computer vision, matrix completion [8,10,29,46], procrustes analysis [18,35] and image processing [47]

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

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

can be modeled as optimization problems where the variables lie on a Riemannian manifold. These kind of problems can be mathematically formulated as

$$\min f(x) \quad \text{s.t.} \quad x \in \mathcal{M}, \tag{1}$$

where $f : \mathcal{M} \to \mathbb{R}$ is a smooth cost function and $\mathcal{M}$ is a Riemannian manifold. A specific case of problem (1) is the well–known Stiefel manifold constrained optimization problem, which can be formulated as

$$\min_{X \in \mathbb{R}^{n \times p}} \mathcal{F}(X) \quad \text{s.t.} \quad x \in \mathsf{St}(n,p), \tag{2}$$

where $\mathcal{F} : \mathbb{R}^{n\times p} \to \mathbb{R}$ is a continuously differentiable function and $\mathsf{St}(n,p) := \{X \in \mathbb{R}^{n\times p} : X^\top X = I_p\}$ is so–called the Stiefel manifold, which was due to Eduard Stiefel (see [41]). Particularly in case of $p = 1$, problem (2) reduces to the spherically constrained optimization problem. In case of $p = n$, the feasible set becomes orthogonal group $\mathcal{O}(n)$. The feasible set of (2) endowed with the Frobenius inner product can be seen as an embedded Riemannian sub–manifold of the Euclidean matrix space $\mathbb{R}^{n\times p}$, whose dimension is equal to $np - \frac{p(p+1)}{2}$, see [4]. This particular Riemannian optimization problem appears frequently in principal component analysis [26,51], Kohn–Sham total energy minimization [25,45], the leakage interference minimization [30], orthogonal procrustes problem [12,18] and low–rank correlation matrix problem [36].

The non–convexity and the nonlinearity of $St(n,p)$ are two of the major challenges of problem (2) since there might be several local minimizers, from which finding global minimizers is generally difficult. In fact, only a few instances of problem (2) (for example, finding the extreme eigenvalues, orthogonal procrustes problems) have global solutions with closed expressions. In most cases, we need to use some iterative method to approximate the solution of this kind of problems. In particular, applications such as the maxcut problem and the leakage interference minimization [30] are NP–hard.

Several Riemannian line–search methods have been developed to address the numerical solution of (2), which typically require the use of geodesics or projection operators, and are characterized by exploiting the geometric structure of $\mathsf{St}(n,p)$ and by preserving the structure of the problem (avoid the vectorization of X). For instance, there exists Riemannian gradient–like methods [2,4,12,11,15,18,31,34,40,45], Riemannian conjugate gradiente methods [1,4,15,33,40,50], Newton's methods, such as those in [4,38,40]. Riemannian and implicit Riemannian trust–region methods are proposed in [4,5]. Other approaches to deal with problem (2) have been introduced in [19,25,28,32,44].

In this paper, we introduce and analyse a new gradient type optimization method to deal with Stiefel manifold constrained optimization problems. Essentially, our approach build the new trial point by performing a curvilinear search on the Stiefel manifold, using an approximation of the implicit gradient of the Lagrangian function along the iterations. The feasibility of any iterate generated by our procedure is preserved by using a projection operator. The numerical performance of our implicit gradient method is further improved by incorporating an adaptive Barzilai and Borwein step–size [6], in combination with the Zhang–Hager non–monotone line–search [48]. The convergence analysis is studied. In addition, we conduct some numerical experiments to compare the behavior of our approach with some state–of–the–art algorithms, considering sphere constrained optimization problems, weighted orthogonal procrustes problems, heterogeneous quadratics minimization problems and nonlinear eigenvalues problems.

Outline. The rest of this paper is organized as follows. In section 2, we introduce some important notations and tools. In subsection 2.1, we establish the Karush–Khun–Tucker conditions for problem (2). In section 3, we describe our gradient–type method with implicit information to tackle optimization problems with orthogonality constraints, and we establish some theoretical properties related to our proposal. A theoretical analysis concerning the global convergence of our procedure is presented in section 4. Numerical results are reported in section 5 in order to illustrate the efficiency and the effectiveness of our approach. Finally, we give some concluding remarks in section 6.

2 Notations and important tools.

In this section, we introduce some important notations for the well understanding of this work, and review the geometry of the Stiefel manifold, as discussed in [4,45]. We also give the Karush–Kuhn–Tucker conditions associated to (2).

Throughout this paper, we say that $W \in \mathbb{R}^{n\times n}$ is skew–symmetric if $W = -W^\top$. The trace of X is defined as the sum of the diagonal elements which we denote by $Tr[X]$. The Frobenius inner product of two matrices $A, B \in \mathbb{R}^{m\times n}$ is given by $\langle A, B\rangle := \sum_{i,j} a_{ij}b_{ij} = Tr[A^\top B]$. The Frobenius norm is defined by $||A||_F = \sqrt{\langle A, A\rangle}$. Let $\mathcal{F} : \mathbb{R}^{n\times p} \to \mathbb{R}$ be a differentiable function, we denote by $G := \mathcal{DF}(X) = (\frac{\partial \mathcal{F}(X)}{\partial X_{ij}})$ the matrix of partial derivatives of $\mathcal{F}$ (the Euclidean gradient of $\mathcal{F}$). Additionally, the directional derivative of $\mathcal{F}$ along a given matrix Z at X is defined by:

$$\mathcal{DF}(X)[Z] := \lim_{t\to 0} \frac{\mathcal{F}(X+tZ) - \mathcal{F}(X)}{t} = \langle \mathcal{DF}(X), Z\rangle. \tag{3}$$

The tangent space $T_X\mathsf{St}(n,p)$ of the Stiefel manifold at $X \in \mathsf{St}(n,p)$ is given by

$$T_X\mathsf{St}(n,p) = \{Z \in \mathbb{R}^{n\times p} : Z^\top X + X^\top Z = 0\}. \tag{4}$$

In addition, under the standard metric $\langle \xi_1, \xi_2\rangle = Tr[\xi_1^\top \xi_2]$, the orthogonal projection $P_X[\cdot]$ at X onto $T_X\mathsf{St}(n,p)$ is expressed as

$$P_X[\xi] = \xi - X\mathrm{sym}(X^\top \xi), \quad \text{with} \quad \xi \in \mathbb{R}^{n\times p}, \tag{5}$$

where $\mathrm{sym}(W) = \frac{1}{2}(W^\top + W)$ denotes the symmetric part of the matrix $W \in \mathbb{R}^{m\times m}$.

The following proposition establishes a useful relationship between the Frobenius norm of a particular skew–symmetric matrix and the trace function.

Proposition 1 *Given $L, R \in \mathbb{R}^{n\times p}$ two rectangular matrices and define $W = LR^\top - RL^\top$. Then*

$$||W||_F^2 = 2\,Tr[\,L^\top WR\,]. \tag{6}$$

Proof By using trace properties we have

$$\begin{aligned}
||W||_F^2 &= Tr[\,W^\top W\,]\\
&= Tr[\,(LR^\top - RL^\top)^\top(LR^\top - RL^\top)\,]\\
&= Tr[\,(RL^\top - LR^\top)(LR^\top - RL^\top)\,]\\
&= Tr[\,RL^\top LR^\top\,] - Tr[\,RL^\top RL^\top\,] - Tr[\,LR^\top LR^\top\,] + Tr[\,LR^\top RL^\top\,]\\
&= Tr[\,RL^\top LR^\top\,] + Tr[\,LR^\top RL^\top\,] - 2\,Tr[\,RL^\top RL^\top\,]\\
&= 2Tr[\,LR^\top RL^\top\,] - 2\,Tr[\,RL^\top RL^\top\,]\\
&= 2(\,Tr[\,L^\top LR^\top R\,] - Tr[\,L^\top RL^\top R\,]\,)\\
&= 2Tr[\,L^\top LR^\top R - L^\top RL^\top R\,]\\
&= 2Tr[\,L^\top\,(\,LR^\top - RL^\top\,)\,R\,]\\
&= 2Tr[\,L^\top WR\,],
\end{aligned}$$

which proves the proposition. □

Another important tool that we shall employ is the projection operator over the Stiefel manifold, which is define by

$$\pi(X) = \arg \min_{P\in \mathsf{St}(n,p)} ||X - P||_F. \tag{7}$$

The following proposition, demonstrated in [31], provides us with a close expression to compute the projection of any matrix over $\mathsf{St}(n,p)$.

Proposition 2 *Let $X \in \mathbb{R}^{n\times p}$ be a rank p matrix. Then, $\pi(X)$ is well defined. Moreover, if the singular value decomposition of X is $X = U\Sigma V^\top$, then $\pi(X) = UI_{n,p}V^\top$, where $I_{n,p} \in \mathbb{R}^{n\times p}$ is the rectangular diagonal matrix with diagonal entries equal 1.*

Remark 1 In view of $p \leq n$, we can compute the projection $\pi(X)$ efficiently using the spectral–decomposition based SVD computation. Specifically, we have

$$\pi(X) = XVD^{-1/2}V^\top,$$

where $V \in \mathbb{R}^{p\times p}$ is an orthogonal matrix and D is a diagonal matrix satisfying the eigenvalue decomposition $X^\top X = VDV^\top$. This procedure corresponds to the Algorithm 3 in [17].

2.1 Optimality conditions.

The Lagrangian function associated to the optimization problem (2) is given by

$$\mathcal{L}(X, \Lambda) = \mathcal{F}(X) - \frac{1}{2}Tr[\,\Lambda^\top (X^\top X - I_p)\,], \tag{8}$$

where Λ represents the Lagrange multipliers matrix, which is symmetric because the constraint $X^\top X$ is also symmetric. By differentiating the Lagrangian function with respect to the primal and dual variables, we obtain the first order necessary optimality conditions related to problem (2):

$$\nabla_X \mathcal{L}(X, \Lambda) = G - X\Lambda = 0, \tag{9}$$

$$\nabla_\Lambda \mathcal{L}(X, \Lambda) = X^\top X - I_p = 0. \tag{10}$$

The following technical result, taking from [45], provides us with an important tool to identify critical points of Lagrangian function.

Lemma 1 *Suppose that X is a local minimizer of problem (2). Then X satisfies the first order optimality conditions* (9)–(10) *with the associated Lagrangian multiplier $\Lambda = G^\top X$. Define*

$$\nabla\mathcal{F}(X) := G - XG^\top X \quad \textit{and} \quad A(X) := GX^\top - XG^\top,$$

then $\nabla\mathcal{F}(X) = A(X)X$. Moreover, $\nabla\mathcal{F}(X) = 0$ if and only if $A(X) = 0$.

Remark 2 Let $X \in \mathsf{St}(n, p)$ be an arbitrary matrix. If we endow the Stiefel manifold with the canonical metric given by

$$\langle \xi_X, \eta_X \rangle_c := Tr\left[\xi_X^\top \left(I - \frac{1}{2}XX^\top\right)\eta_X\right], \quad \forall \xi_X, \eta_X \in T_X\mathsf{St}(n, p),$$

then $\nabla\mathcal{F}(X)$ coincides with the Riemannian gradient of the function $\mathcal{F} : \mathsf{St}(n, p) \to \mathbb{R}$ (the restriction of the objective function to the Stiefel manifold), for an entire description about this fact see [46,Section 4].

3 A feasible implicit gradient method.

In this section, we specify our implicit gradient projection method. It represents our efficient feasible optimization scheme that consists of two main steps. In order to introduce the new approach, let us consider the gradient flow of an energy $\mathcal{F} : \mathcal{M} \to \mathbb{R}$ defined on a Riemannian manifold $\mathcal{M}$ (for the purposes of this work, the Riemannian manifold $\mathcal{M}$ is $\mathsf{St}(n, p)$),

$$\dot{Z}(\tau) = -\nabla_{\mathcal{M}}\mathcal{F}(Z(\tau)), \quad Z(0) = X_0, \tag{11}$$

where $X_0 \in \mathcal{M}$, and $\nabla_{\mathcal{M}}\mathcal{F}(\cdot)$ denotes the Riemannian gradient of $\mathcal{F}$. Now, if we apply the forward Euler method to solve the ordinary differential equation (11), we get the update formula of the classical gradient method

$$Z_{k+1} = Z_k - \tau_k \nabla_{\mathcal{M}}\mathcal{F}(Z_k), \tag{12}$$

which becomes the Riemannian gradient method, if we incorporate some retraction to preserve the feasibility of each iterate.

Another possibility to iteratively solve the gradient flow (11) is by applying the backward Euler method (or also known as the implicit Euler method). The backward Euler method for the gradient flow on matrix Lie groups, is considered in [22]. Specifically, the authors in [22] mention this method to solve gradient flows over the orthogonal group $\mathcal{O}(n) \equiv \mathsf{St}(n,n)$, this set admits a Lie group and its Lie algebra is the set of skew–symmetric matrices. In particular, we propose a slight modification of this method but applied on the Stiefel manifold instead of matrix Lie groups (note that the Stiefel manifold is not a Lie group). The implicit Euler method for the ODE defined in (11) leads to the following update scheme

$$Z_{k+1} = Z_k - \tau_k \nabla_{\mathcal{M}}\mathcal{F}(Z_{k+1}). \tag{13}$$

Since the iterative process (13) is implicitly defined and the Riemannian gradient of $\mathcal{F}$ is typically a highly nonlinear functional, in practice it is difficult to implement this approach in its theoretical form. To overcome this difficulty, we propose to approximate the Riemannian gradient $\nabla_{\mathcal{M}}\mathcal{F}(Z_{k+1})$ as follows

$$\begin{aligned} \nabla_{\mathcal{M}}\mathcal{F}(Z_{k+1}) &= A(Z_{k+1})Z_{k+1} \\ &\approx A(Z_k)Z_{k+1}. \end{aligned} \tag{14}$$

By combining (13) and (14), we obtain part of the update formula of our procedure,

$$Z_{k+1} = Z_k - \tau_k A(Z_k)Z_{k+1}. \tag{15}$$

Based on this idea, we propose a two–phase scheme to update the iterates. Firstly, we generate an auxiliary point $Z(\tau)$ by using the gradient method with implicit information described in (15), which can be seen as an adaptation of the classical steepest descent method (12), after that we compute the new iterate by projecting the auxiliary point over the Stiefel manifold, in order to preserve the manifold structure. Specifically, in the first step, we compute a trial point, from the previous point $X \in \mathsf{St}(n,p)$, determined by the following approximated implicit gradient descent scheme

$$Z(\tau) = X - \tau A(X)Z(\tau), \tag{16}$$

where $\tau > 0$ represents the step–size. Notice that the implicit scheme (16) can be explicitly rewritten as

$$Z(\tau) = (I_n + \tau A(X))^{-1}X. \tag{17}$$

Since the auxiliary point $Z(\tau)$ may not verify the constraints of the optimization problem (2), we use the projection operator over the Stiefel manifold (7). Thus, given the previous point $X \in \mathsf{St}(n,p)$, the new iterate $Y(\tau)$ is determined by the following scheme

$$Y(\tau) := \pi(Z(\tau)) = \pi((I_n + \tau A(X))^{-1}X). \tag{18}$$

Observe that if we replace the matrix product $A(X)Z(\tau)$ with $A(X)X$, in equation (16), then the proposed method would be reduced to the Riemannian steepest descent method proposed by Manton in [31] and improved by Oviedo et. al. in [34]. Hence, the update scheme given by (18) can be seen as a modified version of the Manton's method that introduces implicit information based on the gradient approximation $\nabla\mathcal{F}(Z(\tau)) \approx A(X)Z(\tau)$.

Our main idea focuses on the adaptation of the implicit Euler method to solve (2). However, other well–studied methods in numerical ODEs can also be applied to solve the gradient flow (11) and can lead to new optimization methods to solve (2), e.g., Runge–Kutta methods. This observation was also partly investigated in [12,45] and [20]. In particular, in [45] Wen and Yin are inspired by the Crank–Nicolson method to derive an efficient method for the solution of (2). In addition, Dalmau and Oviedo in [12] adapted the Adams–Moulton method to solve the manifold constrained optimization problem (2).

Remark 3 There is a direct relationship between the Crank–Nicolson based method [45], the Manton's Riemannian gradient method [31] and our proposed method (18). In particular, The Riemannian gradient method based on the Crank–Nicolson formula is implicitly defined as follows

$$Y_{\mathrm{CN}}(\tau) = X - \tau A(X)\left(\frac{1}{2}X + \frac{1}{2}Y_{\mathrm{CN}}(\tau)\right). \tag{19}$$

Then, the new iterate is calculated as $X_{k+1} = Y_{CN}(\tau)$. Since this curve preserves the manifold structure, we have that $X_{k+1} = \pi(Y_{CN}(\tau)) = Y_{CN}(\tau)$. Note that we can write our method and the Wen and Yin method in a single update formula

$$Y_\theta(\tau) = X - \tau A(X)\left((1-\theta)X + \theta Y_\theta(\tau)\right), \tag{20}$$

where $\theta \in [0,1]$, and then we compute the new iterated by $X_{k+1} = \pi(Y_\theta(\tau))$. Observe that if we select $\theta = 0.5$ then the update scheme (20) reduces to the Crank–Nicolson based method (19), while if we choose $\theta = 1$ then we recover our proposed implicit gradient method, see the equations (16)–(18). In addition, if we select $\theta = 0$ in (20), we obtain the Riemannian steepest descent method proposed by Manton in [31].

Example 1: In order to show the numerical behavior of our procedure, we present a numerical example to compare the performance of our update scheme

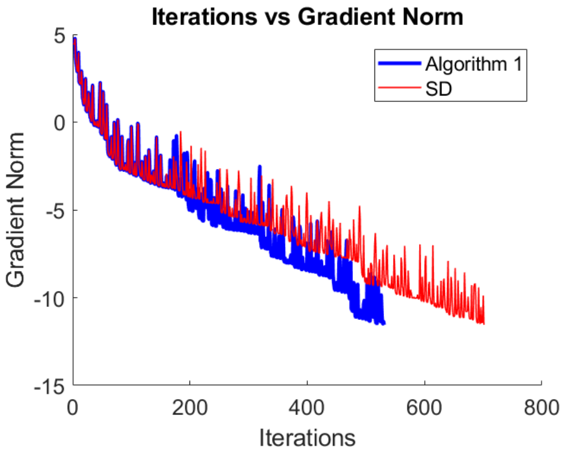


Fig. 1 Convergence history of implicit gradient method (18) and steepest descent method, from the same initial point, for the minimization of a quadratic function over the Stiefel manifold. The y-axis is on a logarithmic scale.

(18) with the steepest descent method proposed by Manton [31], both using the adaptive Barzilai–Borwein step–size described in subsection 3.1. To do this, we maximize the function $\mathcal{F}(X) = \frac{1}{2}Tr[X^\top AX]$ over the Stiefel manifold, considering $n = 5000$ and $p = 3$, where the data matrix A and the starting point X_0 are randomly generated by the following Matlab commands:

$$\texttt{randn}('\text{seed}', 1); \quad A = \texttt{randn}(n); \quad A = 0.5 * (A' + A),$$

and

$$X_0 = \texttt{randn}(n, p); \quad [X_0, \sim] = \texttt{qr}(X_0, 0).$$

For this example, we fix the tolerance $\epsilon = 1e$-5, (the termination rules of our algorithm are explained in detail in Section 5). In Figure 1, we plot the convergence history of the gradient norm of the Lagrangian function along the iterations, for both methods. From Figure 1, we clearly see that the incorporation of implicit information can improve the numerical performance of the Riemannian steepest descent method.

The following lemma establishes some interesting properties of our scheme (18).

Lemma 2 *Consider the curve $Y(\tau)$ defined in* (18)*, with $X \in St(n, p)$. Then the following properties are satisfied.*

1. *The matrix $I_n + \tau A$ is positive definite for all $\tau \in \mathbb{R}$.*
2. *The derivative of $Y(\tau)$ with respect to τ at $\tau = 0$ is*

$$\dot{Y}(0) = -\nabla\mathcal{F}(X). \tag{21}$$

3. If we rewrite the matrix A *as* $A = UV^\top$ *where* $U = [G, X]$ *and* $V = [X, -G]$ *then* (17) *is equivalent to the following scheme*

$$Z(\tau) = X - \tau U(I_{2p} + \tau V^\top U)^{-1}V^\top X. \tag{22}$$

Proof 1. In view of $A(X) = GX^\top - XG^\top$ is skew–symmetric, $q^\top A(X)q = 0$ holds for any non–zero vector $q \in \mathbb{R}^n$, which implies that $I_n + \tau A(X)$ is positive definite for all $\tau \in \mathbb{R}$.

2. By differentiating the curve $Z(\tau)$ defined in (17) with respect to τ, we obtain

$$\dot{Z}(\tau) = -A(X)Z(\tau) - \tau A(X)\dot{Z}(\tau), \tag{23}$$

or equivalently,

$$\dot{Z}(\tau) = -(I_n + \tau A(X))^{-1}A(X)Z(\tau), \tag{24}$$

Substituting $\tau = 0$ in (24) we obtain $\dot{Z}(0) = -\nabla\mathcal{F}(X)$. From Lemma 7 in [12], we have

$$\dot{Y}(0) = P_X[\dot{Z}(0)] = -P_X[\nabla\mathcal{F}(X)] = -\nabla\mathcal{F}(X).$$

3. To obtain this equivalence, we apply the Sherman–Morrison–Woodbury formula to the matrix $(I_n + \tau A(X))^{-1}$, that is

$$(I_n + \tau A(X))^{-1} = (I_n + \tau UV^\top)^{-1} \tag{25}$$
$$= I_n + \tau U(I_{2p} + \tau V^\top U)^{-1}V^\top. \tag{26}$$

Substituting (26) in the update scheme (17), we obtain

$$Z(\tau) = (I_n + \tau A(X))^{-1}X \tag{27}$$
$$= (I_n + \tau U(I_{2p} + \tau V^\top U)^{-1}V^\top)X \tag{28}$$
$$= X + \tau U(I_{2p} + \tau V^\top U)^{-1}V^\top X, \tag{29}$$

which completes the proof. □

Now, we derive a relation between the directional derivative of $\mathcal{F}(\cdot)$, considering the curve (18), and the Riemannian gradient norm of $\mathcal{F}(\cdot)$.

Lemma 3 *Given* $X \in \mathit{St}(n, p)$ *and consider the update scheme* (18)*. Then,*

$$\mathcal{DF}(X)[\dot{Y}(0)] \leq -\frac{1}{2}||\nabla\mathcal{F}(X)||_F^2, \tag{30}$$

that is, $Y(\tau)$ *is a descent curve at* $\tau = 0$*.*

Proof By differentiating the curve $Y(\tau)$ respect τ and evaluating at $\tau = 0$, we obtain that $\dot{Y}(0) = -\nabla\mathcal{F}(X)$. It follows from the definition of the directional derivative (3) and Proposition 1 that

$$\begin{aligned} \mathcal{DF}(X)[\dot{Y}(0)] &= -Tr[G^\top \nabla\mathcal{F}(X)] \\ &= -Tr[G^\top(GX^\top - XG^\top)X] \\ &= -\frac{1}{2}||A(X)||_F^2. \end{aligned} \tag{31}$$

On the other hand, using the Lemma 2 demonstrated in [34], we obtain

$$||\nabla\mathcal{F}(X)||_F^2 = ||A(X)X||_F^2 = ||X^\top A(X)||_F^2 \leq ||A(X)||_F^2.$$

Finally, merging this last expression with (31), we arrive at

$$\mathcal{DF}(X)[\dot{Y}(0)] \leq -\frac{1}{2}||\nabla\mathcal{F}(X)||_F^2,$$

which proves the lemma. □

3.1 Step–size selection rules.

This section is devoted to describe the selection of the step–size τ in our iterative approach (18). It is well–known that the Barzilai–Borwein [6] step–sizes (BB–steps) can greatly speed up the convergence of the gradient type methods, and require low computational demand per iteration. In fact, the BB–steps may allow some significant reduction in the number of line searches and also in the number of function evaluations. In view of these numerical advantages, we use the following two BB–steps,

$$\tau_k^{BB1} = \frac{||S_{k-1}||_F^2}{|\langle S_{k-1}, Y_{k-1}\rangle|} \quad \text{and} \quad \tau_k^{BB2} = \frac{|\langle S_{k-1}, Y_{k-1}\rangle|}{||Y_{k-1}||_F^2}, \tag{32}$$

where $S_{k-1} = X_k - X_{k-1}$ and $Y_{k-1} = \nabla\mathcal{F}(X_k) - \nabla\mathcal{F}(X_{k-1})$. In this work, we adopt an adaptive approach to select the step–size τ_k for the k–th iteration. Such approach considers to choose τ_k as follow

$$\tau_k^{ABB} = \begin{cases} \tau_k^{BB1} \text{ for odd k;} \\ \tau_k^{BB2} \text{ for even k.} \end{cases} \tag{33}$$

Since BB–steps do not necessary guarantee descent of the objective function along the iterations, these step–sizes are frequently used in combination with some globalization technique in order to assure the global convergence. For this end, we use the non–monotone globalization technique proposed by Zhang and Hager in [48]. The above description leads us to Algorithm 1.

Algorithm 1 Implicit Steepest Descent Method for Optimization with Orthogonality Constraints (Implicit–SD)

```
Require: $X_0 \in \mathsf{St}(n,p)$, $\tau > 0$, $0 < \tau_m \leq \tau_M$, $\eta \in [0,1)$, $\rho_1, \epsilon, \delta \in (0,1)$, $Q_0 = 1$, $C_0 = \mathcal{F}(X_0)$, $k = 0$.
Ensure: $X^*$ an $\epsilon$–KKT point.
while $||\nabla\mathcal{F}(X_k)||_F > \epsilon$ do
    while $\mathcal{F}(Y(\tau)) > C_k + \rho_1\tau\mathcal{D}\mathcal{F}(X_k)[\dot{Y}(0)]$ do
        $\tau = \delta\tau$,
    end while
    Update $X_{k+1} = Y(\tau)$, with $Y(\cdot)$ according to (18).
    Calculate $Q_{k+1} = \eta Q_k + 1$ and $C_{k+1} = (\eta Q_k C_k + \mathcal{F}(X_{k+1}))/Q_{k+1}$.
    Choose $\tau = \tau_{k+1}^{ABB}$ with $\tau_{k+1}^{ABB}$ defined as in (33).
    $\tau = \max(\min(\tau, \tau_M), \tau_m)$.
    Increment $k$ as $k = k + 1$.
end while
$X^* = X_k$.
```

4 Convergence Analysis.

In this section, we analyse the global convergence of Algorithm 1. The convergence analysis of our method follows the proof of the global convergence of the Riemannian gradient method with the Zhang–Hager's non–monotone technique [48] presented in [23, Theorem 3], which also appear in [20, Theorem 5.6]. Unlike the Theorem 3 that appear in [23]; in our analysis, we completed the missing steps of the proof of Theorem 3, following the steps of Theorem 4.3.1 described in [4]. In addition, note that in [20, 23] the convergence analysis is presented only for the Riemannian gradient method (Euler explicit). However, we observe that the same convergence analysis can be used to demonstrate the global convergence of any curvilinear procedure $Y_k(\tau)$ that satisfies $\dot{Y}_k(0) = -\nabla\mathcal{F}(X_k)$. Observe that this particular property is demonstrated for our approach in Lemma 2. Therefore, in this section, we follow the demonstration presented in [23], adapted for the case of the Stiefel manifold, in order to make our manuscript self–contained.

Lemma 4 *Let $\{X_k\}$ be an infinite sequence generated by Algorithm 1. Then we have*

$$\mathcal{F}(X_k) \leq C_k, \quad \forall k. \tag{34}$$

Proof From equation (31), we have $\mathcal{D}\mathcal{F}(X_k)[\dot{Y}(0)] = -\frac{1}{2}||A(X_k)||_F^2 < 0$. Now, define $\psi : \mathbb{R} \to \mathbb{R}$ by

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

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

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

Then using the fact that $\mathcal{DF}(X_k)[\dot{Y}(0)] < 0$, it follows from the non–monotone Zhang–Hager condition that

$$\mathcal{F}(X_k) \leq C_{k-1} + \rho_1 \tau \mathcal{DF}(X_k)[\dot{Y}(0)] < C_{k-1}, \tag{35}$$

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

$$\mathcal{F}(X_k) = \psi(0) \leq \psi(\bar{\alpha}) = C_k, \tag{36}$$

which completes the proof. □

Lemma 5 *Let $\{X_k\}$ be an infinite sequence generated by Algorithm 1. Then the sequence $\{C_k\}$ is monotonically decreasing.*

Proof It follows from Lemma 3 and the algorithm construction that

$$\begin{aligned} C_{k+1} &= \frac{\eta Q_k C_k + \mathcal{F}(X_{k+1})}{Q_{k+1}} \\ &\leq \frac{\eta Q_k C_k + C_k + \rho_1 \tau_k \mathcal{DF}(X_k)[\dot{Y}(0)]}{Q_{k+1}} \\ &\leq \frac{\eta Q_k C_k + C_k - \frac{1}{2}\rho_1 \tau_k ||\nabla \mathcal{F}(X_k)||_F^2}{Q_{k+1}} \\ &< \frac{(\eta Q_k + 1) C_k}{Q_{k+1}} \\ &= C_k. \end{aligned}$$

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

Theorem 1 *Let $\{X_k\}$ be an infinite sequence generated by Algorithm 1. Then any accumulation point X_* of $\{X_k\}$ satisfies the Karush–Kuhn–Tucker conditions* (9)–(10).

Proof Let $X_* \in \mathsf{St}(n,p)$ be an arbitrary accumulation point of $\{X_k\}$ and let $\{X_k\}_{\mathcal{K}}$ be a corresponding subsequence that converges to X_*. It follows from the continuity of $\mathcal{F}$ that $\lim_{k\in\mathcal{K}} \mathcal{F}(X_k) = \mathcal{F}(X_*)$, then using Lemma 4 and 5 we can infer that $C_* \in \mathbb{R}$. Hence, we have

$$\sum_{k=0}^{\infty} \frac{-\rho_1 \tau_k \mathcal{DF}(X_k)[\dot{Y}(0)]}{Q_{k+1}} \leq \sum_{k=0}^{\infty} C_k - C_{k+1} = C_0 - C_* < \infty. \tag{37}$$

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

$$\lim_{k\in\mathcal{K}} -\tau_k \mathcal{DF}(X_k)[\dot{Y}(0)] = \lim_{k\in\mathcal{K}} \frac{\tau_k}{2} ||A(X_k)||_F^2 = 0. \tag{38}$$

On the other hand, by contradiction, let us suppose that $\nabla\mathcal{F}(X_*) \neq 0$. Note that, this assumption implies that $||A(X_*)||_F \neq 0$. Now, from (38) and Lemma 2 we have

$$\lim_{k\in\mathcal{K}} \tau_k = \hat{\tau} = 0, \tag{39}$$

and consequently, due to the step–size τ_k is chosen by carrying out a backtracking process, then for all k greater than some $\bar{k}$, $\tau_k = \delta^{m_k}\tau_k^{ABB}$, where $m_k \in \mathbb{N}$ is the smallest positive integer number such that the non–monotone Armijo condition is fulfilled. Thus, the parameter $\bar{\tau} = \frac{\tau_k}{\delta}$ violates the non–monotone Armijo condition, i.e., it holds

$$\rho_1\bar{\tau}\mathcal{D}\mathcal{F}(X_k)[\dot{Y}(0)] < \mathcal{F}(Y(\bar{\tau})) - C_k \leq \mathcal{F}(Y(\bar{\tau})) - \mathcal{F}(X_k), \quad \forall k \geq \bar{k}. \tag{40}$$

Let's set $Z_k(\tau) := \mathcal{F} \circ Y(\tau)$, then (40) is equivalent to

$$-\frac{Z_k(\bar{\tau}) - Z_k(0)}{\bar{\tau} - 0} < -\rho_1\mathcal{D}\mathcal{F}(X_k)[\dot{Y}(0)], \quad \forall k \geq \bar{k}. \tag{41}$$

It follows from the mean value theorem, that there exists $t \in (0, \bar{\tau})$ such that $-\dot{Z}_k(t) < -\rho_1\mathcal{D}\mathcal{F}(X_k)[\dot{Y}(0)]$ for all $k \geq \bar{k}$, or equivalently

$$-Tr[\mathcal{D}\mathcal{F}(Y(t))^\top\dot{Y}(t)] < \frac{\rho_1}{2}||A(X_k)||_F^2, \quad \forall k \geq \bar{k}. \tag{42}$$

Taking limit in (42) and considering (39), we arrive at

$$Tr[G_*^\top\nabla\mathcal{F}(X_*)] \leq \frac{\rho_1}{2}||A(X_*)||_F^2, \tag{43}$$

or equivalently, $||A(X_*)||_F^2 \leq \rho_1||A(X_*)||_F^2$. Since $\rho_1 < 1$ then we have $A(X_*) = 0$, which contradicts the assumption $\nabla\mathcal{F}(X_*) \neq 0$. Therefore, X_* also satisfies the condition (9). That is, any subsequence $\{X_k\}_{k\in\mathcal{K}}$ asymptotically satisfies the KKT conditions (9)–(10). □

The following result establishes the global convergence of the Algorithm 1.

Corollary 1 *Let $\{X_k\}$ be an infinite sequence generated by Algorithm 1. Then* $\lim_{k\to\infty}||\nabla\mathcal{F}(X_k)||_F = 0$.

Proof Suppose, by contradiction, that there exists a subsequence $\{X_k\}_{k\in\mathcal{K}}$ and $\epsilon > 0$ such that $||\nabla\mathcal{F}(X_k)||_F > \epsilon$, for all $k \in \mathcal{K}$. Since $X_k \in \mathsf{St}(n,p)$ for all $k \in \mathcal{K}$ and $\mathsf{St}(n,p)$ is a compact set, then the sequence $\{X_k\}$ has an accumulation point $X_* \in \mathsf{St}(n,p)$. In addition, taking limits and considering that the Riemannian gradient $\nabla\mathcal{F}(\cdot)$ is continuous, we have $||\nabla\mathcal{F}(X_*)||_F \geq \epsilon > 0$, which contradicts Theorem 1. □

5 Numerical results.

The experiments of this section aim to show the practical usefulness of our *Implicit–SD* for eigenvalue computations, weighted orthogonal procrustes problems, the solution of heterogeneous quadratics minimization problems and the solution of nonlinear eigenvalue problems. All the numerical tests were carried out by using MATLAB running on an intel(R) CORE(TM) i7–4770, CPU 3.40 GHz with 500GB HD and 16GB RAM. To illustrate the efficiency of our algorithm, we compare our Algorithm 1, which we denote by *Implicit–SD*, with the Riemannian steepest descent method *SD* introduced by Manton in [31], the Riemannian gradient method based on the Cayley transform *OptSt* developed in [45][1], the Riemannian conjugate gradient methods *Algor.1a*, *Algor.1b* and *Algor.1b+ZH* proposed in [50][2], and also with the "manopt" toolbox [9][3]. The implementation of our algorithm is available in `http://www.optimization-online.org/DB_HTML/2020/03/7682.html`.

In our numerical tests, we consider the following stopping rules. We stop the algorithms if one of the following conditions holds: (i) $k \geq N$; (ii) $||\nabla \mathcal{F}(X_k)||_F \leq \epsilon$, where $N = 5000$ is the maximum number of iteration. We set $\epsilon = 1e$-5 for all methods. In addition, in our Algorithm 1, we set $\tau_m = 1e$-15, $\tau_M = 1e$+15, $\eta = 0.85$ and $\delta = 0.2$. For all experiments we randomly generate the starting point X_0 using the following Matlab command $[X_0, \sim] = \texttt{qr}(\texttt{randn}(n,p),0)$.

Additionally, *Time*, *Nfe*, *Nitr*, *NrmG* and *Fval* denote the averaged total computing time in seconds, the averaged number of function evaluations, the averaged number of iterations, the averaged residual $||\nabla \mathcal{F}(X^*)||_F$ where X^* is the optimum estimated by the method, respectively. In all experiment we solve 10 independent instances for the different values of (n, p) and then we report all these averages.

5.1 The Rayleigh quotient maximization

In this subsection, we consider the problem of maximizing the Rayleigh's quotient, which can be formulated as:

$$\max_{x \in \mathbb{R}^n - \{0\}} \mathcal{F}(x) = \frac{x^\top A x}{x^\top x}, \tag{44}$$

where A is a symmetric matrix with real entries. This problem appears in many applications in engineering and pattern recognition. Observe that problem (44) is equivalent to the following unitary sphere constrained optimization problem

$$\max_{x \in \mathbb{R}^n} \mathcal{F}(x) = x^\top A x \quad \text{s.t.} \quad ||x||_2 = 1, \tag{45}$$

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

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

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

which is a particular case of problem (2).

In order to illustrate the efficiency and effectiveness of the proposed algorithm on real data problems, we compare our Algorithm 1 with the Crank–Nicolson type method [45] *OptSt* and also with the Adams–Moulton type method [12] *AdamsMoulton*, computing the largest eigenvalue using the 27 instances of large scale sparse symmetric and positive definite matrices with $n \geq 4000$ taken from the UF Sparse Matrix Collection [13]. In Table 1, we report the average values *Time*, *Nitr*, *NrmG* and *Fval* required by the three methods with 5 different randomly generated starting points over the unitary sphere for this set of problems. The maximum number of iterations allowed for all algorithms was $N_{\max} = 100000$. A summary of the numerical results on 27 real data testing problems is reported in Table 1.

From Table 1, we can see that our procedure outperforms the OptSt procedure in terms of total number of iterations. In fact, the Implicit–SD method successfully performed 9 problems faster than the OptSt, while the package OptSt obtained the solution, for 3 problems, in less number of iterations than the rest of the methods. Additionally, we observe that our method was considerably competitive with the AdamsMoulton procedure. However, the AdamsMoulton was the best method in terms of number of iterations. In addition, we clearly note that our Algorithm 1 was the best method in terms of CPU–time. To illustrate the effectiveness of our approach, we adopt the performance profile [14] introduced by Dolan and More to compare the whole performance of all the methods, considering the all 135 (27×5) tests. In Figure 2, we plot the performance profile considering the number of iterations and the total computational time for all the methods. From these figure, we observe that the percentage of victory in terms of iterations for the algorithms OptSt, AdamsMoulton and Implicit–SD methods were 11.11%, 59.2% and 33.33% respectively; while in terms of total computational time the corresponding percentages were 25.92%, 33.33% and 40.74%, which suggests that the most efficient method to solve this set of problems was our Implicit–SD.

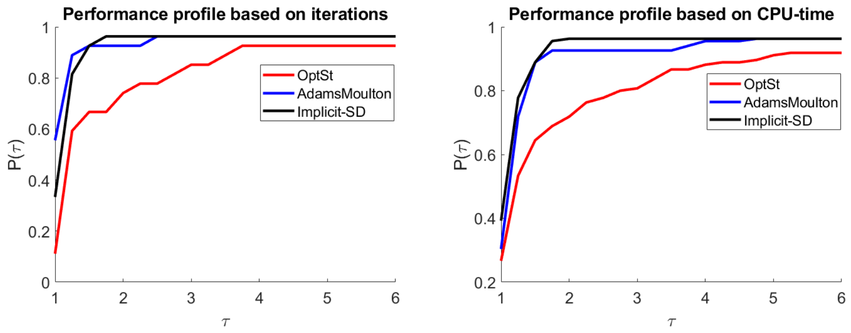
5.2 The weighted orthogonal procrustes problem

The weighted orthogonal Procrustes problem (WOPP) is a linear least–squares problem defined on a Stiefel manifold. The WOPP problem, defined by Viklands in [43] and studied in [34,39], aims to find an orthogonal transformation that reduces the distance between two matrices in terms of the Frobenius norm. Formally, for any three matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times q}$ and $C \in \mathbb{R}^{p\times q}$, the WOPP problem minimizes:

$$\min_{X\in\mathbb{R}^{n\times p}} \frac{1}{2}||AXC - B||_F^2 \quad \text{s.t.} \quad X^\top X = I_p, \tag{46}$$

Table 1 Numerical results on the Rayleigh quotient maximization.

		OptSt			AdamsMoulton			Implicit–SD		
Name	n	NrmG	Nitr	Time	NrmG	Nitr	Time	NrmG	Nitr	Time
msc04515	4515	8.88e-6	1384	0.493	9.72e-6	658	0.298	9.81e-6	1076	0.515
s1rmq4m1	5489	8.67e-6	211	0.080	9.51e-6	197	0.082	4.59e-6	196	0.077
s1rmt3m1	5489	9.99e-6	317	0.160	3.33e-6	302	0.128	9.92e-6	338	0.145
s2rmq4m1	5489	5.01e-6	208	0.075	7.85e-6	177	0.068	8.39e-6	192	0.071
s2rmt3m1	5489	9.94e-6	419	0.266	7.79e-6	220	0.072	6.62e-6	247	0.081
s3rmq4m1	5489	8.38e-6	164	0.064	8.97e-6	162	0.065	9.08e-6	218	0.083
s3rmt3m1	5489	8.55e-6	199	0.062	2.12e-6	246	0.084	4.71e-6	235	0.086
s3rmt3m3	5357	5.42e-6	215	0.069	9.59e-6	202	0.066	3.44e-6	213	0.066
fv1	9604	9.79e-6	1394	0.541	9.96e-6	1475	0.551	9.89e-6	2048	0.716
fv2	9801	8.48e-6	1272	0.489	1.00e-5	1150	0.442	9.65e-6	1307	0.464
fv3	9801	9.95e-6	1776	0.670	9.75e-6	1478	0.557	9.94e-6	1482	0.548
Kuu	7102	9.21e-6	192	0.081	8.64e-6	177	0.084	9.50e-6	191	0.094
Muu	7102	9.28e-6	17	0.005	6.32e-6	18	0.007	8.67e-6	16	0.007
bcsstk16	4884	2.46e-6	117	0.046	3.23e-6	51	0.024	7.96e-6	46	0.020
bcsstk17	10974	4.99e-6	227	0.176	5.39e-6	62	0.057	4.95e-6	67	0.056
bcsstk18	11948	1.94e-6	132	0.079	7.17e-6	50	0.035	3.53e-6	45	0.030
bcsstk28	4410	7.65e-6	26	0.009	1.66e-6	23	0.101	4.01e-6	23	0.009
bcsstk36	23052	7.05e-6	121	0.278	6.35e-6	153	0.347	9.66e-6	124	0.279
cfd1	70656	9.15e-6	110	0.535	8.47e-6	107	0.569	9.52e-6	96	0.443
ex15	6867	9.98e-6	758	0.314	9.59e-6	230	0.063	9.64e-6	274	0.067
finan512	74752	5.04e-6	84	0.277	9.48e-6	82	0.321	8.10e-6	103	0.341
mhd4800b	4800	1.97e-6	16	0.003	1.48e-6	16	0.004	5.09e-6	14	0.003
msc23052	23052	4.01e-7	163	0.384	6.21e-6	124	0.273	3.12e-6	132	0.276
nasa4704	4704	9.77e-6	9605	5.101	6.40e-6	735	0.256	1.91e-6	309	0.066
nd3k	9000	9.30e-6	162	0.550	6.55e-6	109	0.385	2.51e-6	112	0.375
pwtk	217918	9.95e-6	3919	239.969	9.94e-6	2042	111.448	6.54e-6	2124	122.373
sts4098	4098	5.97e-6	33	0.007	2.45e-6	30	0.008	6.82e-6	27	0.007



(a) Performance profile based on the number of iterations

(b) Performance profile based on computational time

Fig. 2 Performance profile based on the number of iterations and the total computational time of all the method for the Rayleigh quotient maximization.

this kind of problem is often used to find rigid body transformations that describe relationships between two objects [37].

In this subsection, we compare the numerical performance of our *Implicit–SD* procedure versus the *OptSt*, *SD* and *AdamsMoulton* methods, solving WOPPs randomly generated as follows: $m = n$, $q = p$, $A = QDR^\top$ and $C = V\Sigma V^\top$, where Q and R are random orthogonal matrices, $V \in \mathbb{R}^{p\times p}$ is a Householder matrix, $\Sigma \in \mathbb{R}^{p\times p}$ is a diagonal matrix with entries uniformly distributed in the interval $[\frac{1}{2}, 2]$ and $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix, whose diagonal elements are generated as follows: $D_{i,i} = 1 + \frac{99(i-1)}{n+1} + 2u_i$, with u_i uniformly distributed in the interval $[0, 1]$. As a initial point X_0, we generate a random matrix satisfying $X_0 \in St(n, p)$ using the following Matlab's command:

$$[X_0, \sim] = \texttt{qr}(\texttt{randn}(n, p), 0).$$

All unspecified random values are generated with the `randn` function of Matlab. In addition, the matrix $B \in \mathbb{R}^{n\times p}$ is given by $B = AX_*C$, where X_* is a random matrix belonging to the Stiefel manifold. This experiment design was taken from [34].

In Table 2, we show the average of the numerical results achieved for all the methods, from 30 simulations generated for each value of (n, p). According to the numerical results contained in Table 2, we note that all the methods obtain very similar results for problems with small p values ($p < n/2$). In particular, in this case, the most efficient methods were our Implicit–SD and the AdamsMoulton solver. On the other hand, for the cases $p > n/2$, we observe that the SD, OptSt and Implicit–SD methods show competitive results, while the AdamsMoulton method was the least efficient.

5.3 The heterogeneous quadratics minimization problem

The heterogeneous quadratics minimization problem minimizes a sum of quadratic functions over a permutation the Stiefel manifold as

$$\min_{X\in\mathbb{R}^{n\times p}} \mathcal{F}(X) = \sum_{i=1}^{p} X_{[i]}^\top A_i X_{[i]} \quad \text{s.t.} \quad X^\top X = I_p, \tag{47}$$

where $X_{[i]} \in \mathbb{R}^n$ denotes the i–th column of X and A_i's are $n\times n$ real symmetric matrices. This problem is also considered in [50]. Following [50], we build two test problems of dimension $n = 5000, 10000$ and $n = 500, 1000$ respectively, where the A_i's matrices having the following different distributions,

1. **Experiment 1**: $A_i = \texttt{diag}(\frac{(i-1)n+1}{p} : \frac{1}{p} : \frac{in}{p})$,
2. **Experiment 2**: $A_i = \texttt{diag}(\frac{(i-1)n+1}{p} : \frac{1}{p} : \frac{in}{p}) + B_i + B_i^\top$,

Table 2 Numerical results on the weighted orthogonal procrustes problem.

Methods	Nfe	NrmG	Nitr	Time
$n = 200$, $p = 200$				
OptSt	958	6.27e-6	910	3.396
SD	934	7.98e-6	896	5.354
AdamsMoulton	791	9.73e-6	757	10.653
Implicit–SD	802	9.76e-6	785	4.388
$n = 300$, $p = 200$				
OptSt	1901	8.92e-6	1846	10.074
SD	1140	9.58e-6	1111	7.298
AdamsMoulton	2702	9.79e-6	2630	33.441
Implicit–SD	2027	8.60e-6	1993	12.853
$n = 1000$, $p = 100$				
OptSt	1342	8.94e-6	1298	13.182
SD	1379	8.45e-6	1332	11.7453
AdamsMoulton	1019	9.41e-6	983	9.267
Implicit–SD	1107	9.36e-6	997	11.379
$n = 5000$, $p = 100$				
OptSt	1350	9.89e-6	1306	166.437
SD	1236	7.69e-6	1201	138.295
AdamsMoulton	1040	8.88e-6	1006	136.121
Implicit–SD	1001	9.43e-6	971	128.920

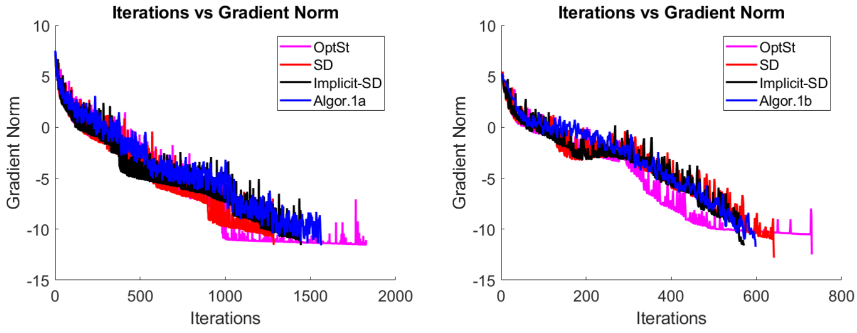
where B_i's were random matrices generated using the Matlab command $B_i = 0.1\texttt{randn}(n)$.

In our implementation, all the parameters of each solver were set to be their default values except the termination criteria of the Riemannian conjugate gradient methods presented in [50] were changed to the ones exposed at the beginning of this section. Table 3 collects the numerical results related to these experiments. From the results, we see that our Implicit–SD performs better than the rest of solvers in terms of the CPU time and the number of iterations. Additionally, we observe that all methods achieve the required accuracy for the gradient norm $||\nabla\mathcal{F}(X_k)||_F$. Furthermore, we clearly note that our procedure are more efficient than the three versions of the Riemannian conjugate gradient methods.

In Figure 3, we present the average convergence history of all the methods for Experiment 1 with $(n, p) = (10000, 10)$ and Experiment 2 with $(n, p) = (1000, 10)$. Figure 3 shows that our algorithm can achieve the required tolerance in fewer iterations than the other methods. However, all the methods show a very similar average behavior.

5.4 Total energy minimization problem.

In this subsection, we consider a class of nonlinear eigenvalue problem, which can be regarded as a simplified version of the Hartree–Fock (HF) total energy



(a) Experiment 1 with $(n, p) = (10000, 10)$ (b) Experiment 2 with $(n, p) = (1000, 10)$

Fig. 3 Convergence behaviour of Implicit–SD, SD, OptSt and the best CG method, from the same initial point, for the minimization of the heterogeneous quadratics function.

Table 3 Numerical results on the heterogeneous quadratics minimization problems.

Methods	Nfe	NrmG	Fval	Nitr	Time
Experiment 1 (47): $n = 1000$, $p = 500$					
OptSt	241	7.84e-6	2.50e+5	222	19.7
SD	273	6.33e-6	2.50e+5	254	33.0
Implicit–SD	251	8.25e-6	2.50e+5	238	23.1
Algor.1a(CG)	254	5.72e-6	2.50e+5	233	36.3
Algor.1b(CG)	226	8.09e-6	2.50e+5	199	28.1
Algor.1b(CG+ZH)	236	6.07e-6	2.50e+5	231	31.0
Experiment 1 (47): $n = 10000$, $p = 10$					
OptSt	2473	8.90e-6	4.50e+4	1732	6.7
SD	1368	8.42e-6	4.50e+4	1235	5.1
Implicit–SD	1531	9.37e-6	4.50e+4	1448	5.9
Algor.1a(CG)	2463	9.20e-6	4.50e+4	1538	8.5
Algor.1b(CG)	2615	8.17e-6	4.50e+4	1636	8.9
Algor.1b(CG+ZH)	2225	9.15e-6	4.50e+4	1611	8.3
Experiment 2 (47): $n = 500$, $p = 100$					
OptSt	2381	9.92e-6	2.43e+4	1470	195.8
SD	1666	7.93e-6	2.43e+4	1606	142.8
Implicit–SD	1340	8.69e-6	2.43e+4	1300	110.8
Algor.1a(CG)	1858	8.32e-6	2.43e+4	1172	152.5
Algor.1b(CG)	1653	7.65e-6	2.43e+4	1070	135.3
Algor.1b(CG+ZH)	1571	7.68e-6	2.43e+4	1144	127.6
Experiment 2 (47): $n = 1000$, $p = 10$					
OptSt	1217	9.82e-6	4.49e+3	729	57.4
SD	684	8.66e-6	4.49e+3	640	31.5
Implicit–SD	604	9.11e-6	4.49e+3	570	27.0
Algor.1a(CG)	1085	9.38e-6	4.49e+3	670	49.0
Algor.1b(CG)	953	8.67e-6	4.49e+3	597	41.7
Algor.1b(CG+ZH)	1198	7.78e-6	4.49e+3	883	52.4

minimization problem. Specifically we consider the following problem

$$\min_{X\in\mathbb{R}^{n\times p}} E_{total}(X) = \frac{1}{2}Tr[X^\top LX] + \frac{\mu}{4}\rho(X)^\top L^\dagger \rho(X) \quad s.t. \quad X^\top X = I_p \tag{48}$$

where L is a discrete Laplacian operator, $\mu > 0$ is a constant, $L^\dagger$ denote the Moore–Penrose generalized inverse of L and $\rho(X) := diag(XX^\top)$ is the vector containing the diagonal elements of the matrix $XX^\top$. The problem (48) appears in electronic structure calculations (see for details [33,45,49]). Note that the first–order optimality conditions for the total energy minimization problem (48) are given by:

$$H(X)X - X\Lambda = 0 \tag{49}$$

$$X^\top X = I_p, \tag{50}$$

where $H(X) := L + \mu Diag(L^\dagger \rho(X))$ and Λ is the Lagrange multipliers matrix. Here, the symbol $Diag(x)$ is a diagonal matrix with a vector x on its diagonal. Observe that the equations (49)–(50) can be seen as a nonlinear eigenvalue problem due to the matrix $H(X)$ is not constant.

The Experiment 3, described below, is similar to the test considered in [33, 49], we replay the experiments over ten independent instances with different initial points, moreover, we use a maximum number of iterations $N = 15000$ and tolerance of $\epsilon = $ 1e-5. In order to show the efficacy of our methods solving the problem (48), we present the numerical results considering different choices of n and p, while setting $\mu = 1$. In this subsection, we compare our Algorithm 1 with the *OptSt* solver, and with the following solvers contained in the “manopt” toolbox: the Riemannian conjugate gradient method (CG), the Riemannian trust–region method (Trust–Reg), proposed in [24] and also with the Riemannian BFGS method (BFGS).

Experiment 3: We consider the nonlinear eigenvalue problem for $\mu = 1$, and varying $n = 200, 300,\ 400, 1000, 5000, 10000$ and $p = 10, 100, 150, 200$.

In all instances, the L matrix is built as the one–dimensional discrete Laplacian with 2 on the diagonal, and 1 on the sub–diagonal and the super–diagonal. The numerical results are shown in Table 4. From Table 4, we observe that our proposal, the OptSt and the trust region method converge faster than the rest of methods when p approaches n. In addition, in these cases (when $p \approx n$ or $p = n/2$), the most efficient method was the Trust–Reg approach. However, if $p << n$ then the rest of methods outperform the Trust–Reg method in terms of CPU–time. Additionally, we can see that when $p << n$ the two most efficient methods were the OptSt and the Implicit–SD.

6 Concluding remarks.

In this article, we have proposed an implicit steepest descent procedure for Stiefel manifold constrained optimization problems. In particular, our pro-

Table 4 Numerical results for Experiment 3.

Method	Nitr	Time	NrmG	Nitr	Time	NrmG
	$(n, p, \mu) = (200, 150, 1)$			$(n, p, \mu) = (300, 200, 1)$		
CG	736	7.024	1.61e-6	945	15.866	4.27e-6
OptSt	1010	2.763	8.38e-6	1274	8.265	7.49e-6
BFGS	534	16.735	7.79e-6	689	39.139	2.10e-6
Trust–Reg	12	3.266	8.92e-10	10	7.048	1.48e-9
Implicit–SD	938	2.613	8.72e-6	1369	9.616	8.09e-6
	$(n, p, \mu) = (400, 200, 1)$			$(n, p, \mu) = (1000, 100, 1)$		
CG	800	25.024	3.02e-6	558	12.236	4.33e-6
OptSt	1430	36.320	9.83e-6	870	14.219	8.57e-6
BFGS	693	57.046	1.67e-6	396	25.402	3.52e-6
Trust–Reg	10	11.884	1.59e-9	12	13.525	3.51e-8
Implicit–SD	1103	33.212	8.66e-6	668	9.201	9.28e-6
	$(n, p, \mu) = (5000, 10, 1)$			$(n, p, \mu) = (10000, 10, 1)$		
CG	63	5.226	5.06e-7	63	16.9387	5.05e-7
OptSt	58	4.177	8.83e-6	59	16.0531	4.04e-6
BFGS	63	5.649	3.40e-7	63	17.645	3.38e-7
Trust–Reg	9	13.175	5.90e-9	9	46.606	5.94e-9
Implicit–SD	56	4.151	4.46e-6	57	14.7851	6.91e-6

posal can be seen as a modified version of the Manton's method [31] that incorporates implicit information based on an approximation of the Riemannian gradient of the cost function. In order to accelerate the convergence of our algorithm, we have adopted the non–monotone globalization technique developed by Zhang and Hager [48], combined with the Barzilai–Borwein's step–sizes. The global convergence is analyzed by extending the idea of the demonstration presented in [48] to the Riemannian optimization context.

Our numerical results indicate that the new approach is suitable for solving large–scale and sparse, as well as small and dense, optimization problems with orthogonality constraints.

Acknowledgements The author thanks the two anonymous referees, whose suggestions and comments greatly improved the quality of this paper.

References

1. Abrudan, T., Eriksson, J., Koivunen, V.: Conjugate gradient algorithm for optimization under unitary matrix constraint. Signal Processing **89**(9), 1704–1714 (2009)
2. Abrudan, T.E., Eriksson, J., Koivunen, V.: Steepest descent algorithms for optimization under unitary matrix constraint. IEEE Transactions on Signal Processing **56**(3), 1134–1147 (2008)
3. Absil, P.A., Gallivan, K.A.: Joint diagonalization on the oblique manifold for independent component analysis. In: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 5, pp. V–V. IEEE (2006)
4. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press (2009)

5. Baker, C.G., Absil, P.A., Gallivan, K.A.: An implicit trust-region method on riemannian manifolds. IMA journal of numerical analysis **28**(4), 665–689 (2008)
6. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA journal of numerical analysis **8**(1), 141–148 (1988)
7. Boufounos, P.T., Baraniuk, R.G.: 1-bit compressive sensing. In: 2008 42nd Annual Conference on Information Sciences and Systems, pp. 16–21. IEEE (2008)
8. Boumal, N., Absil, P.a.: Rtrmc: A riemannian trust-region method for low-rank matrix completion. In: Advances in neural information processing systems, pp. 406–414 (2011)
9. Boumal, N., Mishra, B., Absil, P.A., Sepulchre, R.: Manopt, a matlab toolbox for optimization on manifolds. The Journal of Machine Learning Research **15**(1), 1455–1459 (2014)
10. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. Foundations of Computational mathematics **9**(6), 717 (2009)
11. Dalmau, O., Oviedo, H.: A projection method for optimization problems on the stiefel manifold. In: Mexican Conference on Pattern Recognition, pp. 84–93. Springer (2017)
12. Dalmau, O., Oviedo, H.: Projected nonmonotone search methods for optimization with orthogonality constraints. Computational and Applied Mathematics **37**(3), 3118–3144 (2018)
13. Davis, T.A., Hu, Y.: The university of florida sparse matrix collection. ACM Transactions on Mathematical Software (TOMS) **38**(1), 1 (2011)
14. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical programming **91**(2), 201–213 (2002)
15. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM journal on Matrix Analysis and Applications **20**(2), 303–353 (1998)
16. Eldén, L.: Matrix methods in data mining and pattern recognition, vol. 4. SIAM (2007)
17. Feng, X., Yu, W., Li, Y.: Faster matrix completion using randomized svd. In: 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 608–615. IEEE (2018)
18. Francisco, J., Martini, T.: Spectral projected gradient method for the procrustes problem. TEMA (São Carlos) **15**(1), 83–96 (2014)
19. Gao, B., Liu, X., Yuan, Y.x.: Parallelizable algorithms for optimization problems with orthogonality constraints. SIAM Journal on Scientific Computing **41**(3), A1949–A1983 (2019)
20. Gao, B., Son, N.T., Absil, P.A., Stykel, T.: Riemannian optimization on the symplectic stiefel manifold. arXiv preprint arXiv:2006.15226 (2020)
21. Goldfarb, D., Wen, Z., Yin, W.: A curvilinear search method for p-harmonic flows on spheres. SIAM Journal on Imaging Sciences **2**(1), 84–109 (2009)
22. Hairer, E., Lubich, C., Wanner, G.: Geometric numerical integration: structure-preserving algorithms for ordinary differential equations, vol. 31. Springer Science & Business Media (2006)
23. Hu, J., Liu, X., Wen, Z.W., Yuan, Y.X.: A brief introduction to manifold optimization. Journal of the Operations Research Society of China **8**(2), 199–248 (2020)
24. Iannazzo, B., Porcelli, M.: The riemannian barzilai–borwein method with nonmonotone line search and the matrix geometric mean computation. IMA Journal of Numerical Analysis **38**(1), 495–517 (2017)
25. Jiang, B., Dai, Y.H.: A framework of constraint preserving update schemes for optimization on stiefel manifold. Mathematical Programming **153**(2), 535–575 (2015)
26. Journée, M., Nesterov, Y., Richtárik, P., Sepulchre, R.: Generalized power method for sparse principal component analysis. Journal of Machine Learning Research **11**(Feb), 517–553 (2010)
27. Kokiopoulou, E., Chen, J., Saad, Y.: Trace optimization and eigenproblems in dimension reduction methods. Numerical Linear Algebra with Applications **18**(3), 565–602 (2011)
28. Lai, R., Osher, S.: A splitting method for orthogonality constrained problems. Journal of Scientific Computing **58**(2), 431–449 (2014)
29. Lara, H., Oviedo, H., Yuan, J.: Matrix completion via a low rank factorization model and an augmented lagrangean succesive overrelaxation algorithm. Bulletin of Computational Applied Mathematics **2**(2) (2014)

30. Liu, Y.F., Dai, Y.H., Luo, Z.Q.: On the complexity of leakage interference minimization for interference alignment. In: 2011 IEEE 12th international workshop on signal processing advances in wireless communications, pp. 471–475. IEEE (2011)
31. Manton, J.H.: Optimization algorithms exploiting unitary constraints. IEEE Transactions on Signal Processing **50**(3), 635–650 (2002)
32. Oviedo, H., Dalmau, O., Lara, H.: Two adaptive scaled gradient projection methods for stiefel manifold constrained optimization. Numerical Algorithms pp. 1–21 (2020)
33. Oviedo, H., Lara, H.: A riemannian conjugate gradient algorithm with implicit vector transport for optimization in the stiefel manifold. Tech. rep., Technical report. UFSC-Blumenau, CIMAT (2018)
34. Oviedo, H., Lara, H., Dalmau, O.: A non-monotone linear search algorithm with mixed direction on stiefel manifold. Optimization Methods and Software **34**(2), 437–457 (2019)
35. Oviedo, H.F.: A spectral gradient projection method for the positive semi-definite procrustes problem. CoRR **abs/1908.06497** (2019)
36. Pietersz 4, R., Groenen, P.J.: Rank reduction of correlation matrices by majorization. Quantitative Finance **4**(6), 649–662 (2004)
37. Ravindra, V., Nassar, H., Gleich, D.F., Grama, A.: Rigid graph alignment. In: International Conference on Complex Networks and Their Applications, pp. 621–632. Springer (2019)
38. Sato, H.: Riemannian newton's method for joint diagonalization on the stiefel manifold with application to ica. arXiv preprint arXiv:1403.8064 (2014)
39. Schönemann, P.H.: A generalized solution of the orthogonal procrustes problem. Psychometrika **31**(1), 1–10 (1966)
40. Smith, S.T.: Optimization techniques on riemannian manifolds. Fields institute communications **3**(3), 113–135 (1994)
41. Stiefel, E.: Richtungsfelder und fernparallelismus in n-dimensionalen mannigfaltigkeiten. Commentarii Mathematici Helvetici **8**(1), 305–353 (1935)
42. Urdaneta, H.L., Leon, H.F.O.: Solving joint diagonalization problems via a riemannian conjugate gradient method in stiefel manifold. In: Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, volume=6, number=2, year=2018
43. Viklands, T.: Algorithms for the weighted orthogonal procrustes problem and other least squares problems. Ph.D. thesis, Datavetenskap (2006)
44. Wen, Z., Yang, C., Liu, X., Zhang, Y.: Trace-penalty minimization for large-scale eigenspace computation. Journal of Scientific Computing **66**(3), 1175–1203 (2016)
45. Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Mathematical Programming **142**(1-2), 397–434 (2013)
46. Wen, Z., Yin, W., Zhang, Y.: Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Mathematical Programming Computation **4**(4), 333–361 (2012)
47. Yu, Y., Fang, C., Liao, Z.: Piecewise flat embedding for image segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1368–1376 (2015)
48. Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM journal on Optimization **14**(4), 1043–1056 (2004)
49. Zhao, Z., Bai, Z.J., Jin, X.Q.: A riemannian newton algorithm for nonlinear eigenvalue problems. SIAM Journal on Matrix Analysis and Applications **36**(2), 752–774 (2015)
50. Zhu, X.: A riemannian conjugate gradient method for optimization on the stiefel manifold. Computational Optimization and Applications **67**(1), 73–110 (2017)
51. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. Journal of computational and graphical statistics **15**(2), 265–286 (2006)