A non-monotone Inexact Restoration approach for minimization with orthogonality constraints

Juliano B. Francisco* Douglas S. Gonçalves† Fermín S. V. Bazán‡
Lila L. T. Paredes§

Federal University of Santa Catarina. Department of Mathematics.
88040-900, Florianópolis, SC, Brazil.

October 16, 2018

Abstract

In this work we consider the problem of minimizing a differentiable functional restricted to the set of $n \times p$ matrices with orthonormal columns. This problem appears in several fields such as statistics, signal processing, global positioning system, machine learning, physics, chemistry and others. We present an algorithm based on a recent non-monotone variation of the inexact restoration method for nonlinear programming along with its implementation details. We give a simple characterization of the set of tangent directions (with respect to the orthogonality constraints) and we use it for dealing with the minimization (tangent) phase. For the restoration phase we employ the well-known Cayley transform for bringing the computed point back to the feasible set (i.e., the restoration phase is exact). Under standard assumptions we prove that any limit point of the sequence generated by the algorithm is a stationary point. A numerical comparison with a well established algorithm is also presented on three different classes of the problem.

Key words: Inexact Restoration Orthogonality constraints Stiefel manifold Cayley transform Conjugate Gradient

AMS classification: 49Q99 65K05 90C22 90C26 90C27 90C30

1 Introduction

In this paper we consider the problem:

$$\begin{aligned} \min_{X \in \mathbb{R}^{n \times p}} \quad & f(X) \\ \text{s.t.} \quad & X^T X = I_p, \end{aligned} \tag{1}$$

*juliano@mtm.ufsc.br
†douglas@mtm.ufsc.br
‡fermin@mtm.ufsc.br
§lila@mtm.ufsc.br

where $f : \mathbb{R}^{n\times p} \to \mathbb{R}$ $(p \leq n)$ is a twice continuously differentiable function and I_p denotes the $p \times p$ identity matrix. Sometimes this problem is referred in the literature as minimization on Stiefel manifold [1, 7, 9, 16]. Although our scheme encompasses the general case, we are particularly interested in medium/large scale problems with $p \ll n$. Among a wide variety of applications that can be considered in this context we mention, Kohn-Sham/Hartree-Fock energy minimization problems [14, 22], nearest low-rank correlation matrix problems [26], linear/non-linear eigenvalue problems [13, 25], and sparse principal component analysis (PCA) [17] (see [1] for further applications and algorithms).

The difficulty with Problem (1) is that, in general, neither the feasible set nor the objective function are convex. Therefore, with the exception of a few cases, finding a global minimizer (or sometimes even a stationary point) is a challenging and computationally expensive task. It is precisely because of this difficulty and because the vast number of applications that the problem has attracted the attention of many researchers and consequently several algorithms have been proposed [7, 16, 23, 25]. For an excellent survey on gradient-based methods as well as geometric properties of Problem (1) the reader is referred to [1]. Most of the algorithms in the literature generate a feasible sequence of iterates either by moving along a geodesic path or by projecting a trial point onto the feasible set. Both strategies are combined with some backtracking-like scheme in order to obtain a sufficient decrease (monotone or not) in the objective function.

We propose an algorithm that can be regarded as a geodesic-like numerical scheme combined with a non-monotone line-search along a suitable tangent descent direction, such that instead of a curvilinear search on the manifold of feasible matrices, a backtracking is performed in the tangent subset and then the obtained trial point is restored back to feasibility through a Cayley transform as done in [23] and clarified ahead. In this way, our proposal can be seen as a specialized version of the method introduced in [8], a non-monotonous variation of the inexact restoration method (IRM) [12, 19] (introduced first by Martínez and Pilotta[19] and revisited after by Fischer and Friedlander), for solving Problem (1). In brief, the IRM consists of two phases: given an iterate X_k, in the (inexact) *restoration phase*, we look for a Y_k which is, in some sense, more feasible than X_k and whose objective value $f(Y_k)$ is not too worse than $f(X_k)$. Then, by considering the null-space of the Jacobian of the constraints, in the *minimization phase* (also called tangent phase) we consider the minimization of a model for the objective function f (or for the Lagrangian) in the tangent set at Y_k in order to find a new iterate X_{k+1} that ensures a sufficient decrease of a chosen merit function. An advantage of this approach is its relative freedom in choosing the algorithms for each phase, which is interesting since in these cases it is possible to explore particular features of the minimization problem at hand. Fischer and Friedlander redesigned the IRM and established in [12] a simplified model algorithm to extend its applicability. There, convergence to a stationary point is proved under mild hypotheses. It is important to note, however, that in addition to the number of backtrackings in the line-search, the IRM needs at least two evaluations of the objective function (or of an approximation) per iteration. This can significantly increase the computational time of the overall process in some optimization problems, as occurs in many instances of Problem (1) where each objective function evaluation may demand expensive matrix algebra operations, e.g. multiplication of large dense matrices. To overcome this drawback, Francisco et. al proposed in [8] a non-monotone variation of the IRM introduced in [12].

In this work, we consider the application of the *non-monotone* inexact restoration algorithm of [8] to Problem (1), wherein the restoration phase is exact and it is accomplished by means of the Cayley transform, which in general is computationally less expensive than SVD-based

projections [16]. Briefly, for a feasible point Y, we restore the feasibility of X^+ in the tangent set of the constraints at Y (which is of the form $X^+ = (I+A)Y$ for some skew-symmetric matrix A) by using the Crank-Nicholson-based Cayley transform of [23]. Therefore, the new feasible point can be calculated as

$$Y^+ = (I - \frac{1}{2}A)^{-1}(I + \frac{1}{2}A)Y,$$

or alternatively as a by product of a *local strategy* (see Section 3) that allows to find a new feasible point with greater decrease of the objective function. This makes possible to considerably save the number of function evaluations, thus making our algorithm competitive with state of the art manifold-based gradient methods for Problem (1).

From a theoretical point of view, we provide a suitable characterization of the tangent set of the orthogonality constraints together with efficient formulas to calculate projections on it. Additionally, based on the global convergence property of the non-monotone inexact restoration algorithm of [8], we prove global convergence of the generated sequence to stationary points of (1).

Numerical experiments over different sets of problems such as linear eigenvalue problems, nonlinear eigenvalues problems and orthogonal Procrustes problems, confirm that the proposed algorithm is reliable and competitive with the efficient gradient-like method of [23] for solving the minimization problem (1).

This paper is organized as follows. Section 2 provides the necessary background for the minimization problem with orthogonality constraints, a preliminary version of the proposed algorithm, as well as its main convergence properties. Further, key concepts related to the variant of the inexact restoration method are introduced. Section 3 introduces the proposed algorithm including details of its practical implementation. Section 4 describes a set of numerical experiments carried out with our algorithm over a representative set of problems. Section 5 closes the paper with final remarks.

2 Background

We start with some notation and a few basic concepts. $\mathrm{Tr}(\cdot)$ means the trace of a matrix and, unless stated otherwise, $\langle X, Y\rangle = \mathrm{Tr}(XY^T)$ denotes the trace inner product and $\|X\|$ denotes the Frobenius norm, $\|X\| = \sqrt{\mathrm{Tr}(X^TX)}$. If W is a vector subspace, $W^\perp$ denotes its orthogonal complement. Throughout the text, I denotes the identity matrix of appropriate order, I_p denotes the identity matrix of order p and

$$\widetilde{I} := \begin{bmatrix} I_p \\ 0 \end{bmatrix} \in \mathbb{R}^{n\times p}.$$

In addition, the feasible set of Problem (1) is denoted by Γ,

$$\Gamma = \{X \in \mathbb{R}^{n\times p} \mid H(X) := X^TX - I = 0\},$$

$\mathcal{L} : \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \to \mathbb{R}$ is the associated Lagrangian,

$$\mathcal{L}(X, \Lambda) = f(X) + \mathrm{Tr}(\Lambda(X^TX - I)), \tag{2}$$

and

$$\Phi(X, \theta) = \theta f(X) + (1-\theta)\|H(X)\|$$

the corresponding merit function. Also, we say that $(\bar{X}, \bar{\Lambda})$ is a *critical pair* for (1), if $\bar{X} \in \Gamma$ and $\nabla_X \mathcal{L}(\bar{X}, \bar{\Lambda}) = 0$. In this case, $\bar{X}$ is said to be a stationary point of (1).

Furthermore, following [7] we note that

$$H(Y + tX) = (Y + tX)^T(Y + tX) - I = H(Y) + t(Y^T X + X^T Y) + t^2 X^T X.$$

Thus, by a Taylor series expansion, the null-space of $H'(Y)$ is

$$\ker(H'(Y)) = S_\Gamma(Y) := \{X \in \mathbb{R}^{n\times p} \mid X^T Y + Y^T X = 0\},$$

so that, the tangent set at $Y \in \Gamma$ is

$$\begin{aligned} T_\Gamma(Y) &= \{X \in \mathbb{R}^{n\times p} \mid Y^T(X - Y) \text{ is skew-symmetric}\} \\ &= \{X \in \mathbb{R}^{n\times p} \mid Y^T X + X^T Y = 2I\}. \end{aligned}$$

For simplicity we shall use the short notation $S_k := S_\Gamma(Y_k)$. Also, $P_{S_k}(Z)$ denotes the orthogonal projection of Z onto $S_\Gamma(Y_k)$ (w.r.t. the trace inner product).

Next we summarize the framework of our algorithm.

Let $X_k \in \mathbb{R}^{n\times p}$ be the current iterate, not necessarily feasible. In a simplified version of the non-monotone inexact restoration method, at the restoration phase it is computed a Y_k which is more feasible than X_k in the sense that:

$$\|H(Y_k)\| \leq r\|H(X_k)\|, \tag{3}$$

$$f(Y_k) - f(X_k) \leq \beta\|H(X_k)\|, \tag{4}$$

with $r \in [0, 1)$ and $\beta > 0$.

Afterwards, in the minimization phase, after computing a weight $\theta_{k+1} \leq \theta_k$ (which determines a trade-off between $f(X_k)$ and $\|H(X_k)\|$) and a descent direction $D_k \in S_k$, we determine a step size $t_k > 0$ by means of a backtracking scheme such that

$$\Phi(Y_k + t_k D_k, \theta_{k+1}) \leq T_k + \frac{1 - r}{2}(\|H(Y_k)\| - \|H(X_k)\|), \tag{5}$$

for a suitable parameter T_k. So, we update $X_{k+1} = Y_k + t_k D_k$. Fischer and Friedlander consider $T_k = \Phi(X_k, \theta_{k+1})$ in [12]. Such choice may turn (5) into a very demanding condition in terms of the number of backtrackings and function evaluations.

Instead, we follow [8] and consider

$$T_k = \max\{C_k, \Phi(X_k, \theta_{k+1})\}, \tag{6}$$

with Q_k and C_k updated at every iteration as

$$Q_{k+1} = \eta_k Q_k + 1 \tag{7}$$

$$C_{k+1} = (\eta_k Q_k T_k + \Phi(X_{k+1}, \theta_{k+1}))/Q_{k+1}, \tag{8}$$

where $Q_0 = 1$, $C_0 = \Phi(X_0, \theta_0)$ and $0 \leq \eta_{\min} \leq \eta_k \leq \eta_{\max} < 1$. According to [8], in case $\eta_k = 0$, for k sufficiently large we have $T_k = \Phi(X_k, \theta_{k+1})$. Otherwise, for $\eta_k \in (0, 1)$, the line search mentioned above turns into a non-monotone one based on the work of [24].

This non-monotone line search can be useful as it allow us accepting larger step sizes and saving function evaluations. Additionally, the global convergence of the inexact restoration

algorithm can still be ensured under mild assumptions. In the numerical experiments section we show how an adequate choice for η_k can significantly improve the overall performance of the scheme.

A preliminary version of the non-monotone inexact restoration method for Problem (1) is given in Algorithm 1. It is a particular version of the model algorithm proposed in [8]. There, the authors provide a careful analysis of the non-monotone IR algorithm and prove that it is well-defined and globally convergent under mild assumptions.

It is worthwhile mention that here we compute, at the restoration phase, $Y_k \in \Gamma$ for all k, that is, the restoration step is exact. Therefore, the requirement (3) is always fulfilled for every $r \in [0, 1)$.

Algorithm 1 Exact restoration algorithm for (1) - Preliminary Version

Step 0. Given $\mu, \beta > 0$, $0 \leq \eta_{\min} \leq \eta_{\max} < 1$, $\bar{\mu}, \theta_0 \in (0, 1)$, $r \in (0, 1]$ and $X_0 \in \mathbb{R}^{n\times p}$ such that $X_0^T X_0 = I_p$, set $Y_0 = X_0$, $C_0 = \Phi(X_0, \theta_0)$, $Q_0 = 1$ and $k = 0$.

Step 1. Set θ_{k+1} as the first term of the sequence $\{\theta_k/2^j\}_{j\in\mathbb{N}}$ satisfying

$$\Phi(Y_k, \theta_{k+1}) \leq \Phi(X_k, \theta_{k+1}) - \frac{1}{2}\|H(X_k)\|.$$

Step 2. (Minimization phase) Find $D_k \in S_k$ such that

$$\langle \nabla f(Y_k), D_k\rangle \leq -\bar{\mu}\|D_k\|^2 \quad \text{and} \quad \|D_k\| \geq \mu\|P_{S_k}(-\nabla f(Y_k))\|.$$

Step 3. Set t_k as the first term of the sequence $\{1/2^j\}_{j\in\mathbb{N}}$ such that

$$\Phi(Y_k + t_k D_k, \theta_{k+1}) \leq T_k - \frac{1-r}{2}\|H(X_k)\|,$$

where T_k comes from (6). Update $X_{k+1} = Y_k + t_k D_k$.

Step 4. (Restoration phase) Compute $Y_{k+1} \in \Gamma$ fulfilling (4). Update Q_k and C_k by (7) and (8), set $k = k + 1$ and return to **Step 1.**

Since f is a twice continuously differentiable function and Γ is compact, it follows that there exists a closed and convex set containing Γ in which f and H are Lipschitz continuous. Consequently, the following convergence result is straightforwardly obtained from [8].

Theorem 2.1. $\lim_{k\to\infty} \|D_k\| = 0$.

Proof. : First, note that

$$H(X_{k+1}) = H(X_k + t_k D_k) = H(X_k) + \frac{t_k^2}{2} D_k^T D_k.$$

Then, from [8, Theorem 3.1 and Lemma 3.3], it follows that $\{t_k\}_k$ is bounded away from zero and $\lim_{k\to\infty} \|H(X_k)\| = 0$. Hence, $\|D_k\|^2 = \mathrm{Tr}(D_k^T D_k) \to 0$. ■

Note that if we remove redundant constraints from Γ, the remaining constraints satisfy the Linear Independence Constraint Qualification (LICQ), and straightforward calculations show that every $Y \in \Gamma$ fulfills the Constant Rank Constraint Qualification (CRCQ) [15]. Let us now present the main convergence result for Algorithm 1, which is a consequence of Theorem 2.1.

Corollary 2.1. *Let $\{Y_k\}$ be the sequence generated by Algorithm 1 and $Y^* \in \Gamma$ be an accumulation point of $\{Y_k\}$. Then Y^* is a stationary point of (1).*

Proof. : Let $Y^* \in \Gamma$ be an accumulation point of (1). From Step 2 of Algorithm 1 and Theorem 2.1 we have that $P_{S_*}(\nabla f(Y^*)) = 0$, wherein S_* is the tangent set of Γ at Y^*. Hence Y^* satisfies the Approximate Gradient Projection (AGP) condition and the CRCQ. Consequently, it is a stationary point of (1) (see [20] for further details on AGP condition). ■

A well-known result concerning the local convergence rate of the inexact restoration method is presented in [2] and can be applied to our algorithm. This is the subject of the following theorem.

Theorem 2.2. *Let $\bar{Y}$ be an accumulation point of $\{Y_k\}$ (so a stationary point) with $\bar{\Lambda}$ as Lagrange multiplier. Suppose that there exists $k_0 > 0$ such that $t_k = 1$ for all $k \geq k_0$ in Algorithm 1. Besides, consider Λ_k an approximation for the Lagrange multipliers in such a way that D_k at Step 2 satisfies, for all $k \geq k_0$,*

$$\|P_{S_k}\left(\nabla \mathcal{L}(Y_k + D_k, \Lambda_k)\right)\| \leq \zeta_k \|P_{S_k}\left(\nabla \mathcal{L}(Y_k, \Lambda_k)\right)\|,$$

$$\|D_k\| + \|\Lambda_{k+1} - \Lambda_k\| \leq c\|P_{S_k}\left(\nabla \mathcal{L}(Y_k, \Lambda_k)\right)\|$$

and

$$\|X_k - Y_k\| \leq \hat{c}\|H(X_k)\|,$$

for some $c, \hat{c} > 0$ and $\{\zeta_k\} \subseteq (0, \zeta)$, with $\zeta \in [0, 1)$. Then, there exists $\epsilon > 0$ such that if $\|\Lambda_{k_0} - \bar{\Lambda}\| < \epsilon$ we have $(Y_k, \Lambda_k) \to (\bar{Y}, \bar{\Lambda})$. In addition,
(i) if $\zeta_k \to 0$, the convergence is R-superlinear;
(ii) if $\zeta = 0$, the convergence is R-quadratic.

Proof. : Follows from [2, Theorems 2.4 and 2.5]. ■

Next subsection deals with schemes for solving Step 2 and Step 4. Specifically, the following technical issues are considered: (i) how can we obtain $Y_k \in \mathbb{R}^{n \times p}$ satisfying (4)? (ii) how can we compute a "good" tangent direction $D_k \in S_k$?

2.1 Characterization of the tangent set

We begin this subsection with a smart characterization of the tangent set. It will be useful to compute the orthogonal projection onto $T_\Gamma(Y)$ (or $S_\Gamma(Y)$) as well as to obtain the main results of this work. Then, in the next subsections, we clarify how to perform the restoration and minimization phases for the Problem (1) in order to guarantee global convergence.

Theorem 2.3. *Let $Y \in \Gamma$. Then,*

$$S_\Gamma(Y) = \{AY \in \mathbb{R}^{n \times p} \mid A \in \mathbb{R}^{n \times n},\, A^T = -A\} \tag{9}$$

and

$$T_\Gamma(Y) = \{(I + A)Y \in \mathbb{R}^{n\times p} \mid A \in \mathbb{R}^{n\times n},\, A^T = -A\}. \tag{10}$$

Proof: Since $T_\Gamma(Y) = \{Y + Z \mid Z \in S_\Gamma(Y)\}$, it is sufficient to prove (9). Let us denote $\widetilde{S}(Y) = \{AY \in \mathbb{R}^{n\times p} \mid A \in \mathbb{R}^{n\times n},\, A^T = -A\}$. So, by inspection, we have $\widetilde{S}(Y) \subseteq S_\Gamma(Y)$. We now prove the equality of both sets. Let $Q \in \mathbb{R}^{n\times n}$ be an orthogonal matrix such that $Y = Q\widetilde{I}$. Then

$$\begin{aligned}
\dim(\widetilde{S}(Y)) &= \dim(\{Q^T AY \mid A \in \mathbb{R}^{n\times n},\, A^T = -A\}) \\
&= \dim(\{B\widetilde{I} \mid B \in \mathbb{R}^{n\times n},\, B^T = -B\}) \\
&= p(n-p) + (p-1)p/2.
\end{aligned}$$

Yet, we have that

$$\begin{aligned}
\dim(S_\Gamma(Y)) &= \dim(\{X \in \mathbb{R}^{n\times p} \mid X^T Q\widetilde{I} + \widetilde{I}^T Q^T X = 0\}) \\
&= \dim(\{[Z_1^T, Z_2^T]^T \in \mathbb{R}^{n\times p} \mid Z_2 \in \mathbb{R}^{(n-p)\times p},\, Z_1^T = -Z_1\}) \\
&= (n-p)p + (p-1)p/2.
\end{aligned}$$

Therefore, $S_\Gamma(Y) = \widetilde{S}(Y)$ and the result follows. ■

Corollary 2.2. *Let $Y \in \Gamma$. Then $S_\Gamma(Y)^\perp = \{YL \in \mathbb{R}^{n\times p} \mid L^T = L \in \mathbb{R}^{p\times p}\}$.*

Proof: Let L and A be symmetric and skew-symmetric matrices, respectively. Then, for $Y \in \Gamma$, trace properties imply

$$\begin{aligned}
\langle AY, YL\rangle &= \mathrm{Tr}(AYL^TY^T) = \mathrm{Tr}(AYLY^T) = \mathrm{Tr}(YL^TY^TA^T) \\
&= -\mathrm{Tr}(YL^TY^TA) = -\mathrm{Tr}(AYL^TY^T) = -\langle AY, YL\rangle,
\end{aligned}$$

that is, $\langle AY, YL\rangle = 0$ and $\{YL \in \mathbb{R}^{n\times p} \mid L^T = L \in \mathbb{R}^{p\times p}\} \subseteq S_\Gamma(Y)^\perp$. Since $\dim(\{YL \in \mathbb{R}^{n\times p} \mid L^T = L \in \mathbb{R}^{p\times p}\}) = (p^2+p)/2$, the result follows straightforwardly from Theorem 2.3. ■

Let now $Y \in \Gamma$ be a local minimizer of (1). Hence, since Y satisfies the constant rank constraint qualification, it follows that $\nabla f(Y) \in S_\Gamma(Y)^\perp$ and, from Corollary 2.2, we have

$$\nabla_Y \mathcal{L}(Y, \Lambda) = \nabla f(Y) + Y\Lambda = 0, \tag{11}$$

$$Y^TY - I_p = 0, \tag{12}$$

where the symmetric matrix $\Lambda \in \mathbb{R}^{p\times p}$ contains the Lagrange multipliers corresponding to the orthogonality constraints. Conditions (11) and (12) are the practical representations for *stationary points* of Problem (1).

From (11) and (12) one can also deduce a closed expression for the Lagrange multiplier matrix:

$$\Lambda = -\frac{\nabla f(Y)^TY + Y^T\nabla f(Y)}{2}. \tag{13}$$

Next result is relevant for the tangent phase, specifically, for minimizing a smooth model of f on the tangent set. It gives an explicit formula of the gradient of a continuous differentiable function restricted to the tangent set and thus gradient-based methods (conjugate gradient, spectral gradient and others) can be employed while thinking the tangent phase as an unconstrained minimization problem.

Proposition 2.1. *Let $\ell : \mathbb{R}^{n\times p} \to \mathbb{R}$ be a continuous differentiable function, $Y \in \Gamma$ and, for any skew-symmetric matrix $A \in \mathbb{R}^{n\times n}$ define, $\mathcal{G}_Y(A) = \ell(Y + AY)$, that is, ℓ restricted to the tangent set. Then,*

$$\nabla\mathcal{G}_Y(A) = \frac{\nabla\ell(Y + AY)Y^T - Y\nabla\ell(Y + AY)^T}{2}. \tag{14}$$

Proof: Note that defining $\mathcal{G}_Y(A)$ over the skew-symmetric matrices subspaces is equivalent to define $\mathcal{G}_Y((A - A^T)/2)$ over the vector space of $n \times n$ real matrices. Let $A \in \mathbb{R}^{n\times n}$ be a skew-symmetric matrix and Z be an arbitrary matrix. By Taylor expansion, for a $t \in \mathbb{R}$ it follows that

$$\begin{aligned}
\mathcal{G}_Y(A + \tfrac{t}{2}(Z - Z^T)) &= \ell(Y + AY + \tfrac{t}{2}(Z - Z^T)Y) \\
&= \ell(Y + AY) + t\langle\nabla\ell(Y + AY), (Z - Z^T)Y\rangle/2 + O(t^2) \\
&= \ell(Y + AY) + t\mathrm{Tr}(\nabla\ell(Y + AY)^T(Z - Z^T)Y)/2 + O(t^2) \\
&= \ell(Y + AY)+ \\
&\quad t\mathrm{Tr}((Y\nabla\ell(Y + AY)^T - \nabla\ell(Y + AY)Y^T)Z)/2 + O(t^2).
\end{aligned}$$

Then

$$\nabla\mathcal{G}_Y(A) = \frac{\nabla\ell(Y + AY)Y^T - Y\nabla\ell(Y + AY)^T}{2},$$

as claimed. ■

We list some remarks that can be obtained from Proposition 2.1:
(i) It turns out that if $Y^* \in \Gamma$ is a stationary point of (1) then $\nabla\mathcal{G}_{Y^*}(0) = 0$, therefore $\nabla f(Y^*)(Y^*)^T$ is symmetric. Also, from (11), $\nabla f(Y^*)^T Y^*$ is symmetric as well. This can be useful in measuring how far an iterate is from a stationary point.
(ii) In some applications (e.g. symmetric eigenvalue problem and the electronic structure calculation problem [11]), we have that $\nabla f(Y) = G(Y)Y$ where $G(Y)$ is a symmetric matrix. Thus, in such a case (with $\ell = f$),

$$\nabla\mathcal{G}_Y(A) = \frac{G(Y + AY)YY^T - YY^TG(Y + AY)}{2},$$

and so, if $Y^* \in \Gamma$ is a stationary point, it follows that $G(Y^*)D^* = D^*G(Y^*)$, with $D^* = Y^*(Y^*)^T$, therefore D^* is the orthogonal projector onto the invariant subspace of $G(Y^*)$.

Next subsection deals with the theoretical results related to the restoration phase of the inexact restoration method.

2.2 Restoration phase

We recall a well-known result whose proof can be found in [13]. Such a result has a strict connection with the restoration step of Algorithm 1.

Theorem 2.4. *Let $C \in \mathbb{R}^{n\times p}$ and let $C = U\Sigma V^T$ be its thin SVD decomposition (that is $\Sigma \in \mathbb{R}^{p\times p}$). Then the solution of*

$$\min \|C - X\| \quad \textit{subject to} \quad X^TX = I,$$

is given by $X_* = UV^T$.

Notice that $\|X\| \leq \sqrt{p}$, $\forall X \in \Gamma$, so the feasible set Γ is a bounded set and it is contained in a (large enough) non-empty convex compact set $\Omega \subset \mathbb{R}^{n\times p}$. Since f is twice continuously differentiable, there exists a $L_0 > 0$ such that

$$\|\nabla f(X)\| \leq L_0, \text{ for all } X \in \Omega. \tag{15}$$

In what follows, we present a condition with which requirement (4) always holds true.

Lemma 2.1. *Let $\bar{Y}_k, X_k \in \Omega$ be such that*

$$\|\bar{Y}_k - X_k\| \leq \hat{\beta}\|X_k^T X_k - I\|, \tag{16}$$

for some matrix norm $\|\cdot\|$ and $\hat{\beta} > 0$. If $Y_k \in \mathbb{R}^{n\times p}$ is such that $f(Y_k) \leq f(\bar{Y}_k)$, we have

$$f(Y_k) - f(X_k) \leq \hat{\beta} L_0 \|X_k^T X_k - I\|, \tag{17}$$

wherein $L_0 > 0$ comes from (15).

Proof: From the Taylor expansion with Lagrange remainder and (15), we have

$$f(Y_k) - f(X_k) \leq f(\bar{Y}_k) - f(X_k) \leq L_0\|\bar{Y}_k - X_k\| \leq \hat{\beta} L_0 \|X_k^T X_k - I\|.$$

■

Next, regardless the $X_k \in T_\Gamma(Y_{k-1})$, we give two practical schemes for obtaining a point $\bar{Y}_k \in \Gamma$ where (16) is satisfied for some $\hat{\beta} > 0$.

Theorem 2.5. *Let $Y_{k-1} \in \Gamma$, $X_k \in T_\Gamma(Y_{k-1})$ and Y_k solution of*

$$\min \|X_k - Z\| \quad \textit{subject to} \quad Z^T Z = I.$$

Then,

$$\|X_k - Y_k\| \leq \|(X_k)^T X_k - I\|, \tag{18}$$

that is, X_k and Y_k satisfy condition (16) with $\hat{\beta} = 1$.

Proof: Let $X_k = U_k \Sigma_k V_k^T$ be the thin SVD decomposition of X_k. Hence, from Theorem 2.4, $Y_k = U_k V_k^T$. Now, using invariance of Frobenius norm by orthogonal matrices and denoting the i-th singular value of X_k by σ_k^i, it turns out that

$$\|X_k - Y_k\|^2 = \|\Sigma_k - I_p\|^2 = \sum_{i=1}^{N} (\sigma_k^i - 1)^2.$$

Moreover, using orthogonal invariance again

$$\|(X_k)^T X_k - I_p\|^2 = \|\Sigma_k^2 - I_p\|^2 = \sum_{i=1}^{N} ((\sigma_k^i)^2 - 1)^2.$$

Since each σ_k^i is nonnegative, we have that $|\sigma_k^i - 1| \leq |(\sigma_k^i)^2 - 1|$ for all $i \in \{1, \ldots, N\}$. Thus $\sum_{i=1}^{N}(\sigma_k^i - 1)^2 \leq \sum_{i=1}^{N}((\sigma_k^i)^2 - 1)^2$ and, consequently, $\|X_k - Y_k\| \leq \|(X_k)^T X_k - I_p\|$. ■

Theorem 2.6. *Let $\bar{\beta} > 0$, $Y_{k-1} \in \Gamma$ and $X_k = Y_{k-1} + AY_{k-1} \in T_\Gamma(Y_{k-1})$, for some skew-symmetric matrix $A \in \mathbb{R}^{n\times n}$ such that $\|AY_{k-1}\|_2 \geq 1/\bar{\beta}$. Then $\bar{Y}_k = (I-\frac{1}{2}A)^{-1}(I+\frac{1}{2}A)Y_{k-1} \in \Gamma$ and $\|\bar{Y}_k - X_k\|_2 \leq \bar{\beta}\|X_k^T X_k - I\|_2$. That is, $\bar{Y}_k$ satisfies (16) with $\hat{\beta} = \bar{\beta}$.*

Proof: Since $(I - \frac{1}{2}A)(I + \frac{1}{2}A) = (I + \frac{1}{2}A)(I - \frac{1}{2}A)$ and A is skew-symmetric we have that $(I - \frac{1}{2}A)^{-1}(I + \frac{1}{2}A)$ and $(I + \frac{1}{2}A)^{-1}(I - \frac{1}{2}A)$ are orthogonal matrices, from where it results that $\bar{Y}_k = (I - \frac{1}{2}A)^{-1}(I + \frac{1}{2}A)Y_{k-1} \in \Gamma$. Also, we have that $((I - \frac{1}{2}A)^{-1}(I + \frac{1}{2}A))^{-1} = ((I + \frac{1}{2}A)^{-1}(I - \frac{1}{2}A))^T$ and thus $(I + \frac{1}{2}A)^{-1}(I - \frac{1}{2}A) = (I + \frac{1}{2}A)(I - \frac{1}{2}A)^{-1}$. Therefore, since $\|.\|_2$ is invariant by orthogonal matrices, on one hand we have

$$\begin{aligned} \|X_k - \bar{Y}_k\|_2 &= \|(I + A)Y_{k-1} - (I + \tfrac{1}{2}A)(I - \tfrac{1}{2}A)^{-1}Y_{k-1}\|_2 \\ &\leq \|(I + \tfrac{1}{2}A)Y_{k-1} - (I + \tfrac{1}{2}A)(I - \tfrac{1}{2}A)^{-1}Y_{k-1}\|_2 + \tfrac{1}{2}\|AY_{k-1}\|_2 \\ &= \|(I + \tfrac{1}{2}A)(I - (I - \tfrac{1}{2}A)^{-1})Y_{k-1}\|_2 + \tfrac{1}{2}\|AY_{k-1}\|_2 \\ &= \|(I + \tfrac{1}{2}A)(I - \tfrac{1}{2}A)^{-1}((I - \tfrac{1}{2}A) - I)Y_{k-1}\|_2 + \tfrac{1}{2}\|AY_{k-1}\|_2 \\ &= \|AY_{k-1}\|_2. \end{aligned} \tag{19}$$

On the other hand,

$$\begin{aligned} \|X_k^T X_k - I\|_2 &= \|Y_{k-1}^T(I - A)(I + A)Y_{k-1} - I\|_2 \\ &= \|Y_{k-1}^T(I - A^2)Y_{k-1} - I\|_2 \\ &= \|(AY_{k-1})^T(AY_{k-1})\|_2. \end{aligned} \tag{20}$$

Let σ_{max}^k be the largest singular value of AY_{k-1}. Then, since

$$\bar{\beta}\sigma_{max}^k = \bar{\beta}\|AY_{k-1}\|_2 \geq 1,$$

from (19) and (20) we obtain

$$\begin{aligned} \|\bar{Y}_k - X_k\|_2 &\leq \|AY_{k-1}\|_2 = \sigma_{max}^k \\ &\leq \bar{\beta}(\sigma_{max}^k)^2 = \bar{\beta}\|(AY_{k-1})^T AY_{k-1}\|_2 = \bar{\beta}\|(X_k)^T X_k - I\|_2, \end{aligned}$$

from which the result follows. ■

Theorem 2.6 provides us a scheme to restore a point from the tangent subspace to the feasible set Γ that is free of SVD computations. Furthermore, the orthogonal matrix $(I - A)^{-1}(I + A)$ obtained from a skew-symmetric A is known as Cayley Transform. This restoration approach has also been employed by Wen and Yin for solving (1) [23]. There, a gradient descent type method based on this transform is presented that results in a SVD-free scheme for minimization with orthogonality constraints. Briefly, given a feasible iterate Y_k and the particular choice $A_k = Y_k\nabla f(Y_k)^T - \nabla f(Y_k)Y_k^T$, the method devised in [23] computes

$$Y_{k+1} = Y(t_k) = (I + \frac{t_k}{2}A_k)^{-1}(I - \frac{t_k}{2}A_k)Y_k, \tag{21}$$

where $t_k > 0$ satisfies a Armijo-like sufficient decrease condition. A non-monotone Barzilai-Borwein based choice of t_k is used in order to reduce the number of steepest descent iterations as well as the number of backtrackings in the curvilinear search. For more details, the reader is referred to [23].

Theorems 2.5 and 2.6 are essential from an algorithmic point of view. Indeed, we have to assure that the procedure to bring X_k back to feasibility fulfills both conditions (3) and

(4). As mentioned before, the first requirement is automatically satisfied since we restore X_k to feasibility in a exact way (up to floating point precision). On the other hand, if we use the procedures given by Theorem 2.5 or 2.6, second requirement is assured as well, with a convenient choice for $\beta > 0$.

2.3 Minimization phase

In the minimization phase of the inexact restoration method, we need to find X_{k+1} in $T_\Gamma(Y_k)$ such that condition (5) is fulfilled. For this, given $Y_k \in \Gamma$, according to [8] it is sufficient at Step 2 of Algorithm 1 to compute $D_k \in S_\Gamma(Y_k)$ such that

$$\|D_k\| \geq \mu \|P_{S_k}(-\nabla f(Y_k))\|, \tag{22}$$

for $\mu > 0$, and

$$\langle \nabla f(Y_k), D_k \rangle \leq -\bar{\mu}\|D_k\|^2, \tag{23}$$

for $\bar{\mu} \in (0, 1]$. In other words, if $D_k \in S_k$ satisfies (22) and (23) then there exists a $t_k > 0$ such that (5) is verified at $X_{k+1} = Y_k + t_k D_k$.

Due to the characterization of $S_k = S_\Gamma(Y_k)$ (see Theorem 2.3), the orthogonal projection onto S_k can be easily computed, as describes the next proposition.

Proposition 2.2. *Let $Y \in \Gamma$ and $Z \in \mathbb{R}^{n\times p}$. Then, the skew-symmetric matrix A that minimizes $\|Z - AY\|$ is given by*

$$A^* = (I - \frac{1}{2}YY^T)ZY^T - YZ^T(I - \frac{1}{2}YY^T), \tag{24}$$

and the orthogonal projection of Z onto $S_\Gamma(Y)$ is

$$P_{S_\Gamma(Y)}(Z) = A^*Y = Z - \frac{1}{2}Y(Z^TY + Y^TZ).$$

Proof: Notice that Y can be expressed as

$$Y = Q\widetilde{I} = [Y\ Y^\perp] \begin{bmatrix} I_p \\ 0 \end{bmatrix},$$

where $Y^\perp$ is a matrix whose columns form an orthonormal basis for $\mathcal{R}(Y)^\perp$, so that $YY^T + (Y^\perp)(Y^\perp)^T = I$. From the invariance of Frobenius norm by orthogonal transformations

$$\|A^*Y - Z\|^2 = \|\bar{A}\widetilde{I} - Q^TZ\|^2 = \|\bar{A}_{11} - Y^TZ\|^2 + \|\bar{A}_{12} - (Y^\perp)^TZ\|^2, \tag{25}$$

where

$$\bar{A} = Q^TA^*Q = \begin{bmatrix} \bar{A}_{11} & -\bar{A}_{12}^T \\ \bar{A}_{12} & \bar{A}_{22} \end{bmatrix},$$

with $\bar{A}_{11}^T = -\bar{A}_{11}$. Expression (25) is minimized by $\bar{A}_{11}^* = \frac{1}{2}(Y^TZ - Z^TY)$, $\bar{A}_{12}^* = (Y^\perp)^TZ$ and by an arbitrary $\bar{A}_{22} \in \mathbb{R}^{(n-p)\times(n-p)}$; we choose $\bar{A}_{22}^* = 0$. Then

$$A^* = Q\bar{A}^*Q^T = (I - \frac{1}{2}YY^T)ZY^T - YZ^T(I - \frac{1}{2}YY^T),$$

and therefore

$$P_{S_\Gamma(Y)}(Z) = A^*Y = Z - \frac{1}{2}Y(Z^TY + Y^TZ).$$

■

Proposition 2.3. *Let $Y \in \Gamma$ and consider A^* as in (24) with $Z = D \in S_\Gamma(Y)$. Denote*

$$\mathcal{B}(t) = (I - \frac{t}{2}A^*)^{-1}(I + \frac{t}{2}A^*)Y.$$

Then, $\mathcal{B}'(0) = D$. Besides,

$$\mathcal{B}(t) = Y + tU(I - \frac{t}{2}V^TU)^{-1}V^TY, \tag{26}$$

wherein $U = [(I - \frac{1}{2}YY^T)D \mid Y] \in \mathbb{R}^{n\times 2p}$, $V = [Y \mid -(I - \frac{1}{2}YY^T)D] \in \mathbb{R}^{n\times 2p}$.

Proof: First, note that $(I - t/2A^*)\mathcal{B}(t) = (I + t/2A^*)Y$. So, by the chain rule, $(I - t/2A^*)\mathcal{B}'(t) = 1/2A^*(Y + \mathcal{B}(t)) = 1/2A^*(I + (I - \frac{t}{2}A^*)^{-1}(I + \frac{t}{2}A^*))Y = A^*(I - t/2A^*)^{-1}Y$. Then

$$\mathcal{B}'(t) = (I - \frac{t}{2}A^*)^{-1}A^*(I - \frac{t}{2}A^*)^{-1}Y$$

and $\mathcal{B}'(0) = AY$. Now, since $D \in S_\Gamma(Y)$, it follows from Theorem 2.3 and Proposition 2.2 that $D = AY$ and the first part is proved. For the second part, note from (24) that $A = UV^T$ with $Z = D$. Statement (26) follows by applying the Sherman-Morrison-Woodbury formula [13]:

$$(I - \frac{t}{2}UV^T)^{-1} = I + \frac{t}{2}U(I - \frac{t}{2}V^TU)^{-1}V^T$$

and by using that

$$I + (I - \frac{t}{2}V^TU)^{-1} + \frac{t}{2}(I - \frac{t}{2}V^TU)^{-1}V^TU = 2(I - \frac{t}{2}V^TU)^{-1}.$$

■

From Proposition 2.3 we conclude that if $D \in S_\Gamma(Y)$ is such that $\langle \nabla f(Y), D\rangle < 0$ then

$$\langle \nabla f(Y), \mathcal{B}'(0)\rangle = \langle \nabla f(Y), D\rangle < 0,$$

and so $\mathcal{B}(t)$ is a descent curve for f in Γ, for $t > 0$. Also, Equation (26) can be applied when $p < n/2$ in order to reduce computational effort for the case of the restoration via Cayley transform (21) with

$$A_k = (I - \frac{1}{2}Y_kY_k^T)D_kY_k^T - Y_kD_k^T(I - \frac{1}{2}Y_kY_k^T)$$

and $D_k \in S_k$ a search direction such that $\langle \nabla f(Y_k), D_k\rangle < 0$.

2.3.1 Basic scheme based on Spectral Projected Gradient

A possibility to choose a search direction D_k satisfying (22) and (23) is to use the projection of a positive multiple of the negative gradient onto S_k, namely

$$D_k^S := P_{S_k}(-\frac{1}{\alpha_k}\nabla f(Y_k)) = -\frac{1}{\alpha_k}\left(\nabla f(Y_k) - \frac{1}{2}Y_k[\nabla f(Y_k)^T Y_k + Y_k^T \nabla f(Y_k)]\right),$$

where α_k is a positive scalar based on Barzilai-Borwein spectral stepsize:

$$\alpha_k^1 = \min\left\{\max\left\{\frac{|\langle \Delta G_{k-1}, \Delta Y_{k-1}\rangle|}{\langle \Delta Y_{k-1}, \Delta Y_{k-1}\rangle}, \underline{\alpha}\right\}, \bar{\alpha}\right\}$$

or

$$\alpha_k^2 = \min\left\{\max\left\{\frac{\langle \Delta G_{k-1}, \Delta G_{k-1}\rangle}{|\langle \Delta G_{k-1}, \Delta Y_{k-1}\rangle|}, \underline{\alpha}\right\}, \bar{\alpha}\right\},$$

with $0 < \underline{\alpha} < \bar{\alpha} < +\infty$ and

$$\Delta Y_{k-1} := Y_k - Y_{k-1}, \qquad \Delta G_{k-1} := \nabla f(Y_k) - \nabla f(Y_{k-1}).$$

It is not hard to show that D_k^S fulfills both conditions (22) and (23) and that $D_k^S = 0$ if and only if Y_k is a stationary point with corresponding Lagrange multiplier

$$\Lambda_k = -\frac{\nabla f(Y_k)^T Y_k + Y_k^T \nabla f(Y_k)}{2}. \tag{27}$$

In other words, for $Y_k \in \Gamma$, condition (11) is equivalent to $P_{S_k}(\nabla f(Y_k)) = 0$. This justifies the use of $\|P_{S_k}(\nabla f(Y_k))\|$ as stationarity measure at feasible points.

2.3.2 Accelerating final convergence with Conjugate Gradient

An advantage of the inexact restoration approach is the flexibility to choose specific algorithms for the restoration and minimization phases. For example, in order to minimize f over the tangent set at a feasible point Y_k, one could employ second order methods (rather than the simpler projected gradient scheme presented in the previous section) that exhibit better local convergence properties, mainly when Y_k is close enough to a strict local minimizer of (1).

In the light of R-quadratic (or R-superlinear) convergence given in the Theorem 2.2, we consider the following subproblem

$$\begin{aligned} \min_{D} \quad & \mathcal{Q}(Y_k + D) \\ \text{s.t} \quad & D \in S_k, \end{aligned} \tag{28}$$

as an alternative to determine the search direction D_k in the tangent phase. In (28), $\mathcal{Q}(Y_k + D)$ denotes a quadratic model for the Lagrangian

$$\mathcal{L}(Y_k + D, \Lambda_k) = f(Y_k + D) + \langle \Lambda_k, H(Y_k + D)\rangle,$$

where $H(X) = X^T X - I_p$ and Λ_k is chosen as in (27). For solving (28), we adopted a constrained conjugate gradient proposed by Shariff [21] for minimizing a quadratic function subject to linear constraints. Such conjugate gradient method requires the projection onto the affine subspace

defined by the linear constraints. In our case, we have a closed form expression for the projection onto $S_\Gamma(Y_k) = \{D \in \mathbb{R}^{n\times p} : Y_k^T D + D^T Y_k = 0\}$, given in Proposition 2.2.

As we will discuss ahead, in the numerical experiments, we consider to invoke the Conjugate Gradient (CG) algorithm for computing a search direction D_k^{CG} based on (28) as soon as the stationarity measure falls below a certain threshold: $\|P_{S_k}(\nabla f(Y_k))\| < \delta_{CG}$.

Some care must be taken because we may find a direction of negative curvature inside CG or reach the maximum number of inner iterations. Moreover, the direction D_k^{CG} is used as search direction on the tangent set only if conditions (22)-(23) are verified – otherwise we use the spectral projected gradient direction.

For certain problems, where the Hessian $\nabla^2 \mathcal{Q}(X)$ has a favorable eigenvalue distribution, these CG iterations may considerably improve the final convergence.

3 The algorithm for minimization over orthogonality constraints

Section 2 gave the necessary background concerning the restoration and minimization phases of an inexact restoration algorithm applied to Problem (1). In this section, we update Algorithm 1 and summarize in Algorithm 2 the ideas discussed so far. Since this algorithm is a particular case of Algorithm 1, the convergence properties of both are the same, particularly, convergence to a stationary point of (1) is assured.

In the restoration step (Step 4), we remark that if $\|t_k D_k\| > 2/\beta$, then the restoration based on the Cayley transform is used since in this case Theorem 2.6 ensures the β-condition. Otherwise, we employ the SVD based projection onto Γ, where condition (4) holds true with $\beta = L_0$ (see Lemma 2.1 and Theorem 2.5).

In Step 2, we start by computing the scaled projected gradient direction D_k^S (see Section 2.3.1). If $\|P_{S_k}(\nabla f(Y_k))\| < \delta_{CG}$, then a certain number of conjugate gradient iterations is considered to obtain D_k^{CG} based on (28). If such direction fulfills conditions (22) and (23), then $D_k = D_k^{CG}$, otherwise $D_k = D_k^S$.

Step 5 is optional, because it asks for a feasible Y_{k+1} with objective value no greater than $f(\bar{Y}_{k+1})$, which is clearly satisfied by the choice $Y_{k+1} = \bar{Y}_{k+1}$. However, in Step 5 we can also consider a few "local iterations" that are allowed to move $\bar{Y}_{k+1}$ to another feasible point with a possibly smaller objective value. Here we consider M local iterations of the spectral projected gradient (no line search) followed by a restoration step.

4 Numerical experiments

In this section, we present some numerical experiments on three classes of Problem (1) and compare the results obtained by the proposed Algorithm 2, henceforth named `ERNM`, with those of [23]. Recall from Section 2.2, that the authors of [23] also employ the Cayley Transform (21) for a particular choice of A_k to devise a manifold based gradient method for Problem (1). The corresponding algorithm will be called `StiefelGBB`.

In `ERNM`, the iterations are stopped as soon as

$$\|P_{S(Y_k)}(-\nabla f(Y_k))\| < \varepsilon \tag{29}$$

Algorithm 2 Exact Restoration Algorithm for Problem (1)

Step 0. Given $\beta, \mu > 0$, $0 \leq \eta_{\min} \leq \eta_{\max} < 1$, $\bar{\mu}, \theta_0 \in (0,1)$, $r \in (0,1]$, and $X_0 \in \mathbb{R}^{n\times p}$ such that $X_0^T X_0 = I_p$, set $Y_0 = X_0$, $C_0 = \Phi(X_0, \theta_0)$, $Q_0 = 1$ and $k = 0$.

Step 1. Set θ_{k+1} as the first term of the sequence $\{\theta_k/2^j\}_{j\in\mathbb{N}}$ satisfying

$$\Phi(Y_k, \theta_{k+1}) \leq \Phi(X_k, \theta_{k+1}) - \frac{1}{2}\|H(X_k)\|.$$

Step 2. Find $D_k \in S_k$ such that

$$\langle \nabla f(Y_k), D_k\rangle \leq -\bar{\mu}\|D_k\|^2, \quad \text{and} \quad \|D_k\| \geq \mu \|P_{S_k}(-\nabla f(Y_k))\|.$$

Step 3. Set t_k as the first term of the sequence $\{1/2^j\}_{j\in\mathbb{N}}$ such that

$$\Phi(Y_k + t_k D_k, \theta_{k+1}) \leq T_k - \frac{1-r}{2}\|H(X_k)\|,$$

where $T_k = \max\{C_k, \Phi(X_k, \theta_{k+1})\}$. Update $X_{k+1} = Y_k + t_k D_k$. Choose $\eta_k \in [\eta_{\min}, \eta_{\max}]$ and update

$$Q_{k+1} = \eta_k Q_k + 1,$$
$$C_{k+1} = (\eta_k Q_k T_k + \Phi(X_{k+1}, \theta_{k+1}))/Q_{k+1}.$$

Step 4. (Restoration step) If $\|t_k D_k\| \geq 2/\beta$, then set

$$A_k = (I_n - \frac{1}{2}Y_k Y_k^T)D_k Y_k^T - Y_k D_k^T (I_n - \frac{1}{2}Y_k Y_k^T)$$

and update

$$\bar{Y}_{k+1} = \left(I_n - \frac{t_k}{2}A_k\right)^{-1}\left(I_n + \frac{t_k}{2}A_k\right)Y_k$$

or via (26). Otherwise, define $\bar{Y}_{k+1}$ as the orthogonal projection of X_{k+1} on Γ.

Step 5. Find a feasible Y_{k+1} such that

$$f(Y_{k+1}) \leq f(\bar{Y}_{k+1}).$$

Update $k = k + 1$ and return to **Step 1.**

i.e., when the stationarity measure is small enough. Also, in case

$$\frac{\langle \nabla f(Y_k), D_k \rangle}{\|\nabla f(Y_k)\| \|D_k\|} > -\varepsilon_X \tag{30}$$

or

$$\|X_k - X_{k-1}\| < \varepsilon_X, \text{ and } \quad |f(X_k) - f(X_{k-1})| < \varepsilon_F \tag{31}$$

for small positive tolerances $\varepsilon_X, \varepsilon_F$, the iterations are interrupted as well as when either the maximum number of iterations $\text{IT}_{\max}$ or the maximum number of function evaluations $\text{FE}_{\max}$ is reached.

We remark that `StiefelGBB` uses as stationarity measure $\|\tilde{P}_{S(Y)}(\nabla f(Y))\|$ where

$$\tilde{P}_{S(Y)}(\nabla f(Y)) = \nabla f(Y) - Y(\nabla f(Y)^T Y)$$

which is, in general, different from

$$P_{S(Y)}(\nabla f(Y)) = \nabla f(Y) - Y\left(\frac{\nabla f(Y)^T Y + Y^T \nabla f(Y)}{2}\right),$$

agreeing only at stationary points.

Unless stated otherwise, the tolerances used in the stopping criteria for `ERNM` (and also for `StiefelGBB`) are $\varepsilon = \varepsilon_{CG} = 10^{-4}$, $\varepsilon_X = 10^{-10}$, $\varepsilon_F = 10^{-10}$, $\text{FE}_{\max} = 2000$, along with the following parameters: $\text{ITCG}_{\max} = 50$, $\mu = 10^{-4}$ and $\bar{\mu} = 10^{-8}$. Concerning the parameter r, we use $r = 10^{-4}$ when D_k is obtained by CG and $r = 0.9998$ when D_k is the spectral projected gradient direction. In the non-monotone line search we use $\eta_k = \eta, \forall k$. The choice of η is discussed in the next subsection.

We also remark that we invoke CG to determine the search direction D_k whenever $\|P_{S(Y_k)}(-\nabla f(Y_k))\| \leq \delta_{CG}$, allowing at most $\text{ITCG}_{\max}$ inner iterations to achieve a tolerance ε_{CG}. We start with $\delta_{CG} = 10^{-2}$ and, in order to avoid premature CG iterations, we update this parameter according to

$$\delta_{CG} \leftarrow \max\{10^{-4}, 0.1\delta_{CG}\},$$

every time the direction does not satisfy the conditions of Step 2 in Algorithm 2.

All codes were implemented in Matlab R2016b and run in a MacBook Pro 2.4Ghz Intel Core i7, 8GB RAM.

4.1 Monotone × Non-monotone line search

In order to assess the impact of the non-monotone line search strategy in Algorithm 2, and to set a suitable value for the parameter η, we performed some experiments on a set of 48 random problems with the following features:

- Linear eigenvalue problem: 20 instances with $n \in \{500, 1000, 2000, 3000\}$ and $p \in \{10, 50, 100, 200, 300\}$;
- Nonlinear eigenvalue problem: 16 instances with $n \in \{200, 400, 800, 1000\}$ and $p \in \{10, 20, 30, 40\}$;
- Orthogonal Procrustes problem: 12 instances with $n \in \{500, 1000, 2000\}$ and $p \in \{10, 20, 50, 100\}$.

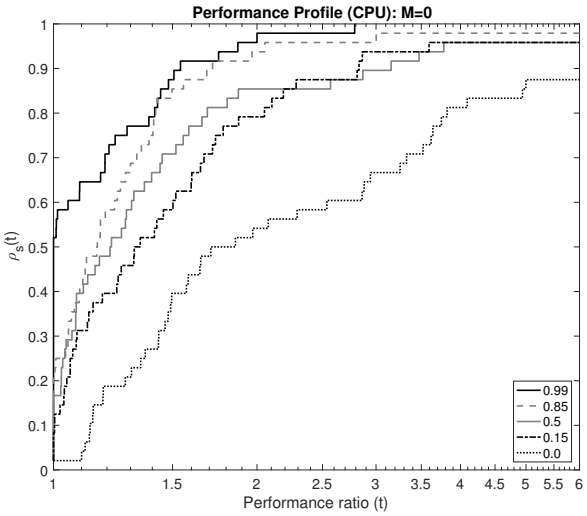

Figure 1: Performance Profile (CPU time) for different choices of η.

The definitions of such problems and the details on how the random instances were generated are presented in Sections 4.3, 4.4 and 4.5, respectively.

For this test set we limit the number of iterations to 2,000, function evaluations to 4,000 and CPU time to 300s. We consider $M = 0$ (no local iterations), and $\eta = 0, 0.15, 0.5, 0.85, 0.99$. Recall that $\eta = 0$ favors a monotone line search.

Figure 1 shows the performance profile [6] for these choices of η using the CPU time as performance measure. We can see that, for this set of problems, $\eta = 0.99$ yields the most efficient and robust version of Algorithm 2 among the considered choices of η. Henceforth, we use $\eta = 0.99$ for the following experiments.

4.2 Local $\times$ Globalized iterations

In Step 5 of Algorithm 2, we need to find a feasible Y_{k+1} such that $f(Y_{k+1}) \leq f(\bar{Y}_{k+1})$. Of course, this condition holds for $Y_{k+1} = \bar{Y}_{k+1}$ and one could just skip Step 5 and go on. However, this "local window" allows the application of heuristics in order to find another feasible point with an improved objective value and to accelerate the overall convergence while keeping the global convergence properties (see Section 2).

In these experiments we consider as such heuristic M local iterations of the spectral projected gradient (discussed in Sec. 2.3.1) – without line search ($t_k = 1$) – followed by the restoration step described in Sec. 2.2.

We have solved the same 48 problems from the previous section (using $\eta = 0.99$) for different values of M, namely $M = 0, 5, 10, 15, 20$. Figure 2 brings the performance profile in terms of CPU time. Although there is no clear winner in terms of efficiency and robustness, $M = 15$ appears to be slightly better in the majority of the problems in this test set.

In view of preliminary results obtained so far, in the numerical experiments of the next sections we consider the parameters $\eta = 0.99$ and $M = 15$ (unless stated otherwise). In the following sections, F^* denotes the known optimal value and $\hat{F}$ stands for the objective value obtained by the algorithms.

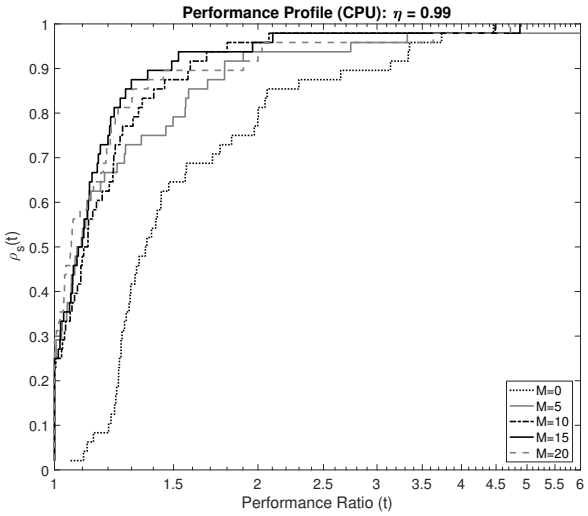

Figure 2: Performance Profile (CPU time) for different choices of M.

4.3 Linear eigenvalue problem

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ its eigenvalues. The problem of determining the p largest eigenvalues of A may be formulated as

$$\begin{aligned} \min_{X\in\mathbb{R}^{n\times p}} \quad & -\operatorname{Tr}(X^T A X) \\ \text{s.t.} \quad & X^T X = I_p, \end{aligned} \tag{32}$$

and the columns of an optimal $X^* \in \mathbb{R}^{n\times p}$ are the corresponding normalized eigenvectors.

In the following subsections we present computational results in random instances of problem (32) and on large sparse instances of the problem, respectively.

4.3.1 Random instances

In these instances, $A = B^T B$ whose entries of $B \in \mathbb{R}^{n\times n}$ are sampled from a standard normal distribution. Initial points were generated by the procedure

$$\bar{X}_0 = \texttt{randn}(n, p); X_0 = \texttt{orth}(\bar{X}_0);$$

in Matlab, which obtains a feasible X_0 from the columns of a random $n \times p$ matrix $\bar{X}_0$.

Table 1 presents the CPU time in seconds spent by `StiefelGBB` and `ERNM` to reach some stop criterion. We see that `StiefelGBB` is slightly faster than `ERNM` in almost all instances, but it fails to solve two of them in less than $\mathrm{FE}_{\max}$ function evaluations, namely, for $(n = 1000, p = 300)$ and $(n = 3000, p = 300)$. Moreover, for these two problems $|\hat{F} - F^*| \approx 10^{-4}$ for `StiefelGBB` whereas `ERNM` obtained $|\hat{F} - F^*| \approx 10^{-8}$ in all instances (here F^* is obtained by using `eigs` routine from Matlab R2016b).

Concerning the cost per iteration, we remark that each backtracking in the curvilinear search (on the manifold) of `StiefelGBB` requires to compute a Cayley transform and a function evaluation whereas in `ERNM` the line search is performed in the tangent set and requires only additional function evaluation. This difference may be significant for higher values of p.

Table 1: CPU time for `StiefelGBB` and `ERNM` in random instances of the Linear Eigenvalue Problem

	`StiefelGBB`				`ERNM`			
$p \setminus n$	500	1000	2000	3000	500	1000	2000	3000
10	0.14	0.22	1.94	2.82	0.16	0.31	2.19	5.30
50	0.42	0.65	2.50	14.97	0.52	0.86	3.28	15.03
100	1.33	2.7	15.69	16.29	1.13	3.38	13.43	19.01
200	3.38	6.18	21.21	64.09	2.93	6.05	22.67	68.20
300	8.82	117.15	72.81	430.08	8.12	**30.09**	68.08	**289.91**

Table 2: CPU time for `StiefelGBB` and `ERNM` in some sparse instances of LEP ($p = 2$).

		`StiefelGBB`			`ERNM`		
Name	n	NF	CPU	$\lvert F^* - \hat{F}\rvert$	NF	CPU	$\lvert F^* - \hat{F}\rvert$
crack	10240	402	0.4540	5.4816e-07	469	0.6470	9.9935e-09
tsyl201	20685	121	0.6096	2.9413e-09	141	1.0613	3.9037e-11
bcsstm35	30237	14	0.0879	3.1025e-08	17	0.0704	2.0224e-07
obstclae	40000	498	1.3914	4.6208e-06	980	3.1531	3.8665e-06
nasasrb	54870	149	1.0591	3.0136e-04	182	1.5082	2.8610e-06
wing	62032	408	2.4835	4.6407e-08	600	3.4285	9.7303e-09
cfd1	70656	220	1.6557	2.4863e-07	561	6.8381	6.7763e-09
thermal1	82654	144	1.3336	1.4721e-07	160	1.5036	3.8733e-09
fe_rotor	99617	149	1.8208	6.0000e-08	118	1.8073	1.1764e-09
ford2	100196	21	0.2468	2.9139e-10	33	0.4167	2.3515e-11
filter3D	106437	20	0.3141	0.0151	46	1.0375	0.0151
x104	108384	152	2.9749	1.8533e-07	218	6.2391	2.3283e-10

4.3.2 Sparse instances

We also consider a subset of sparse symmetric matrices from University of Florida sparse matrix collection [5].

In Table 2, we report the results obtained by `StiefelGBB` and `ERNM` in 12 large sparse symmetric matrices. As in [23], we consider the estimation of the $p = 2$ largest eigenvalues. Again, `StiefelGBB` was faster but the precision of `ERNM` was at least as good as the manifold algorithm. Notice also that both failed to find the global minimizer for the problem "filter3D".

4.4 Nonlinear eigenvalue problem

Consider a map $A : \mathbb{R}^{n\times p} \to \mathbb{S}^n$, where $\mathbb{S}^n$ is the set of $n \times n$ real symmetric matrices. In the nonlinear eigenvalue problem, we look for $X \in \mathbb{R}^{n\times p}$ such that

$$A(X)X = X\Lambda_p, \qquad X^T X = I_p, \tag{33}$$

where $\Lambda_p \in \mathbb{R}^{p\times p}$ is a diagonal matrix containing the p smallest eigenvalues of the symmetric matrix $A(X)$. Problem (33) encounters many interesting applications in science and engineering

[4].

Here we focus on a particular class of problem (33), namely, the total energy minimization problem [25]:

$$\begin{aligned} \min_{X\in\mathbb{R}^{n\times p}} \quad & \frac{1}{2}\mathrm{Tr}(X^TLX) + \frac{\alpha}{4}\rho(X)^TL^{-1}\rho(X) \\ \text{s.t.} \quad & X^TX = I_p, \end{aligned} \tag{34}$$

where L is a discrete Laplacian operator, $\alpha > 0$ is a constant, $\mathrm{diag}(M) = (m_{11}, m_{22}, \ldots, m_{nn})^T$, and

$$\rho(X) = \mathrm{diag}(XX^T) = (X \odot X)e,$$

where e is the column vector of ones and $\odot$ stands for the Hadamard product.

In accordance with [25], a necessary optimality condition for (34) is given by (33) with $A(X) = L + \alpha\mathrm{Diag}(L^{-1}\rho(X))$, where $\mathrm{Diag}(x)$ is a diagonal matrix with the elements of a vector x in its diagonal. Thus (34) is a particular case of (33).

The parameter $\alpha \geq 0$ controls the degree of nonlinearity in the eigenvalue problem. In Table 3, we present the results of `StiefelGBB` and `ERNM` algorithms for $p = 10$, $n \in \{200, 400, 800, 1000\}$ and increasing values of α.

As in [25], we consider random starting points generated according to the following procedure (in Matlab slang):

$$\bar{X} = \texttt{randn}(n, p),\ [U, S, V] = \texttt{svd}(\bar{X}, 0),\ \hat{X} = UV^T,\ X_0 = \texttt{eigs}(A(\hat{X}), p, \text{'sm'}).$$

The tolerances used in the stopping criteria were $\varepsilon = 10^{-4}$, $\varepsilon_X = 10^{-10}$, $\varepsilon_F = 10^{-10}$, $\mathrm{IT}_{\max} = 2000$.

For the instances of Table 3, we can observe a similar behavior for both algorithms, with `StiefelGBB` demanding less function evaluations in all the instances. However, for $\alpha = 100$, we can see in Table 4 that as p increases, `StiefelGBB` starts to face some difficulties, not achieving the desired precision before reaching the maximum number of iterations, whereas `ERNM` appears to remain stable, solving all the instances within the criterion $\|P_{S(\bar{Y})}(\bar{Y})\| < \varepsilon = 10^{-4}$.

4.5 Orthogonal Procrustes problem

Given $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times p}$, $n \geq p$, the orthogonal Procrustes problem reads

$$\begin{aligned} \min_{X\in\mathbb{R}^{n\times p}} \quad & \frac{1}{2}\|AX - B\|^2 \\ \text{s. t.} \quad & X^TX = I_p. \end{aligned} \tag{35}$$

It is interesting to mention that although this problem has a closed form solution when $n = p$, given by $X^* = UV^T$, where $A^TB = U\Sigma V^T$ is the SVD of A^TB, (1) is a non-convex optimization problem which has more than one stationary point.

In the numerical experiments of this section, besides varying the dimensions n and p, we also consider different singular value distributions for A inspired by [9, 10]. Namely, $A = U\Sigma V^T$ where U and V are random orthogonal matrices and Σ is a non-negative diagonal matrix. Then, a random $\tilde{Q} \in \mathbb{R}^{n\times p}$ with orthonormal columns is generated and $B = A\tilde{Q}$.

Initial points were generated by the procedure

$$\bar{X}_0 = X^* + 0.001\,\texttt{randn}(n, p); X_0 = \texttt{orth}(\bar{X}_0);$$

where X^* is the known solution.

Table 3: CPU time for `StiefelGBB` and `ERNM` in some instances of the Nonlinear Eigenvalue Problem ($p = 10$).

		`StiefelGBB`			`ERNM`		
α	n	NF	CPU	$\|P_{S(\bar{Y})}(\bar{Y})\|$	NF	CPU	$\|P_{S(\bar{Y})}(\bar{Y})\|$
0.01	200	92	0.03561	6.8396×10^{-5}	164	0.05131	1.7545×10^{-4}
0.01	400	129	0.0890	6.3604×10^{-5}	171	0.09423	9.4575×10^{-5}
0.01	800	100	0.2611	9.5978×10^{-5}	186	0.4559	4.9998×10^{-5}
0.01	1000	102	0.4401	8.7415×10^{-5}	239	1.000	6.8828×10^{-5}
1.00	200	48	0.02278	9.0125×10^{-5}	74	0.0284	6.5455×10^{-5}
1.00	400	50	0.03683	7.3631×10^{-5}	70	0.05672	6.8436×10^{-5}
1.00	800	52	0.1343	8.3579×10^{-5}	71	0.1791	6.2643×10^{-5}
1.00	1000	48	0.2088	9.1069×10^{-5}	71	0.2971	6.3466×10^{-5}
10	200	63	0.02394	7.7930×10^{-5}	216	0.07786	9.3466×10^{-5}
10	400	61	0.04328	7.6372×10^{-5}	225	0.1277	7.8082×10^{-5}
10	800	70	0.1825	2.8644×10^{-5}	138	0.344	4.4447×10^{-5}
10	1000	74	0.3195	4.9461×10^{-5}	142	0.5587	2.1190×10^{-5}
100	200	82	0.09359	9.2752×10^{-5}	208	0.1424	4.4285×10^{-5}
100	400	90	0.0611	8.9270×10^{-5}	250	0.1593	6.6543×10^{-5}
100	800	91	0.2323	9.4751×10^{-5}	208	0.4991	6.6366×10^{-5}
100	1000	121	0.5195	9.7985×10^{-5}	226	0.91	7.0921×10^{-5}

Table 4: CPU time for `StiefelGBB` and `ERNM` in some instances of the Nonlinear Eigenvalue Problem ($\alpha = 100$).

		`StiefelGBB`			`ERNM`		
p	n	NF	CPU	$\|P_{S(\bar{Y})}(\bar{Y})\|$	NF	CPU	$\|P_{S(\bar{Y})}(\bar{Y})\|$
10	200	82	0.03695	9.2752×10^{-5}	208	0.09656	4.4285×10^{-5}
10	400	101	0.05986	2.9693×10^{-5}	243	0.1322	4.0926×10^{-5}
10	800	83	0.2273	8.0371×10^{-5}	221	0.5746	3.8825×10^{-5}
10	1000	98	0.4717	9.1351×10^{-5}	248	1.112	8.9374×10^{-5}
20	200	9426	3.48	1.1047×10^{-3}	553	0.2642	8.2123×10^{-5}
20	400	201	0.1601	5.5251×10^{-5}	908	0.6944	5.4108×10^{-5}
20	800	9254	27.18	1.9862×10^{-3}	618	2.105	2.9033×10^{-5}
20	1000	9508	44.43	5.4834×10^{-4}	801	3.264	5.8985×10^{-5}
30	200	378	0.253	4.4016×10^{-5}	1593	1.131	9.5992×10^{-5}
30	400	371	0.4885	7.9130×10^{-5}	1191	1.654	8.6490×10^{-5}
30	800	9044	32.15	9.2117×10^{0}	1183	3.582	6.3497×10^{-5}
30	1000	2432	14.5	7.8938×10^{-3}	2098	11.72	4.7705×10^{-5}
40	200	7469	4.674	6.1828×10^{-2}	1362	1.03	5.8821×10^{-5}
40	400	555	0.958	1.0339×10^{-4}	1038	1.617	5.0187×10^{-5}
40	800	504	2.545	1.2082×10^{-4}	1234	5.125	1.6375×10^{-4}
40	1000	5019	30.85	3.2227×10^{0}	1060	6.436	8.8694×10^{-5}

Table 5: CPU time for `StiefelGBB` and `ERNM` in random instances of OPP (uniformly distributed singular values)

	`StiefelGBB`			`ERNM`		
$p \setminus n$	500	1000	2000	500	1000	2000
10	0.06	0.05	0.13	0.05	0.04	0.11
20	0.06	0.07	0.16	0.04	0.06	0.16
50	0.10	0.14	0.36	0.08	0.11	0.31
100	0.24	0.38	0.98	0.10	0.22	0.63

Table 6: CPU time for `StiefelGBB` and `ERNM` in random instances of OPP (equally spaced singular values)

	`StiefelGBB`			`ERNM`		
$p \setminus n$	500	1000	2000	500	1000	2000
10	0.06	0.18	1.54	0.11	0.27	2.03
20	0.10	0.41	3.19	0.16	0.60	3.00
50	0.27	1.23	2.98	0.39	0.93	4.47
100	0.71	**1.45**	**80.49**	0.86	**11.10**	**30.11**

4.5.1 Uniformly distributed singular values

The singular values are uniformly distributed in the interval $[10, 12]$. In this case, the matrix A is well-conditioned and we expect a good performance for both algorithms.

As expected both algorithms solve these instances very fast (see Table 5), expending around 20 function evaluations, and obtaining very good accuracy: $|\hat{F} - F^*| \approx 10^{-10}$.

4.5.2 Different equally spaced singular values

The singular values of A follow the arithmetic progression:

$$\sigma_i = 1 + \frac{i}{100}, \quad i = 1, 2, \ldots, n.$$

This implies in n different and equally spaced singular values in the interval $[1, 1 + n/100]$.

Though the condition number of A is not so high, these instances are difficult for gradient methods due to this specific eigenvalue distribution.

From Table 6 we observe that `StiefelGBB` is faster than `ERNM` in the majority of instances, except for the very last one with $n = 2000$ and $p = 100$. In fact for this instance, `StiefelGBB` exceeded the maximum number of 2,000 function evaluations, returning an objective value $\hat{F}$ such that $|\hat{F} - F^*| \approx 2.0306$. `ERNM` found the wrong stationary point for $n = 1000$ and $p = 100$, obtaining $|\hat{F} - F^*| \approx 1.8968$.

Table 7: CPU time and relative error for `StiefelGBB` in random instances of OPP (clustered singular values)

	CPU time (secs.)			Relative Error: $\lvert\hat{F} - F^*\rvert$		
$n \setminus p$	10	20	50	10	20	50
500	2.6858	6.7247	14.7839	1.3403×10^{0}	1.7920×10^{0}	0.0063×10^{0}
1000	12.2200	16.8900	32.5965	3.2388×10^{0}	5.0986×10^{0}	1.7202×10^{1}
2000	36.0926	47.3296	97.3582	8.8538×10^{0}	1.6301×10^{1}	3.8828×10^{1}

Table 8: CPU time and relative error for `ERNM` in random instances of OPP (clustered singular values)

	CPU time (secs.)			Relative Error: $\lvert\hat{F} - F^*\rvert$		
$n \setminus p$	10	20	50	10	20	50
500	0.9329	1.0058	11.4787	5.8667×10^{-10}	2.6106×10^{-9}	1.0419×10^{-9}
1000	3.2726	11.7934	16.4994	7.3380×10^{-10}	2.7454×10^{-10}	2.4048×10^{-10}
2000	20.6557	25.0054	78.1576	1.1640×10^{-10}	1.4985×10^{-9}	4.3011×10^{-10}

4.5.3 Clustered singular values

In these instances we generate the singular values of A in some clusters, according to the formulae

$$\sigma_i = 1 + 100\left(\left\lfloor \frac{i}{100} \right\rfloor\right) + \delta_i, \quad i = 1, 2, \ldots, n,$$

where δ_i are independent and identically distributed samples from a standard Gaussian $N(0, 0.1)$. Such singular value distribution implies in an ill-conditioned Hessian $A^T A$ with few eigenvalue clusters.

Since gradient-like methods perform poorly for ill-conditioned Hessians, in `StiefelGBB` we increase the maximum number of iterations to 5,000.

In `ERNM`, the "local iterations" described in Sect. 3 were disabled ($M = 0$) and we set

$$\delta_{CG} = \max\{10^{-2}, \min\{10^{2}, 10^{-3}\|P_{S(Y_0)}\nabla f(Y_0)\|\}\}$$

in order to favor early Conjugate Gradient iterations.

From Tables 7 and 8 we observe that `ERNM` with these adjustments was faster than `StiefelGBB` and also obtained better solutions. In fact, in all problems of this set, `StiefelGBB` faced convergence problems and stopped with the maximum number of iterations. On the other hand, the results of `ERNM` are justified by the good behavior of the Conjugate Gradient (used in the minimization phase) for this kind of problem.

4.6 Higher precision solutions

In this last section of numerical experiments we consider again the Linear Eigenvalue Problem (LEP) described in Section 4.3 but requiring now a higher precision, namely

$$\varepsilon = \varepsilon_{CG} = 10^{-8}, \quad \varepsilon_X = \varepsilon_F = 10^{-12}.$$

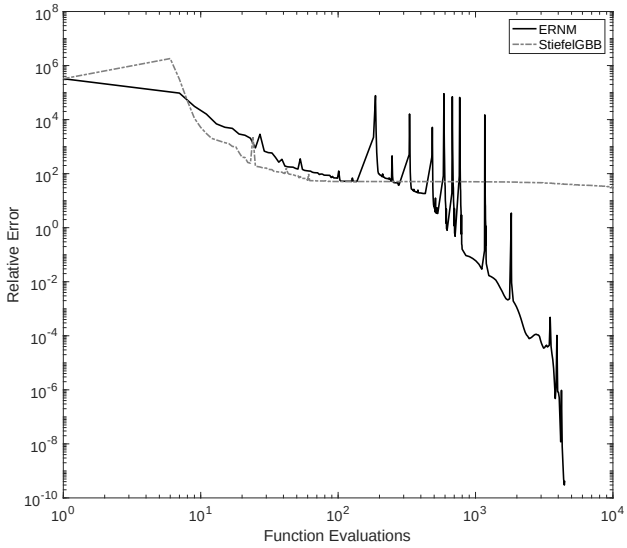

Figure 3: Relative error decay ($n = 2000$, $p = 50$).

Table 9: CPU time and optimality measure of `StiefelGBB` in LEP instances

	CPU time (secs.)			$\|P_{S(Y)}(\nabla f(Y))\|$		
$n \setminus p$	10	50	100	10	50	100
500	0.1525	4.4038	4.0377	5.3175×10^{-8}	$\mathbf{1.0835}\times10^{-3}$	1.1602×10^{-6}
1000	0.3862	10.4264	20.9483	1.0084×10^{-7}	$\mathbf{1.5551}\times10^{-2}$	$\mathbf{6.2479}\times10^{-3}$
2000	0.9369	24.2097	52.6929	2.4871×10^{-7}	$\mathbf{4.8590}\times10^{-4}$	$\mathbf{1.5396}\times10^{-3}$

With this we intend to highlight once more that the flexibility in the minimization phase of Algorithm 2 allows one to employ methods exhibiting better local convergence rates that are more suitable when high precision in the solution is paramount. Here we consider again the linearly constrained conjugate gradient of Section 2.3.2.

Table 9 presents the results obtained by `StiefelGBB`. We remark that the maximum number of 2,000 iterations was reached for five problems (whose values of $\|P_{S(Y)}(\nabla f(Y))\|$ are in bold).

From Table 10 we see that the desired precision $\varepsilon = 10^{-8}$ was also not achieved by `ERNM` in all problems, though it achieved relatively better results than `StiefelGBB` as expected. Figures in bold correspond to problems where `ERNM` stopped by criterion (31) and figures in italics correspond to the stopping criterion (30).

In Figure 3 we display how the relative error on the objective function decays as a function of the number of function evaluations spent by the algorithms, for the LEP with $n = 2000$ and $p = 50$. This significant decay of the relative error in the final stage of the process is in general observed when full CG iterations are accepted (with $t_k = 1$) from a given iteration $\bar{k}$.

Table 10: CPU time and optimality measure of `ERNM` in LEP instances

	CPU time (secs.)			$\|P_{S(Y)}(\nabla f(Y))\|$		
$n \setminus p$	10	50	100	10	50	100
500	0.1643	0.7542	1.5811	4.2003×10^{-9}	$\mathbf{7.6564}\times10^{-6}$	$\mathbf{5.3753}\times10^{-6}$
1000	0.7749	1.9112	6.3284	$\mathbf{3.7871}\times10^{-6}$	$\mathit{1.0084}\times10^{-7}$	$\mathit{6.9316}\times10^{-8}$
2000	2.0149	10.2729	14.6052	5.5677×10^{-9}	$\mathbf{7.5310}\times10^{-5}$	9.0842×10^{-9}

5 Final remarks

In this work we presented a numerical scheme for minimizing a differentiable function over the set of (rectangular) matrices with orthonormal columns. The proposed algorithm is based on a non-monotone variation of the inexact restoration method and has the advantage of allowing specific approaches for solving the two phases, namely, minimization (tangent) and restoration phases, so that the structure of the problem can be better explored. Our theoretical results guarantee convergence of the generated sequence to stationary points as well as local super-linear/quadratic convergence under reasonable conditions. In particular, we have shown that suitable numerical schemes for treating subproblems (with second-order information) lead to a substantial improvement in some classes of problems, such as problems with Hessian mismatch or those where high accuracy is required. To validate our algorithm and illustrate the theoretical results in practical problems, results of numerical experiments were reported on three different representative classes of Problem (1). The numerical results indicate that our algorithm is reliable and performs as efficiently as a state-of-the-art manifold-based gradient algorithm while endorsing the convergence theory established in Section 4, for example, global and R-quadratic (or superlinear) convergences.

In future works we plan to investigate two issues where there is room to improve in our algorithm: (1) since at the tangent phase the minimization is performed over a linear subspace, the backtracking over the merit function can be replaced by a standard interpolation scheme in order to reduce the number of function evaluations; (2) problem-oriented extrapolation schemes could be employed at the restoration phase in order to compute the restored point Y_k such that $f(Y_k) \leq f(\bar{Y}_k)$ (here called "local window").

Acknowledgements

J.B.F., D.S.G and F.S.V.B are grateful to CNPq by the financial support (Grant n. 421386/2016-9 and 308523/2017-2). L.L.T.P would like to thank to CAPES by the Ph.D. scholarship. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

References

[1] P. A. Absil, R. Mahony and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press (2008)

[2] E. G. Birgin and J. M. Martínez, Local Convergence of an Inexact-Restoration Method and Numerical Experiments, *J. Optim. Theory Appl.*, 127(2), 229–247 (2005)

[3] L. F. Bueno, G. Haeser and J. M. Martínez, A Flexible Inexact-Restoration Method for Constrained Optimization, *J. Optim. Theory Appl.*, 165, 188–208 (2015)

[4] E. Cancès, R. Chakir and Y. Maday, Numerical Analysis of Nonlinear Eigenvalue Problems, *Journal of Scientific Computing*, 45, 90–117 (2010)

[5] T. A. Davis and Y. Hu. The University of Florida Sparse Matrix Collection *ACM Transactions on Mathematical Software*, 38(1):1–25 (2011)

[6] E. D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91, 201–213 (2002)

[7] A. Edelman, T. A. Arias and S. T. Smith, The geometry of algorithms with orthogonality constraints, *SIAM J. Matrix Anal. Appl.*, 20(2), 303–353 (1998)

[8] J. B. Francisco, D. S. Gonçalves, F. S. V. Bazán and L. E. T. Paredes, Non-monotone inexact restoration method for non-linear programming, submitted (2018). Avaiable online at http://www.optimization-online.org/DB_HTML/2018/10/6858.html.

[9] J. B. Francisco and F. S. Viloche Bazán, Nonmonotone algorithm for minimization on closed sets with application to minimization on Stiefel manifolds, *J. Comp. and Appl. Math.*, 236(10), 2717–2727 (2012)

[10] J. B. Francisco, F. S. V. Bazán and M. Weber Mendonça, Non-monotone algorithm for minimization on arbitrary domains with applications to large-scale orthogonal Procrustes problem, *Appl. Num. Math.*, 112, 51–64 (2017)

[11] J. B. Francisco, J. M. Martínez, L. Martínez, F. Pisnitchenko, Inexact restoration method for minimization problems arising in electronic structure calculations, *Comput. Optim. Appl.*, 50, 555–590 (2011)

[12] A. Fischer and A. Friedlander, A new line search inexact restoration approach for nonlinear programming, *Comput. Optim. Appl.*, 46, 333–346 (2010)

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4ed., The Johns Hopkins University Press (2013)

[14] T. Helgaker, J. Jørgensen and J. Olsen, *Molecular electronic - Structure theory*, Wiley (2000)

[15] R. Janin, Directional derivative of the marginal function in non linear programming. In Sensitivity, Stability and Parametric Analysis, *Mathematical Programming Studies*, 110-126, Springer Berlin Heidelberg (1984)

[16] B. Jiang and Y. H. Dai, A Framework of Constraint Preserving Update Schemes for Optimization on Stiefel Manifold, *Mathematical Programming*, 153(2) 535-575 (2015)

[17] M. Journée, Y. Nesterov, P. Richtárik and R. Sepulchre, Generalized Power Method for Sparse Principal Component Analysis, *Journal of Machine Learning Research*, 11 517–553 (2010)

[18] Huikang Liu, Anthony Man-Cho So, Weijie Wu., Quadratic Optimization with Orthogonality Constraint: Explicit Łojasiewicz Exponent and Linear Convergence of Retraction-Based line search and Stochastic Variance-Reduced Gradient Methods. *Preprint* (2017)

[19] J. M. Martínez and E. A. Pilotta, Inexact restoration algorithm for constrained optimization, *Journal of Optimization Theory and Applications*, 104(1), 135–163 (2000)

[20] J. M. Martínez and B. F. Svaiter, A practical optimality condition without constraint qualifications for nonlinear programming, *Journal of Optimization Theory and Applications*, 118(1), 117–133 (2003)

[21] M. Shariff, A constrained conjugate gradient method and the solution of linear equations, *Computers & Mathematics with Applications*, 30(11), 25–37 (1995)

[22] W. Kohn, Nobel Lecture: Electronic structure of matter–wave functions and density functionals, *Reviews of Modern Physics*, 71(5), 1253-1266 (1999)

[23] Zaiwen Wen and Wotao Yin, A feasible method for optimization with orthogonality constraints, *Math. Program., Ser. A*, 142, 397-434 (2013)

[24] H. Zhang and W. Hager, A nonmonotone line search technique and its application to unconstrained optimization *SIAM Journal on Optimization*, 14(4), 1043–1056 (2004)

[25] Zhi Zhao, Zheng-Jian Bai, and Xiao-Qing Jin, A Riemannian Newton algorithm for nonlinear eigenvalue problems, *SIAM J. Matrix Anal. & Appl.*, 36(2) 752–774 (2015)

[26] X. Zhu, A feasible filter method for the nearest low-rank correlation matrix problem, *Numerical Algorithms*, 69 763–784 (2015)