Second order optimality and the Newton algorithm on orthogonal Stiefel manifolds

Petre Birtea, Ioan Cașu, and Dan Comănescu
Department of Mathematics, West University of Timișoara
Bd. V. Pârvan, No 4, 300223 Timișoara, România
petre.birtea@e-uvt.ro, ioan.casu@e-uvt.ro, dan.comanescu@e-uvt.ro

Abstract
The Newton algorithm is one of the main second order numerical algorithms. We give an explicit construction of the Newton algorithm on orthogonal Stiefel manifolds. We introduce a local frame appropriate for the computation of the Hessian matrix for a cost function defined on Stiefel manifolds. As an application we rediscover second order conditions for optimality for the orthonormal Procrustes problem. For a Brockett cost function defined on the orthogonal Stiefel manifold $S^4_2$ we give a classification of the critical points and we present some numerical simulations of the Newton algorithm.

MSC: 53Bxx; 65Kxx; 90Cxx; 49M15.
Keywords: Newton algorithm on manifold, optimization, second order optimality, constraint manifold, orthogonal Stiefel manifold, Procrustes problem, Brockett cost function

1 Introduction
Newton algorithm is a type of algorithm that uses the second order information about a cost function in order to characterize its critical points. For practical reasons, it is very useful to adapt this algorithm to the case of differential manifolds. There exists a rich literature that deals with construction of Newton algorithm on manifolds, see [1], [9], [12], and [16]. Optimization problems on Stiefel manifolds appear in important applications such as statistical analysis of data [15], blind signal separation [11], distance metric learning [14], among many other problems.

In Section 2 we regard an orthogonal Stiefel manifold as a preimage of a regular value for a set of constraint functions. We construct the specific elements which appear in the formula (5) for the Hessian of a cost function defined on an orthogonal Stiefel manifold. We apply this theoretical setting for the orthonormal Procrustes cost function and we obtain second order conditions for this optimization problem. Similar computations can be made in order to obtain second order conditions for the Penrose regression cost function and for the Brockett cost function (see [7] and [1]).

In order to explicitly compute the components of the Hessian matrix of a cost function, in Section 3 we introduce an explicit local frame for an orthogonal Stiefel manifold and we present some important properties of this local frame. We determine the components of the Hessian matrices of the constraint functions in this frame.

In Section 4 we present in detail the Newton algorithm in the case of an orthogonal Stiefel manifold. We first recall the setting used in [6]. The iterative scheme of Newton algorithm on Riemannian manifolds is given by:

$$x_{k+1} = \overline{R}_{x_k}(\overline{v}_{x_k}),$$

where the sequence $(x_k)_{k \in \mathbb{N}}$ belongs to a smooth Riemannian manifold $(S, g_S)$, $\overline{G} : S \to \mathbb{R}$ is a smooth cost function, $\overline{R} : TS \to S$ is a smooth retraction, and the tangent vector $\overline{v}_{x_k} \in T_{x_k}S$ is the solution of the (contravariant) Newton equation

$$\mathcal{H}\overline{G}(x_k) \cdot \overline{v}_{x_k} = -\nabla_{g_S} \overline{G}(x_k).$$

(1)
Using the link between the Hessian operator $\mathcal{H}\tilde{G} : TS \to TS$ and its associated symmetric bilinear form $\text{Hess} \tilde{G} : TS \times TS \to \mathbb{R}$, the equation (1) can be written equivalently as the (covariant) Newton equation:

$$\text{Hess} \tilde{G}(x_k) \cdot \tilde{v}_{x_k} = -d\tilde{G}(x_k). \tag{2}$$

In [6] we have solved the above equation for the case when the manifold $S$ is a constraint manifold and we have embedded it in a larger manifold $M$ (usually an Euclidean space). Let us consider an ambient smooth Riemannian manifold $(M, g)$ and a smooth constraint map $F = (F_1, ..., F_r) : M \to \mathbb{R}^r$ such that $S = F^{-1}(c)$, where $c \in \mathbb{R}^r$ is a regular value of $F$. The details of the construction are presented in the following box.

**Embedded Newton algorithm (see [6]):**

1. Consider a smooth prolongation $G : M \to \mathbb{R}$ of the cost function $\tilde{G} : S \to \mathbb{R}$.
2. Construct an adapted frame $\{b_1, ..., b_s, \nabla F_1, ..., \nabla F_r\}$, where the vector fields $b_i \in TM$ are tangent to the submanifold $S$.
3. Compute the coordinate functions
   $$g_i = dG \cdot b_i, \ i \in 1, s.$$
4. Compute the Lagrange multiplier functions, see Appendix eq. (13), $\sigma_\alpha : M \to \mathbb{R}$, for $\alpha = 1, r$.
5. Compute the components of the Hessian matrix $\text{Hess} \tilde{G}$ of the cost function $\tilde{G}$ (according to equation (2.6) in [5])
   $$h_{ij} = \left(\text{Hess} G - \sum_{\alpha=1}^r \sigma_\alpha \text{Hess} F_\alpha\right)(b_i, b_j), \ i, j \in 1, s.$$
6. Choose a retraction $\mathcal{R} : T_xM \to M$ such that for any $v_x \in T_xS \subset T_xM$ we have $\mathcal{R}_x(v_x) \in S$.
7. Input $x_0 \in S$ and $k = 0$.
8. **repeat**
   - Solve the linear system (2) with the unknowns $(v^1_{x_k}, ..., v^s_{x_k})$,
     $$\begin{cases}
     \sum_{i=1}^s h_{i1}(x_k) v^i_{x_k} = -g_1(x_k) \\
     \vdots \\
     \sum_{i=1}^s h_{is}(x_k) v^i_{x_k} = -g_s(x_k).
     \end{cases} \tag{3}$$
   - Construct the line search vector
     $$v_{x_k} = \sum_{j=1}^s v^j_{x_k} b_j(x_k).$$
   - Set $x_{k+1} = \mathcal{R}_{x_k}(v_{x_k})$.
   **until** $x_{k+1}$ sufficiently minimizes $\tilde{G}$.  

2
In the last section we exemplify the algorithm for a particular Brockett cost function on the orthogonal Stiefel manifold $St^2_p$. We give the list of all critical points and we completely characterize them. Starting from different initial points we present some numerical simulations of the algorithm and we discuss its behavior compared with the steepest descent algorithm (3).

2 Hessian matrix on orthogonal Stiefel manifolds

For $n \geq p \geq 1$, we consider the orthogonal Stiefel manifold:

$$St^n_p = \{ U \in M_{n \times p}(\mathbb{R}) \mid U^T U = I_p \}.$$  

Denote with $u_1, \ldots, u_p \in \mathbb{R}^n$ the vectors that form the columns of the matrix $U \in M_{n \times p}(\mathbb{R})$. The condition that the matrix $U$ belongs to the orthogonal Stiefel manifold is equivalent with the vectors $u_1, \ldots, u_p \in \mathbb{R}^n$ being orthonormal.

The functions that describe the constraints defining the orthogonal Stiefel manifold as a preimage of a regular value are $F_{aa}, F_{bc} : M_{n \times p}(\mathbb{R}) \to \mathbb{R}$ given by\footnote{\text{We denote by $(\cdot, \cdot)$ the canonical scalar product on $\mathbb{R}^n$ and by $\| \cdot \|$ the induced norm.}}

$$F_{aa}(U) = \frac{1}{2}\|u_a\|^2, \quad 1 \leq a \leq p, \quad F_{bc}(U) = (u_b, u_c), \quad 1 \leq b < c \leq p.$$  

More precisely, we have $F : M_{n \times p}(\mathbb{R}) \to \mathbb{R}^{(p+1)p/2}$, $F := (\ldots, F_{aa}, \ldots, F_{bc}, \ldots)$,

$$St^n_p \simeq F^{-1}(\ldots, 1/2, \ldots, 0, \ldots) \subset M_{n \times p}(\mathbb{R}).$$

Using the description of a Stiefel manifold as a preimage of a regular value, in [3] is given the following elegant explicit form for the tangent space at a point $U \in St^n_p$:

$$T_U St^n_p = \{ U A + (I_n - U U^T) C \mid A \in M_{p \times p}(\mathbb{R}), A = -A^T, C \in M_{n \times p}(\mathbb{R}) \}.$$  

The Lagrange multiplier functions for the case of an orthogonal Stiefel manifold, see [3], are given by the formulas:

$$\sigma_{aa}(U) = \langle \nabla G(U), \nabla F_{aa}(U) \rangle = \left\langle \frac{\partial G}{\partial u_a}(U), u_a \right\rangle;$$

$$\sigma_{bc}(U) = \langle \nabla G(U), \nabla F_{bc}(U) \rangle = \frac{1}{2} \left( \left\langle \frac{\partial G}{\partial u_b}(U), u_b \right\rangle + \left\langle \frac{\partial G}{\partial u_c}(U), u_c \right\rangle \right),$$

where $G : M_{n \times p}(\mathbb{R}) \to \mathbb{R}$ is the extension of the cost function $\tilde{G} : St^n_p \to \mathbb{R}$.

The Hessian matrices of the constraint functions are given by\footnote{By $X \otimes Y$ we denote the Kronecker product of the matrices $X, Y$. The vectors $f_1, \ldots, f_p$ form the canonical basis in the Euclidean space $\mathbb{R}^p$. The $p \times p$ matrix $f_a \otimes f_b^T$ has 1 on the $a$-th row and $b$-th column and the rest 0.}

$$[\text{Hess } F_{aa}(U)] = \begin{bmatrix} 0_n & \cdots & 0_n \\ \vdots & \ddots & \vdots \\ 0_n & \cdots & 0_n \end{bmatrix} \left( f_a \otimes f_b^T \right) \otimes I_n;$$
Theorem 2.1 The Hessian of the cost function \( \tilde{G} : St^n_p \rightarrow \mathbb{R} \) is given by

\[
\text{Hess} \tilde{G}(U) = (\text{Hess} G(U) - \Sigma(U)) \otimes I_p |_{T_U St^n_p \times T_U St^n_p}.
\]

We introduce the symmetric matrix

\[
\Sigma(U) := [\sigma_{kl}(U)] \in M_{p \times p}(\mathbb{R}),
\]

where for \( k \leq l \) we use the formulas (4) and for \( k > l \) we define \( \sigma_{kl}(U) := \sigma_{lk}(U) \).

Using the above expressions for the Hessian matrices of the constraint functions we obtain the following result.

**Theorem 2.2** A matrix \( U \in St^n_p \) is a critical point for the Procrustes cost function if and only if:

(i) the matrix \( B^T AU \) is symmetric;

(ii) \( (I_n - UU^T)(A^T AU - A^T B) = 0 \).
The following formula holds: \( \text{Hess} \ G(U) = \mathbb{I}_p \otimes (A^T A) \).

For a tangent vector \( \Delta(U) = UA_1 + (I_n - UU^T)C_1 \in T_U S^n_p \), we have the following computation:

\[
\text{Hess} \ G(U) (\Delta(U), \Delta(U)) = \langle \mathbb{I}_p \otimes (A^T A) - \Sigma(U) \otimes I_n \rangle (UA_1 + (I_n - UU^T)C_1) - \langle \mathbb{I}_p \otimes (A^T A) \rangle (UA_1 + (I_n - UU^T)C_1)
\]

\[
= - \text{tr}(A_1 U^T A^T U A_1) - 2 \text{tr}(A_1 U^T A^T (I_n - UU^T) C_1) + \text{tr}(C_1^T (I_n - UU^T) A^T A (I_n - UU^T) C_1)
\]

\[
+ \text{tr}(A_1^T \Sigma(U)) - \text{tr}(C_1^T (I_n - UU^T) C_1 \Sigma(U)).
\]

Using the equality (3.8) from [3] and the condition (i) of Theorem 3.3 from [3], the Lagrange multipliers matrix in a critical point \( U \) is given by

\[ \Sigma(U) = U^T A^T A U - U^T A^T B. \]

Consequently, a critical point \( U \) is a local minimum if

\[
\langle B^T A U A_1, A_1 \rangle + 2 \langle A^T A U A_1, (I_n - UU^T) C_1 \rangle + \langle A^T A (I_n - UU^T) C_1, (I_n - UU^T) C_1 \rangle
\]

\[
- \langle (I_n - UU^T) C_1 U^T A^T (AU - B), (I_n - UU^T) C_1 \rangle > 0
\]

holds for all skew-symmetric matrices \( A_1 \in M_{p \times p}(\mathbb{R}) \) and arbitrary matrices \( C_1 \in M_{n \times p}(\mathbb{R}) \), such that \( A_1 \) and \( C_1 \) are not simultaneously null matrices. This condition has been previously obtained in [7] using a different method.

\section{Local frames on Stiefel manifolds}

There exist two frequently used methods to prove that a certain set has a manifold structure. One of them is to prove that the desired set is the preimage of a regular value of a smooth function. Another possibility is to explicitly construct compatible local coordinates (local charts) that cover the entire set. The first approach gives an implicit description of the tangent space. The second approach gives an explicit formula for a basis of the tangent space. Regarding the orthogonal Stiefel manifold as a preimage of a regular value, we explicitly construct a local frame on this manifold.

\textbf{I. Construction of an explicit local frame for orthogonal Stiefel manifolds.}

On the tangent space we consider the Frobenius scalar product:

\[ \langle \Delta_1(U), \Delta_2(U) \rangle = \text{tr}(\Delta_1^T(U) \Delta_2(U)), \quad \Delta_1(U), \Delta_2(U) \in T_U S^n_p. \]

We construct a basis \( \mathcal{B}_U \) for the tangent space \( T_U S^n_p \). The explicit description of the vectors in this basis, as we will see below, decisively depends on the choice of a full rank \( p \times p \) submatrix of \( U \). We split the basis \( \mathcal{B}_U \) as the following union

\[ \mathcal{B}_U = \mathcal{B}_U^{\prime} \cup \mathcal{B}_U^{\prime\prime}. \]

The set \( \mathcal{B}_U^{\prime} \) is formed with tangent vectors of the form

\[ \Delta_{ab}^{\prime}(U) = U A_{ab}, \quad 1 \leq a < b \leq p, \]

where

\[ A_{ab} = (-1)^{a+b} (f_a \otimes f_b^T - f_b \otimes f_a^T), \quad 1 \leq a < b \leq p, \]

form the standard basis for the \( p \times p \) skew-symmetric matrices.

For the next computations we use the following rule for matrix multiplication:

\[ (u \otimes v_\otimes) \cdot (v_\otimes \otimes w^T) = \delta_{\otimes, \otimes} (u \otimes w^T), \]

where \( v_\otimes \) and \( v_\underline{\otimes} \) belong to the same vectorial space.

\footnote{\text{vec}(A)^T (D \otimes B) \text{vec}(C) = \text{tr}(A^T BCD^T).}
Proposition 3.1 The tangent vectors in $\mathcal{B}_U'$ are nonzero and orthogonal one to another.

Proof Assume $\Delta'_{ab}(U) = 0_{p \times p}$. Multiplying the equality to the left with the matrix $U^T$, we obtain that $A_{ab} = 0_{p \times p}$ which is a contradiction.

By a direct computation, for $\langle a_1, b_1 \rangle \neq (a_2, b_2)$, we obtain:

$$
\langle \Delta_{a_1 b_1} (U), \Delta_{a_2 b_2} (U) \rangle = \langle U A_{a_1 b_1}, U A_{a_2 b_2} \rangle = \text{tr}(A_{a_1 b_1} U^T U A_{a_2 b_2}) = \text{tr}(A_{a_1 b_1} A_{a_2 b_2})
$$

$$
= (-1)^{a_1 + b_1 + a_2 + b_2} \text{tr}((f_{b_1} \otimes f_{a_1} - f_{a_1} \otimes f_{b_1}) \cdot (f_{a_2} \otimes f_{b_2} - f_{b_2} \otimes f_{a_2}))
$$

$$
= \varepsilon(\delta_{a_1 a_2} \text{tr}(f_{b_1} \otimes f_{b_2}^T) - \delta_{a_1 b_2} \text{tr}(f_{a_1} \otimes f_{b_2}^T)) - \delta_{a_2 b_1} \text{tr}(f_{a_1} \otimes f_{a_2}^T) + \delta_{b_1 b_2} \text{tr}(f_{a_1} \otimes f_{a_2}^T)) = 0,
$$

where $\varepsilon = (-1)^{a_1 + b_1 + a_2 + b_2}$. ■

As a consequence, we obtain that the set $\mathcal{B}_U'$ has $\frac{p(p-1)}{2}$ linearly independent tangent vectors.

A matrix $U \in \mathcal{S}^n_p$ has rank $p$ and the construction of tangent vectors in $\mathcal{B}_U''$ crucially depends on the choice of a $p \times p$ full rank submatrix of $U$. Assume that this $p \times p$ full rank submatrix $U_p$ is formed with rows $1 \leq i_1 < i_2 < \ldots < i_p \leq n$ of the matrix $U$.

Define the tangent vectors of $\mathcal{B}_U''$ as

$$
\Delta''_{ic}(U) = (I_n - U U^T) C_{ic}, \ i \in \{1, \ldots, n\}\setminus\{i_1, \ldots, i_p\}, \ c \in \{1, \ldots, p\}
$$

where

$$
C_{ic} = e_i \otimes f_c^T
$$

We further decompose the set $\mathcal{B}_U''$ as

$$
\mathcal{B}_U'' = \bigcup_{c=1}^{p} \mathcal{B}_U''
$$

where, for a fixed $c \in \{1, \ldots, p\}$:

$$
\mathcal{B}_U'' = \{ \Delta''_{ic}(U) \mid i \in \{1, \ldots, n\}\setminus\{i_1, \ldots, i_p\} \}.
$$

Proposition 3.2 The tangent vectors in $\mathcal{B}_U''$ have the following properties:

(i) For a fixed $c \in \{1, \ldots, p\}$ the tangent vectors in $\mathcal{B}_U''$ are linearly independent.

(ii) For $c_1, c_2 \in \{1, \ldots, p\}$ and $c_1 \neq c_2$ we have

$$
\langle \Delta''_{c_1}(U), \Delta''_{c_2}(U) \rangle = 0, \ \forall k_1, k_2 \in \{1, \ldots, n\}\setminus\{i_1, \ldots, i_p\}.
$$

Proof (i) We make the notation $I_p = \{i_1, \ldots, i_p\}$. Using the tensorial description of a $n \times p$ matrix we have:

$$
U = \sum_{j \not\in I_p}^{j \not\in I_p} u_{jb} e_j \otimes f_b^T + \sum_{k \in I_p}^{k \not\in I_p} u_{ka} e_k \otimes f_a^T,
$$

$$
U^T = \sum_{s \not\in I_p}^{s \not\in I_p} u_{sd} f_d \otimes e_s^T + \sum_{r \in I_p}^{r \not\in I_p} u_{rg} f_g \otimes e_r^T.
$$

\footnote{The vectors $e_1, \ldots, e_n$ form the canonical basis in the Euclidean space $R^n$. The $n \times p$ matrix $e_i \otimes f_c^T$ has 1 on the $i$-th row and $c$-th column and the rest 0.}
where \( e_j, e_k, e_r, e_s \in \mathbb{R}^n \) and \( f_a, f_b, f_d, f_g \in \mathbb{R}^p \). By a direct computation we obtain the tensorial description of \( U U^T \):

\[
UU^T = \sum_{j \neq l_p} u_{j\beta} u_{\alpha\beta} e_j e_s^T + \sum_{j \neq l_p, r \in l_p} u_{j\beta} u_{\alpha\beta} e_j e_r^T
+ \sum_{k \in l_p, s \neq l_p} u_{k\alpha} u_{s\alpha} e_k e_s^T + \sum_{k, r \in l_p} u_{k\alpha} u_{r\alpha} e_k e_r^T.
\]

For proving the linear independence of the vectors in \( \mathcal{B}''_{U} \), we have the computation:

\[
\mathbb{O}_{n \times p} = \sum_{i \notin l_p} \alpha_i \Delta_{a}^n (U) = \sum_{i \notin l_p} \alpha_i (I_n - UU^T) C_i c
= \sum_{i \notin l_p} \alpha_i e_i \otimes f_c^T - \sum_{i, j \neq l_p} \alpha_i u_{i\beta} u_{\alpha\beta} \delta_{is} e_j e_s^T - \sum_{i, j \neq l_p, r \in l_p} \alpha_i u_{i\beta} u_{\alpha\beta} \delta_{ir} e_j e_r^T
- \sum_{i, s \neq l_p} \sum_{k \in l_p} \alpha_i u_{i\alpha} u_{s\alpha} \delta_{ik} e_k e_s^T - \sum_{k, r \in l_p} \alpha_i u_{k\alpha} u_{r\alpha} \delta_{ir} e_k e_r^T
= \sum_{i \notin l_p} \left( \alpha_i - \sum_{i \neq l_p} \alpha_i u_{i\beta} u_{\alpha\beta} \right) e_i \otimes f_c^T - \sum_{k \in l_p} \left( \sum_{a \notin l_p} \alpha_i u_{k\alpha} u_{i\alpha} \right) e_k \otimes f_c^T,
\]

where \( \delta_{ir} = 0 \) for any \( i \notin l_p \) and \( r \in l_p \). Decomposing the above matrix equality on the subspaces \( \text{Span}\{e_j \otimes f_c^T | j \notin l_p \} \) and \( \text{Span}\{e_k \otimes f_c^T | k \in l_p \} \) we obtain:

\[
\alpha_j - \sum_{i \neq l_p} \alpha_i u_{i\beta} u_{\alpha\beta} = 0, \quad \forall j \notin l_p \tag{8}
\]

\[
\sum_{i \notin l_p} \alpha_i u_{k\alpha} u_{i\alpha} = 0, \quad \forall k \in l_p. \tag{9}
\]

In order to write the above system in a matrix form we need to relabel the elements of the set \( \{1, ..., n\} \setminus I_p \). There exists a unique strictly increasing function \( \sigma : \{1, ..., n\} \setminus I_p \rightarrow \{1, ..., n - p\} \). Analogously, we relabel the set \( I_p \) using the unique strictly increasing function \( \tau : I_p \rightarrow \{1, ..., p\} \).

The \( p \times p \) full rank submatrix \( U_p \) has the following tensorial form:

\[
U_p = \sum_{k \in I_p} u_{ka} \mathbf{f}_{\tau(k)}^T \otimes \mathbf{f}_a^T
\]

and the \( (n - p) \times p \) complement \( U_{n-p} \) of the full rank submatrix \( U_p \) in the matrix \( U \) has the following tensorial form:

\[
U_{n-p} = \sum_{j \notin l_p} u_{ja} \mathbf{h}_{\sigma(j)} \otimes \mathbf{f}_a^T. \tag{5}
\]

The vectors \( \mathbf{h}_{\sigma(i)} \) form the canonical basis of \( \mathbb{R}^{n-p} \).
We define the following \((n-p) \times p\) submatrix of \(C_{ic}\) removing the rows \(i_1, \ldots, i_p\),
\[
\tilde{C}_{ic} = h_{\pi(i)} \otimes f^T_i.
\]
The equations \(\PageIndex{8}\) and \(\PageIndex{9}\) have the equivalent forms:
\[
\sum_{i \notin I_p} \left( \alpha_i \tilde{C}_{ic} - \alpha_i U_{n-p} U^T_{n-p} \tilde{C}_{ic} \right) = O_{(n-p) \times p}
\]
and respectively
\[
\sum_{i \notin I_p} \alpha_i U^T_{n-p} \tilde{C}_{ic} = O_{p \times p}.
\]
Because the full rank submatrix \(U_p\) is invertible we obtain \(\sum_{i \notin I_p} \alpha_i U^T_{n-p} \tilde{C}_{ic} = O_{p \times p}\) and consequently,
\[
\sum_{i \notin I_p} \alpha_i \tilde{C}_{ic} = O_{(n-p) \times p}.
\]
The linear independence of the matrices \(\tilde{C}_{ic}\) implies \(\alpha_i = 0\) for all \(i \notin I_p\),
which proves the linear independence of the vectors in \(\mathcal{C}_{B''_U}\).

(ii) We make the notation \(Z = I_n - UU^T\) and a direct computation shows that \(Z^T Z = Z\).
Consequently, when \(c_1 \neq c_2\) we have:
\[
\langle \Delta''_{k_1c_1}(U), \Delta''_{k_2c_2}(U) \rangle = tr(C^T_{k_1c_1} Z C_{k_2c_2}) = tr(C^T_{k_1c_1} Z C_{k_2c_2})
\]
\[
= tr \left( (f_{c_1} \otimes e_{k_1}^T) \cdot \left( \sum_{i,j} z_{ij} e_i \otimes e_j^T \right) \cdot (e_{k_2} \otimes f^T_{c_2}) \right)
\]
\[
= tr \left( z_{k_1k_2} f_{c_1} \otimes f^T_{c_2} \right) = 0. \tag{\*}
\]

**Proposition 3.3** The tangent vectors in \(\mathcal{B}'_U\) are orthogonal to the vectors in \(\mathcal{B}''_U\).

**Proof** We notice that \(UU^T (I_n - UU^T) = O_{p \times n}\). Consequently, for \(1 \leq a < b \leq p\), \(i \notin I_p\) and \(c \in \{1, \ldots, p\}\) we have:
\[
\langle \Delta''_{ab}(U), \Delta''_{ic}(U) \rangle = tr(A^T_{ab} U^T (I_n - UU^T) C_{ic}) = 0. \tag{\*}
\]

The above results can be summarized in the following theorem.

**Theorem 3.4** For \(U \in St^n_p\) the vectors of the set \(\mathcal{B}_U = \mathcal{B}'_U \cup \mathcal{B}''_U\) form a basis for the tangent space \(T_U St^n_p\). Among them, we have the following orthogonality properties:

(i) \(\mathcal{B}'_U\) is an orthogonal set.

(ii) For \(c_1, c_2 \in \{1, \ldots, p\}\) and \(c_1 \neq c_2\) we have \(c_1 \mathcal{B}'_U \perp c_2 \mathcal{B}''_U\).

(iii) \(\mathcal{B}''_U \perp \mathcal{B}_U\).

In order to have all vectors in the basis \(\mathcal{B}_U\) orthogonal one to the other, we can apply the Gram-Schmidt algorithm to the vectors in \(\mathcal{C}_{B''_U}\) and do this for each \(c \in \{1, \ldots, p\}\).

The above construction of the vectors in \(\mathcal{B}_U\) crucially depends on the choice of the full rank \(p \times p\) submatrix \(U_p\). Choosing another full rank submatrix leads to a change of basis for the tangent space \(T_U St^n_p\).

**II. Local frames on sphere** \(S^{n-1}\). For the case when \(p = 1\), the Stiefel manifold \(St^n_1\) becomes the sphere \(S^{n-1} \subset \mathbb{R}^n\). In this case, for \(x \in S^{n-1}\), we have \(\mathcal{B}'_x = \emptyset\). For a point \(x \in S^{n-1}\), we choose an index \(j \in \{1, \ldots, n\}\) such that \(x_j \neq 0\). Consequently, a local frame for the sphere is given by
\[
\Delta''_{i_1}(x) = (I_n - x \otimes x^T) e_i \otimes f^T_i = (I_n - x \otimes x^T) e_i = e_i - x_i x, \quad i \in \{1, \ldots, n\} \setminus \{j\}.
\]
III. The Hessian of the constraint functions computed on the basis $B_U$. The orthogonality among the elements of the basis $B_U$ has the computational advantage that it renders the Hessian matrices of the constraint functions in a very simple form where most of the entries are zero.

More precisely, let $\Delta'_{\alpha_1, \beta_1}(U)$ and $\Delta'_{\alpha_2, \beta_2}(U)$ be two elements of $B'_U$. By a direct computation, we have the following formulas:

$$
Hess \left( \Delta'_{\alpha_1, \beta_1}(U), \Delta'_{\alpha_2, \beta_2}(U) \right) = (-1)^{\alpha_1 + \beta_1 + \alpha_2 + \beta_2} \left( \delta_{\alpha_1, \alpha_2} \delta_{\beta_1, \beta_2} + \delta_{\alpha_1, \beta_2} \delta_{\beta_1, \alpha_2} \right);
$$

$$
Hess \left( \Delta'_{\alpha_1, \beta_1}(U), \Delta'_{\alpha_2, \beta_2}(U) \right) = (-1)^{\alpha_1 + \beta_1 + \alpha_2 + \beta_2} \left( \delta_{\alpha_1, \alpha_2} \delta_{\beta_1, \beta_2} - \delta_{\alpha_1, \beta_2} \delta_{\beta_1, \alpha_2} + \delta_{\alpha_1, \beta_2} \delta_{\beta_1, \alpha_2} + \delta_{\beta_1, \beta_2} \delta_{\alpha_1, \alpha_2} \right)
$$

Further analyzing the above formulas we have:

$$
Hess \left( \Delta'_{\alpha_1, \beta_1}(U), \Delta'_{\alpha_2, \beta_2}(U) \right) = \begin{cases} 
0 & \text{if } (\alpha_1, \beta_1) \neq (\alpha_2, \beta_2) \\
0 & \text{if } (\alpha_1, \beta_1) = (\alpha_2, \beta_2), a \notin \{\alpha_1, \beta_1\} \\
1 & \text{if } (\alpha_1, \beta_1) = (\alpha_2, \beta_2), a \in \{\alpha_1, \beta_1\}.
\end{cases}
$$

Also, the value $Hess \left( \Delta'_{\alpha_1, \beta_1}(U), \Delta'_{\alpha_2, \beta_2}(U) \right)$ is 0 or $\pm 1$ depending on the ordering and relative position of the integer numbers $b, c, \alpha_1, \beta_1, \alpha_2, \beta_2$.

For the case when $\Delta''_{ab}(U) \in B''(U)$ and $\Delta''_{bd}(U) \in B''(U)$ we have:

$$
Hess \left( \Delta''_{\alpha_1, \beta_1}(U), \Delta''_{\alpha_2, \beta_2}(U) \right) = Hess \left( \Delta''_{\alpha_3, \beta_3}(U), \Delta''_{\alpha_4, \beta_4}(U) \right) = 0.
$$

Let $\Delta''_{j_1d_1}(U)$ and $\Delta''_{j_2d_2}(U)$ be two elements of $B''_U$. By a direct computation, using the notation $Z = [z_{kl}] = I_n - UU^T$, we have the following formulas:

$$
Hess \left( \Delta''_{j_1d_1}(U), \Delta''_{j_2d_2}(U) \right) = \delta_{ad_1} \delta_{ad_2} z_{j_1,j_2} \begin{cases} 
0 & \text{if } a = d_1 = d_2 \\
1 & \text{otherwise}
\end{cases};
$$

$$
Hess \left( \Delta''_{j_1d_1}(U), \Delta''_{j_2d_2}(U) \right) = (\delta_{bd_1} \delta_{bd_2} + \delta_{bd_1} \delta_{bd_2}) z_{j_1,j_2} \begin{cases} 
0 & \text{if } (b, c) = (d_1, d_2) \text{ or } (b, c) = (d_2, d_1) \\
1 & \text{otherwise}
\end{cases}.
$$

For two tangent vectors $\Delta_1(U) = UA_1 + (I_n - UU^T)C_1, \Delta_2(U) = UA_2 + (I_n - UU^T)C_2 \in T_vS^m$ we have

$$
\left( (\Sigma(U) \otimes I_n)(\Delta_1(U), \Delta_2(U)) \right) = -\text{tr}(A_1A_2^\Sigma) + \text{tr}(C_1^T ZC_2\Sigma).
$$

Let $\Delta'_{\alpha_1, \beta_1}(U), \Delta'_{\alpha_2, \beta_2}(U)$ be two elements of $B'_U$ and $\Delta''_{j_1d_1}(U), \Delta''_{j_2d_2}(U)$ be two elements of $B''_U$. Then we have

$$
\left( (\Sigma(U) \otimes I_n)(\Delta'_{\alpha_1, \beta_1}(U), \Delta'_{\alpha_2, \beta_2}(U)) \right) = (-1)^{\alpha_1 + \beta_1 + \alpha_2 + \beta_2} \cdot \left( \delta_{\alpha_1, \alpha_2} \sigma_{\beta_1, \beta_2} + \delta_{\beta_1, \beta_2} \sigma_{\alpha_1, \alpha_2} \right)
$$

$$
-\delta_{\alpha_1, \beta_1} \sigma_{\alpha_1, \alpha_2} - \delta_{\beta_1, \beta_2} \sigma_{\alpha_1, \alpha_2}
$$

and

$$
\left( (\Sigma(U) \otimes I_n)(\Delta''_{j_1d_1}(U), \Delta''_{j_2d_2}(U)) \right) = z_{j_1,j_2} \sigma_{d_1,d_2}.
$$

4 Newton algorithm on Stiefel manifolds

In what follows, we customize the Embedded Newton Algorithm from Introduction to the specific case of orthogonal Stiefel manifolds.
Embedded Newton algorithm on Stiefel manifolds:

1. Consider a smooth prolongation $G : \mathcal{M}_{n \times p}(\mathbb{R}) \to \mathbb{R}$ of the cost function $\tilde{G} : S_{\mathcal{T}}^{\alpha} \to \mathbb{R}$.

2. Compute the Lagrange multiplier functions:
   \[
   \sigma_{aa}(U) = \frac{\partial G}{\partial u_a}(U), \quad \sigma_{be}(U) = \frac{1}{2} \left( \frac{\partial G}{\partial u_c}(U) + \frac{\partial G}{\partial u_b}(U) \right),
   \]
   where $1 \leq a \leq p$, $1 \leq b < c \leq p$. Form the symmetric matrix $\Sigma(U)$.

3. Choose the retraction $\mathcal{R} : T_{U}S_{\mathcal{T}}^{\alpha} \subset T_{U}M_{n \times p}(\mathbb{R}) \to S_{\mathcal{T}}^{\alpha}$, $\mathcal{R}_{U}(v_{U}) = qf(U + v_{U})$, where $qf(A)$ denotes the $Q$ factor of the decomposition of $A \in \mathcal{M}_{n \times p}(\mathbb{R})$ as $A = QR$, where $Q$ belongs to $S_{\mathcal{T}}^{\alpha}$ and $R$ is an upper triangular $n \times p$ matrix with strictly positive diagonal elements (see [3]).

4. Input $U^{(0)} \in S_{\mathcal{T}}^{\alpha}$ and $k = 0$.

5. repeat
   - Determine a set $I_{p}(U^{(k)})$ containing the indexes of the rows that form a full rank submatrix of $U^{(k)}$. Construct the basis $\mathcal{B}_{U^{(k)}} = \mathcal{B}_{U^{(k)}}^{1} \cup \mathcal{B}_{U^{(k)}}^{2}$ using the equations [6] and [7].
   - Compute the coordinate functions:
     \[
     g'_{ab}(U^{(k)}) = (-1)^{a+b} \left( \frac{\partial G}{\partial u_a}(U^{(k)}), u_{b}^{(k)} \right) - \left( \frac{\partial G}{\partial u_b}(U^{(k)}), u_{a}^{(k)} \right), \quad 1 \leq a < b \leq p
     \]
     \[
     g''_{ij}(U^{(k)}) = \left( \frac{\partial G}{\partial u_i}(U^{(k)}), u_{j}^{(k)} \right), \quad i \in \{1, 2, ..., n\} \setminus I_{p}(U^{(k)}), \quad c \in \{1, 2, ..., n\},
     \]
     where $z_{i}^{(k)}$ is the vector formed with the $i$-th column of the matrix $Z^{(k)} = I_{n} - U^{(k)}U^{(k)T}$.
   - Compute the components of the Hessian matrix $\text{Hess} \tilde{G}$ of the cost function $\tilde{G}$ (using formulas [10] and [11]):
     \[
     h_{\alpha_1, \beta_1, \alpha_2, \beta_2}(U^{(k)}) = \left( \text{Hess} G(U^{(k)}) - \Sigma(U^{(k)}) \otimes I_{n} \right) \left( \Delta'_{\alpha_1, \beta_1}(U^{(k)}), \Delta'_{\alpha_2, \beta_2}(U^{(k)}) \right)
     \]
     \[
     h_{\alpha, \beta, d}(U^{(k)}) = \text{Hess} G(U^{(k)}) \left( \Delta'_{\alpha, \beta}(U^{(k)}), \Delta''_{d}(U^{(k)}) \right)
     \]
     \[
     h_{j_1, d_1, j_2, d_2}(U^{(k)}) = \left( \text{Hess} G(U^{(k)}) - \Sigma(U^{(k)}) \otimes I_{n} \right) \left( \Delta''_{j_1, d_1}(U^{(k)}), \Delta''_{j_2, d_2}(U^{(k)}) \right),
     \]
     where $\alpha_1, \beta_1, \alpha_2, \beta_2, \alpha, \beta \in \{1, 2, ..., p\}$ with $\alpha_1 < \beta_1$, $\alpha_2 < \beta_2$, $\alpha < \beta$, and $j_1, j_2 \in \{1, 2, ..., n\} \setminus I_{p}(U^{(k)})$ and $d, d_1, d_2 \in \{1, 2, ..., n\}$.
   - Solve the linear system [2] with the unknowns $\left( v^{\alpha \beta}, ..., v^{j d}, ... \right)$:
     \[
     \begin{bmatrix}
     h_{\alpha_1, \beta_1, \alpha_2, \beta_2}(U^{(k)}) & h_{\alpha_1, \beta_1, j_2, d_2}(U^{(k)}) \\
     h_{j_1, d_1, \alpha_2, \beta_2}(U^{(k)}) & h_{j_1, d_1, j_2, d_2}(U^{(k)})
     \end{bmatrix}
     \begin{bmatrix}
     v^{\alpha_2 \beta_2} \\
     v^{j_2 d_2}
     \end{bmatrix}
     = - \begin{bmatrix}
     g'_{\alpha_1, \beta_1}(U^{(k)}) \\
     g''_{j_1, d_1}(U^{(k)})
     \end{bmatrix}.
     \]
   - Construct the line search vector
     \[
     v_{U^{(k)}} = \sum_{1 \leq \alpha < \beta \leq p} v^{\alpha \beta} \Delta'_{\alpha, \beta}(U^{(k)}) + \sum_{j \in \{1, 2, ..., n\} \setminus I_{p}(U^{(k)})} v^{j d} \Delta''_{j, d}(U^{(k)}).
     \]
   - Set $U^{(k+1)} = R_{U^{(k)}}(v_{U^{(k)}}) = qf(U^{(k)} + v_{U^{(k)})}$.
   until $U^{(k+1)}$ sufficiently minimizes $\tilde{G}$.
For particular cases of cost functions defined on Stiefel manifolds, using ingenious matrix representations of the Hessian and the gradient vector field, in [13] and [2] are given explicit formulas to solve the Newton equation.

5 Example: A Brockett cost function

We consider the following particular Brockett cost function on $St_4^2$

$$\tilde{G}(U) = \mu_1 u_1^T A u_1 + \mu_2 u_2^T A u_2,$$

where $\mu_1 = 1$, $\mu_2 = 2$, and $A = \text{diag}(1, 2, 3, 4)$. The cost function being quadratic it is invariant under the sign change of the vectors that give the columns of the matrix $U$, but it is not invariant under the order of these column vectors. As shown in [3], a matrix $U \in St_4^2$ is a critical point of the Brockett cost function if and only if every column vector of the matrix $U$ is an eigenvector of the matrix $A$. Using the formulas [12] for the components of the Hessian matrix, we obtain the following characterization of the critical points for the above Brockett cost function:

- four global minima generated by $[e_2, e_4]$ (i.e., $[e_2, e_1]$, $[-e_2, e_1]$, $[e_2, -e_1]$, and $[-e_2, -e_1]$) with the value of the cost function equals 4.
- eight saddle points generated by $[e_1, e_2]$ and $[e_3, e_1]$ with the value of the cost function equals 5.
- four saddle points generated by $[e_4, e_1]$ with the value of the cost function equals 6.
- eight saddle points generated by $[e_1, e_3]$ and $[e_3, e_2]$ with the value of the cost function equals 7.
- eight saddle points generated by $[e_2, e_3]$ and $[e_4, e_2]$ with the value of the cost function equals 8.
- four saddle points generated by $[e_1, e_4]$ with the value of the cost function equals 9.
- eight saddle points generated by $[e_2, e_4]$ and $[e_4, e_3]$ with the value of the cost function equals 10.
- four global maxima generated by $[e_3, e_4]$ with the value of the cost function equals 11.

We have run the algorithm for some initial points and we present the convergence behavior of the sequence of iterations toward the corresponding critical points.

(a) For $U_0 = \left[ -\frac{\sqrt{3}}{4} (e_2 + e_4), \frac{3\sqrt{2}}{4} (e_1 - e_3) \right]$ the Hessian matrix of $\tilde{G}$ at this point is degenerate and consequently the Newton algorithm does not start.

(b) For $U^*_0 = \left[ -\frac{\sqrt{3}}{3} (e_2 + e_4), \frac{3\sqrt{2}}{3} (e_1 + e_2 - e_4) \right]$ the Newton algorithm converges to the saddle point $[-e_4, e_2].$

(c) For $U^{**}_0 = \left[ \frac{\sqrt{3}}{3} (e_1 - e_3 + e_4), -\frac{\sqrt{2}}{3} (e_1 + e_3) \right]$ the Newton algorithm converges to the saddle point $[-e_3, -e_1].$

(d) For

$$U^{***} = \begin{bmatrix}
19280764801941338 & \sqrt{3} & 378260825115369995 & \sqrt{3} \\
263166861948156647 & \sqrt{2} & 789506658444694947 & \sqrt{3} \\
421362408154489663 & \sqrt{2} & 2026400034047076523 & \sqrt{3} \\
6842390410652097232 & 0 & 1026358651978109233 & \sqrt{3} \\
2324483553559405177 & \sqrt{2} & -2612795675858573497 & \sqrt{3} \\
6842390410652097232 & 0 & 1026358651978109233 & \sqrt{3}
\end{bmatrix},$$

the Newton algorithm converges to the global minimum $[-e_2, e_1].$

In Figure 1 we show a comparison between the convergence behavior of every algorithm for a non-convex cost function. Despite the fact that Newton algorithm is considerably faster than steepest descent algorithm it is more prone
to go to saddle points. It also may happen that the Hessian matrix of the cost function at the initial point is degenerate (case of $U_0^*$) and consequently the equation (3) cannot be solved.

An alternative method for finding critical points of a cost function is to consider the dynamical system generated by the gradient vector field associated to the cost function. This approach has been implemented in [7], where an ODE solver has been chosen to approximate the solution of the dynamical system that converges to a critical point. Such ODE solvers should be carefully chosen so that they preserve the constraint manifold. To tackle this possible inconvenience, in [10] an additional non-linear projection scheme has been suggested. In contrast, the embedded Newton algorithm presented above is constructed such that it preserves the constraint manifold at each iteration.

6 Conclusions

In order to solve an optimization problem on Stiefel manifolds we regard a Stiefel manifold as a constraint manifold and we compute the Hessian matrix of the cost function which defines the optimization problem. We introduce an explicit local frame on a Stiefel manifold, which allows the explicit description of the Newton algorithm for such a manifold. For a particular Brockett cost function we characterize the critical points and we discuss some numerical simulations of the algorithm.

Acknowledgements. This work was supported by a grant of Ministry of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P4-ID-PCE-2016-0165, within PNCDI III.

Appendix

The Lagrange multiplier functions $\sigma_\alpha : (M, g) \to \mathbb{R}$, for $\alpha = 1, r$, are given by, see [4] and [5]

$$\sigma_\alpha(x) := \frac{\det \text{Gram}_{(F_1, \ldots, F_{\alpha-1}, F_\alpha, F_{\alpha+1}, \ldots, F_r)}(x)}{\det \text{Gram}_{(F_1, \ldots, F_r)}(x)},$$

(13)
with the Gramian matrix defined by

\[
\text{Gram}_{(f_1, \ldots, f_r)}^{(h_1, \ldots, h_r)} = \begin{bmatrix}
    g(\text{grad } h_1, \text{grad } f_1) & \cdots & g(\text{grad } h_r, \text{grad } f_1) \\
    \vdots & \ddots & \vdots \\
    g(\text{grad } h_1, \text{grad } f_r) & \cdots & g(\text{grad } h_r, \text{grad } f_r)
\end{bmatrix}.
\]

References

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

[2] Aihara, K., Sato, H.: A matrix-free implementation of Riemannian Newton’s method on the Stiefel manifold. Optim. Lett. 11, 1729-1741 (2017)

[3] Birtea, P., Cașu, I., Comănescu, D.: First order optimality conditions and steepest descent algorithm on orthogonal Stiefel manifolds. Optim. Lett., https://doi.org/10.1007/s11590-018-1319-x (2018)

[4] Birtea, P., Comănescu, D.: Geometric dissipation for dynamical systems. Comm. Math. Phys. 316, 375-394 (2012)

[5] Birtea, P., Comănescu, D.: Hessian operators on constraint manifolds. J. Nonlinear Science 25, 1285-1305 (2015)

[6] Birtea, P., Comănescu, D.: Newton algorithm on constraint manifolds and the 5-Electron Thomson problem. J. Optim. Theor. Appl. 173, 563-583 (2017)

[7] Chu, M.T., Trendafilov, N.T.: The orthogonally constrained regression revisited. J. Comput. and Graphical Statistics 10, 746-771 (2001)

[8] Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. 20, 303-353 (1998)

[9] Gabay, D.: Minimizing a differentiable function over a differential manifold. J. Optim. Theory Appl. 37, 177-219 (1982)

[10] Gear, C.W.: Maintaining solution invariants in the numerical solution of ODEs. SIAM J. Sci. Stat. Comput. 7, 734-743 (1986)

[11] Lu, J., Davidson, T.N., Luo, Z.-Q.: Blind separation of BPSK signals using Newton’s method on the Stiefel manifold. IEEE International Conference on Acoustics, Speech, and Signal Processing 4, 301-304 (2003)

[12] Manton, J.H.: A framework for generalising the Newton method and other iterative methods from Euclidean space to manifolds. Numer. Math. 129, 91-125 (2015)

[13] Sato, H.: Riemannian Newton-type methods for joint diagonalization on the Stiefel manifold with application to independent component analysis. A Journal of Mathematical Programming and Operations Research 66, 2211-2231 (2017)

[14] Shukla, A., Anand, S.: Metric learning based automatic segmentation of patterned species. IEEE Conference on Image Processing, Phoenix, AZ, USA, September 2016

[15] Turaga, P., Veeraraghavan, A., Chellappa, R.: Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision. IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, June 2008

[16] Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Math. Progr. (Series A) 142, 397-434 (2013)