Riemannian BFGS Algorithm with Applications

Chunhong Qi[1], Kyle A. Gallivan[1], and P.-A. Absil[2]

[1] Department of Mathematics, Florida State University, Tallahassee, FL, 32306, USA, {cqi, gallivan}@math.fsu.edu

[2] Département d'ingénierie mathématique, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium, absil@inma.ucl.ac.be

Summary. We present an algorithm model, called Riemannian BFGS (RBFGS), that subsumes the classical BFGS method in $\mathbb{R}^n$ as well as previously proposed Riemannian extensions of that method. Of particular interest is the choice of transport used to move information between tangent spaces and the different ways of implementing the RBFGS algorithm.

1 Introduction

Optimization on manifolds, or *Riemannian optimization*, concerns finding an optimum (global or local) of a real-valued function defined over a smooth manifold. A brief introduction to the area can be found in [1] in this volume, and we refer to [3] and the many references therein for more details. Optimization on manifolds finds applications in two broad classes of situations: classical equality-constrained optimization problems where the constraints specify a submanifold of $\mathbb{R}^n$; and problems where the objective function has continuous invariance properties that we want to eliminate for various reasons, e.g., efficiency, consistency, applicability of certain convergence results, and avoiding failure of certain algorithms due to degeneracy. As a result, the generalization to manifolds of algorithms for unconstrained optimization in $\mathbb{R}^n$ can yield useful and efficient numerical methods; see, e.g., recent work on Riemannian trust-region methods [2] and other methods mentioned in [3]. Since BFGS is one of the classical methods for unconstrained optimization (see [7, 10]), it is natural that its generalization be a topic of interest.

Some work has been done on BFGS for manifolds. Gabay [9, §4.5] discussed a version using parallel transport. Brace and Manton [6] have a version on the Grassmann manifold for the problem of weighted low-rank approximations. Savas and Lim [11] apply a version on a product of Grassmann manifolds to the problem of best multilinear low-rank approximation of tensors.

Gabay's Riemannian BFGS [9, §4.5] differs from the classical BFGS method in $\mathbb{R}^n$ (see, e.g., [10, Alg. 6.1]) in five key aspects: (i) The search space, to which the iterates x_k belong, is a Riemannian submanifold M of $\mathbb{R}^n$ specified by equality constraints; (ii) The search direction at x_k is a tangent vector to M at x_k; (iii) The update along the search direction is performed along the geodesic determined by the search direction; (iv) The usual quantities s_k and y_k that appear in the secant equation are tangent vectors to M at x_{k+1}, obtained using the Riemannian parallel transport (i.e., the parallel transport induced by the Levi-Civita connection) along the geodesic. (v) The Hessian approximation $\mathcal{B}_k$ is a linear transformation of the tangent space $T_{x_k}M$ that gets updated using a generalized version of the BFGS update formula. This generalized formula specifies recursively how $\mathcal{B}_k$ applies to elements of $T_{x_k}M$.

In this paper, we present an algorithm model (or meta-algorithm), dubbed RBFGS, that subsumes Gabay's Riemannian BFGS method. Whereas Gabay's method is fully specified by the Riemannian manifold, the cost function, and the initial iterate, our RBFGS algorithm offers additional freedom in the choice of a retraction and a vector transport (see Section 2 for a brief review of these two concepts). This additional freedom affects points (iii) and (iv) above. For (iii), the curves along which the update is performed are specified by the retraction. For (iv), the Levi-Civita parallel transport is replaced by the more general concept of vector transport. If the retraction is selected as the Riemannian exponential and the vector transport is chosen to be the Levi-Civita parallel transport, then the RBFGS algorithm reduces to Gabay's algorithm (barring variations of minor importance, e.g., in the line-search procedure used).

The impact of the greater freedom offered by the RBFGS algorithm varies according to the manifold of interest. On the sphere, for example, the computational cost of the Riemannian exponential and the Levi-Civita parallel transport is reasonable, and there is not much to be gained by choosing computationally cheaper alternatives. In contrast, as we will show in numerical experiments, when the manifold is the Stiefel manifold, $\mathrm{St}(p,n)$, of orthonormal p-frames in $\mathbb{R}^n$, the improvement in computational time can be much more significant.

This paper also improves on Gabay's work by discussing the practical implementation of the algorithm. When the manifold M is a submanifold of $\mathbb{R}^n$, we offer the alternatives of either representing the tangent vectors and the approximate Hessian using a basis in the tangent spaces, or relying on the canonical inclusion of M in $\mathbb{R}^n$. The latter leads to representations of tangent vectors as n-tuples of real numbers and of the approximate Hessian as an $n \times n$ matrix. This approach may offer a strong advantage when the co-dimension of M is sufficiently small.

Another feature of RBFGS is that it does not assume that M is a submanifold of a Euclidean space. As such, it can be applied to quotient manifolds as well. However, in this paper, we concentrate the practical implementation discussion on the submanifold case.

This paper is a first glimpse at ongoing work that aims at a systematic analysis and evaluation of the Riemannian versions of the BFGS algorithm. It is organized as follows. The general RBFGS algorithm is given in Section 3. The two implementation approaches and the particular implementation on certain manifolds are given in Section 4. In Section 5, we summarize the results of our numerical experiments for two application problems: the Rayleigh quotient problem on the sphere S^{n-1} and a matrix Procrustes problem on the compact Stiefel manifold.

2 Mathematical preliminaries

The notion of retraction on a manifold, due to Adler *et al.* [4], encompasses all first-order approximations to the Riemannian exponential. Here we recall the definition as given in [3].

Definition 1. *A* retraction *on a manifold M is a mapping R from the tangent bundle TM onto M with the following properties. Let R_x denote the restriction of R to T_xM.*

1. *R is continuously differentiable.*
2. *$R_x(0_x) = x$, where 0_x denotes the zero element of T_xM.*
3. *With the canonical identification $T_{0_x}T_xM \simeq T_xM$, R_x satisfies $\mathrm{D}R_x(0_x) = id_{T_xM}$, where D denotes the derivative and id_{T_xM} denotes the identity mapping on T_xM.*

The retraction is used as a way to take a step in the direction of a tangent vector. Choosing a good retraction amounts to finding an approximation of the exponential mapping that can be computed with low computational cost while not adversely affecting the behavior of the optimization algorithm.

Next we recall the concept of vector transport, which specifies how to move a tangent vector from one tangent space to another. This is also used to move a linear operator from one tangent space to another, e.g., the approximate Hessian in (4). The notion of vector transport was introduced in [3] for reasons similar to those that motivated the introduction of retractions, namely, to provide a framework for using computationally less expensive approximations of the Levi-Civita parallel translation. The definition below, illustrated in Figure 1, invokes the Whitney sum $TM \oplus TM$, which stands for the set of all ordered pairs of tangent vectors with same foot.

Definition 2. *A* vector transport *on a manifold M is a smooth mapping: $TM \oplus TM \to TM$, $(\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}(\xi_x) \in TM$ satisfying the following properties for all $x \in M$.*

1. *(Associated retraction) There exists a retraction R, called the* retraction associated with *$\mathcal{T}$, such that, for all η_x, ξ_x, it holds that $\mathcal{T}_{\eta_x}\xi_x \in T_{R_x(\eta_x)}M$.*
2. *(Consistency) $\mathcal{T}_{0_x}\xi_x = \xi_x$ for all $\xi_x \in T_xM$;*

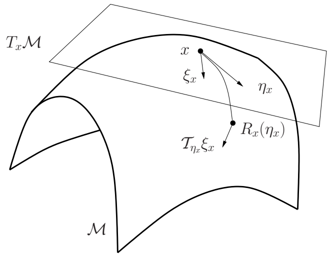

Fig. 1. Vector transport.

3. (Linearity) The mapping $\mathcal{T}_{\eta_x} : T_xM \to T_{R(\eta_x)}M,\ \xi_x \mapsto \mathcal{T}_{\eta_x}(\xi_x)$ is linear.

Note that, in general, vector transports are not isometries; in fact, the definition of a vector transport does not even assume an underlying Riemannian metric. When M is a Riemannian manifold and the vector transport is selected to be the Levi-Civita parallel translation, then it is an isometry. When it exists, the inverse of the linear map $\mathcal{T}_{\eta_x}$ is denoted by $(\mathcal{T}_{\eta_x})^{-1}$. Observe that $(\mathcal{T}_{\eta_x})^{-1}(\xi_{R_x(\eta_x)})$ belongs to T_xM. If M is an embedded submanifold of a Euclidean space and M is endowed with a retraction R, then a particular choice of vector transport is given by

$$\mathcal{T}_{\eta_x}\xi_x := \mathrm{P}_{R_x(\eta_x)}\xi_x, \tag{1}$$

where P_x denotes the orthogonal projector onto T_xM. Depending on the manifold, this vector transport may be much less expensive to compute than the Levi-Civita parallel transport. Other choices may also be used to achieve computational savings. It may happen that the chosen vector transport and its inverse are not defined everywhere, but then the set of problematic points is usually of measure zero, and no difficulty is observed in numerical experiments.

3 The RBFGS Algorithm

The structure of the RBFGS algorithm is given in Algorithm 1. Recall that, given a smooth scalar field f on a Riemannian manifold M with Riemannian metric g, the gradient of f at x, denoted by $\operatorname{grad} f(x)$, is defined as the unique element of T_xM that satisfies:

$$g_x(\operatorname{grad} f(x), \xi) = \mathrm{D}f(x)[\xi], \forall \xi \in T_xM. \tag{2}$$

The line-search procedure in Step 4 of RBFGS uses Armijo's condition.

The RBFGS algorithm can also be reformulated to work with the inverse Hessian approximation $\mathcal{H}_k = {\mathcal{B}_k}^{-1}$ rather than with the Hessian approximation B_k. In this case, Step 6 of RBFGS is replaced by

Algorithm 1 RBFGS

```
Given: Riemannian manifold $M$ with Riemannian metric $g$; vector transport $\mathcal{T}$ on $M$ with associated retraction $R$; smooth real-valued function $f$ on $M$; initial iterate $\mathbf{x}_0 \in M$; initial Hessian approximation $\mathcal{B}_0$.
for k = 0, 1, 2, ... do
    Obtain $\eta_k \in T_{\mathbf{x}_k} M$ by solving $\mathcal{B}_k \eta_k = -\operatorname{grad} f(\mathbf{x}_k)$.
    Set step size $\alpha = 1$, $c = g(\operatorname{grad} f(\mathbf{x}_k), \eta_k)$. While $f(R_{\mathbf{x}_k}(2\alpha\eta_k)) - f(\mathbf{x}_k) < \alpha c$, set $\alpha := 2\alpha$. While $f(R_{\mathbf{x}_k}(\alpha\eta_k)) - f(\mathbf{x}_k) \geq 0.5\alpha c$, set $\alpha := 0.5\alpha$. Set $\mathbf{x}_{k+1} = R_{\mathbf{x}_k}(\alpha\eta_k)$.
    Define $s_k = \mathcal{T}_{\alpha\eta_k}(\alpha\eta_k)$ and $y_k = \operatorname{grad} f(\mathbf{x}_{k+1}) - \mathcal{T}_{\alpha\eta_k}(\operatorname{grad} f(\mathbf{x}_k))$.
    Define the linear operator $\mathcal{B}_{k+1} : T_{\mathbf{x}_{k+1}} M \to T_{\mathbf{x}_{k+1}} M$ by
        $$\mathcal{B}_{k+1} p = \tilde{\mathcal{B}}_k p - \frac{g(s_k, \tilde{\mathcal{B}}_k p)}{g(s_k, \tilde{\mathcal{B}}_k s_k)} \tilde{\mathcal{B}}_k s_k + \frac{g(y_k, p)}{g(y_k, s_k)} y_k \quad \text{for all } p \in T_{\mathbf{x}_{k+1}} M, \qquad (3)$$
    with
        $$\tilde{\mathcal{B}}_k = \mathcal{T}_{\alpha\eta_k} \circ \mathcal{B}_k \circ (\mathcal{T}_{\alpha\eta_k})^{-1}. \qquad (4)$$
end for
```

$$\mathcal{H}_{k+1} p = \tilde{\mathcal{H}}_k p - \frac{g(y_k, \tilde{\mathcal{H}}_k p)}{g(y_k, s_k)} s_k - \frac{g(s_k, p_k)}{g(y_k, s_k)} \tilde{\mathcal{H}}_k y_k + \frac{g(s_k, p) g(y_k, \tilde{\mathcal{H}}_k y_k)}{g(y_k, s_k)^2} s_k + \frac{g(s_k, s_k)}{g(y_k, s_k)} p \qquad (5)$$

with

$$\tilde{\mathcal{H}}_k = \mathcal{T}_{\eta_k} \circ \mathcal{H}_k \circ (\mathcal{T}_{\eta_k})^{-1}. \qquad (6)$$

This yields a mathematically equivalent algorithm. It is useful because it makes it possible to cheaply compute an approximation of the inverse of the Hessian. This may make RBFGS advantageous even in the case where we have a cheap exact formula for the Hessian but not for its inverse.

4 Practical Implementation of RBFGS

4.1 Two Approaches

A practical implementation of RBFGS requires the following ingredients: (i) an efficient numerical representation for points x on M, tangent spaces $T_x M$ and the inner products $g_x(\xi_1, \xi_2)$ on $T_x M$; (ii) an implementation of the chosen retraction $R_x : T_x M \to M$; (iii) efficient formulas for $f(x)$ and $\operatorname{grad} f(x)$; (iv) an implementation of the chosen vector transport $\mathcal{T}_{\eta_x}$ and its inverse $(\mathcal{T}_{\eta_x})^{-1}$; (v) a method for solving

$$\mathcal{B}_k \eta_k = -\operatorname{grad} f(\mathbf{x}_k), \qquad (7)$$

where $\mathcal{B}_k$ is defined recursively through (3), or alternatively, a method for computing $\eta_k = -\mathcal{H}_k \mathrm{grad}\, f(\mathbf{x}_k)$ where $\mathcal{H}_k$ is defined recursively by (5). Point (v) is the main difficulty. In this paper, we restrict to the case where M is a submanifold of $\mathbb{R}^n$, and we construct explicitly a matrix representation of $\mathcal{B}_k$. We discuss two implementation approaches.

Approach 1 realizes $\mathcal{B}_k$ as an $n \times n$ matrix $B_k^{(n)}$. Since M is a submanifold of $\mathbb{R}^n$, tangent spaces $T_x M$ are naturally identified with subspaces of $\mathbb{R}^n$ (see [3, §3.5.7] for details), and it is very common to use the same notation for a tangent vector and its corresponding element of $\mathbb{R}^n$. However, to explain Approach 1, it is useful to distinguish the two objects. To this end, let ι_x denote the natural inclusion of $T_x M$ in $\mathbb{R}^n$, $\iota_x : \ T_x M \to \mathbb{R}^n, \ \ \xi_x \mapsto \ \iota_x(\xi_x)$.

To represent $\mathcal{B}_k$, we pick $B_k^{(n)} \in \mathbb{R}^{n \times n}$ such that, for all $\xi_{x_k} \in T_{x_k} M$,

$$B_k^{(n)} \iota_{x_k}(\xi_{x_k}) = \iota_{x_k}(\mathcal{B}_k \xi_{x_k}). \tag{8}$$

Note that condition (8) does not uniquely specify $B_k^{(n)}$; its action on the normal space is irrelevant. Solving the linear system (7) then amounts to finding $\iota_{x_k}(\eta_k)$ in $\iota_{x_k}(T_{x_k} M)$ that satisfies

$$B_k^{(n)} \iota_{x_k}(\eta_k) = -\iota_{x_k}(\mathrm{grad}\, f(x_k)). \tag{9}$$

It remains to give an expression for the update formula (3). To this end, let $T_{\alpha\eta_k}^{(n)}$ be the $n \times n$ matrix that satisfies $T_{\alpha\eta_k}^{(n)} \iota_{x_k}(\xi_{x_k}) = \iota_{x_{k+1}}(\mathcal{T}_{\alpha\eta_k} \xi_{x_k})$ for all $\xi_{x_k} \in T_{x_k} M$ and $T_{\alpha\eta_k}^{(n)} \zeta_k = 0$ for all $\zeta_k \perp \iota_{x_k}(T_{x_k} M)$. Since M is an embedded submanifold of $\mathbb{R}^n$, the Riemannian metric is given by $g(\xi_x, \eta_x) = \iota_x(\xi_x)^T \iota_x(\eta_x)$ and the update equation (3) is then

$$B_{k+1}^{(n)} = \tilde{B}_k^{(n)} - \frac{\tilde{B}_k^{(n)} \iota_{x_{k+1}}(s_k) \iota_{x_{k+1}}(s_k)^T \tilde{B}_k^{(n)}}{\iota_{x_{k+1}}(s_k)^T \tilde{B}_k^{(n)} \iota_{x_{k+1}}(s_k)} + \frac{\iota_{x_{k+1}}(y_k) \iota_{x_{k+1}}(y_k)^T}{\iota_{x_{k+1}}(y_k)^T \iota_{x_{k+1}}(s_k)},$$

where $\tilde{B}_k^{(n)} = T_{\alpha\eta_k}^{(n)} B_k^{(n)} \big((T_{\alpha\eta_k})^{(n)}\big)^{\dagger}$ and † denotes the pseudoinverse.

Approach 2 realizes $\mathcal{B}_k$ by a $d \times d$ matrix $B_k^{(d)}$ using bases, where d denotes the dimension of M. Given a basis $(E_{k,1}, \ldots, E_{k,d})$ of $T_{x_k} M$, if $\hat{G}_k \in \mathbb{R}^d$ is the vector of coefficients of $\mathrm{grad}\, f(x_k)$ in the basis and $B_k^{(d)}$ is the $d \times d$ matrix representation of $\mathcal{B}_k$ in the basis, then we must solve $B_k^{(d)} \hat{\eta}_k = -\hat{G}_k$ for $\hat{\eta}_k \in \mathbb{R}^d$, and the solution η_k of (7) is given by $\eta_k = \sum_{i=1}^d E_{k,i}(\hat{\eta}_k)_i$.

4.2 Implementation on the Unit Sphere

We view the unit sphere $S^{n-1} = \{x \in \mathbb{R}^n : x^T x = 1\}$ as a Riemannian submanifold of the Euclidean space $\mathbb{R}^n$. In the rest of the paper, we abuse the notation by ignoring the inclusions to simplify the formulas.

The tangent space at x, orthogonal projection onto the tangent space at x, and the retraction chosen are given by

$$\begin{aligned} T_x S^{n-1} &= \{\xi \in \mathbb{R}^n \ : \ x^T \xi = 0\} \\ \mathrm{P}_x \xi_x &= \xi - x x^T \xi_x \\ R_x(\eta_x) &= (x + \eta_x)/\|(x + \eta_x)\|, \end{aligned}$$

where $\|\cdot\|$ denotes the Euclidean norm.

Vector transport (1) on S^{n-1} is given by

$$\mathcal{T}_{\eta_x} \xi_x = \left(I - \frac{(x + \eta_x)(x + \eta_x)^T}{\|x + \eta_x\|^2} \right) \xi_x \tag{10}$$

which takes a vector ξ_x that belongs to the orthogonal complement of x (because it is in the tangent space to the sphere at x) and projects it along $(x+\eta_x)$ into the orthogonal complement of $(x + \eta_x)$. To invert (10), we start from a vector in the orthogonal complement of $(x+\eta_x)$ and project it along $(x+\eta_x)$ into the orthogonal complement of x. The result is an oblique projection

$$(\mathcal{T}_{\eta_x})^{-1}(\xi_{R_x(\eta_x)}) = \left(I - \frac{(x + \eta_x) x^T}{x^T (x + \eta_x)} \right) \xi_{R_x(\eta_x)} \tag{11}$$

For the unit sphere, the Levi-Civita parallel transport of $\xi \in T_x S^{n-1}$ along the geodesic, γ, from x in direction $\eta \in T_x S^{n-1}$ is [5]

$$P_\gamma^{t \leftarrow 0} \xi = \left(I_n + (\cos(\|\eta\| t) - 1) \frac{\eta \eta^T}{\|\eta\|^2} - \sin(\|\eta\| t) \frac{x \eta^T}{\|\eta\|} \right) \xi.$$

This parallel transport and its inverse have computational costs comparable to the chosen vector transport and its inverse.

4.3 Implementation on the Compact Stiefel Manifold $\mathrm{St}(p, n)$

We view the compact Stiefel manifold $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\}$ as a Riemannian submanifold of the Euclidean space $\mathbb{R}^{n \times p}$ endowed with the canonical Riemannian metric $g(\xi, \eta) = \mathrm{tr}(\xi^T \eta)$. The tangent space at X and the associated orthogonal projection are given by

$$\begin{aligned} T_X \mathrm{St}(p, n) &= \{Z \in \mathbb{R}^{n \times p} : X^T Z + Z^T X = 0\} \\ &= \{X \varOmega + X^\perp K : \varOmega^T = -\varOmega, K \in \mathbb{R}^{(n-p) \times p}\} \\ \mathrm{P}_X \xi_X &= (I - X X^T) \xi_X + X \mathrm{skew}(X^T \xi_X) \end{aligned}$$

We use the retraction given by $R_X(\eta_X) = \mathrm{qf}(X + \eta_X)$, where $\mathrm{qf}(A)$ denotes the Q factor of decomposition of $A \in \mathbb{R}_*^{n \times p}$ as $A = QR$, where $\mathbb{R}_*^{n \times p}$ denotes the set of all nonsingular $n \times p$ matrices, $Q \in \mathrm{St}(p, n)$ and R is an upper triangular $n \times p$ matrix with strictly positive diagonal elements.

Vector transport (1) and its inverse on $\mathrm{St}(p, n)$ are given by

$$\mathcal{T}_{\eta_X}\xi_X = (I - YY^T)\xi_X + Y\mathrm{skew}(Y^T\xi_X)$$
$$(\mathcal{T}_{\eta_X})^{-1}\xi_Y = \xi_Y + \zeta,$$

where $Y := R_X(\eta_X)$, ζ is in the normal space at Y which implies $\zeta = YS$ where S is a symmetric matrix, and $(\xi_Y + YS) \in T_x\mathrm{St}(p,n)$ which implies $X^T(\xi_Y + YS)$ is skew symmetric. We therefore have

$$X^TYS + SY^TX + X^T\xi_Y + \xi_Y^TX = 0.$$

Therefore, S can be found by solving a Lyapunov equation.

For $\mathrm{St}(p,n)$, the parallel transport of $\xi \neq H$ along the geodesic $\gamma(t)$ from Y in direction H, denoted by $w(t) = P_\gamma^{t\leftarrow 0}\xi$, satisfies [8, §2.2.3]:

$$w'(t) = -\frac{1}{2}\gamma(t)(\gamma'(t)^Tw(t) + w(t)^T\gamma'(t)), \quad w(0) = \xi. \tag{12}$$

In practice, the differential equation is solved numerically and the computational cost of parallel transport may be significantly higher than that of vector transport.

5 Applications and numerical experiment results

We have experimented extensively with the versions of RBFGS described above. Here we present the results of two problems that provide leading evidence supporting the value of using retraction and vector transport in RBFGS and its limits. We obtained similar iteration counts using different x_0.

For a symmetric matrix A, the unit-norm eigenvector, v, corresponding to the smallest eigenvalue, defines the two global minima, $\pm v$, of the Rayleigh quotient $f : S^{n-1} \to \mathbb{R}$, $x \mapsto x^TAx$. The gradient of f is given by

$$\mathrm{grad}\, f(x) = 2\mathrm{P}_x(Ax) = 2(Ax - xx^TAx).$$

We show results of the minimization of the Rayleigh quotient to illustrate the performance of RBFGS on S^{n-1}.

On $\mathrm{St}(p,n)$ we consider a matrix Procrustes problem that minimizes the cost function $f : \mathrm{St}(p,n) \to \mathbb{R},\ X \to \|AX - XB\|_F$ given $n \times n$ and $p \times p$ matrices A and B respectively. The gradient of f on the submanifold of $\mathbb{R}^{n\times p}$ used to represent $\mathrm{St}(p,n)$ is

$$\mathrm{grad}\, f(X) = \mathrm{P}_X\mathrm{grad}\, \bar{f}(X) = Q - X\mathrm{sym}(X^TQ),$$
$$Q := A^TAX - A^TXB - AXB^T + XBB^T.$$

The versions of RBFGS that update B and B^{-1} perform similarly for these problems so we report data from the B^{-1} version. Approach 1 and Approach 2 display similar convergence behavior and on these manifolds Approach 2 has a higher computational complexity so we report data from Approach 1.

Table 1. Vector transport vs. Parallel transport

	Rayleigh		Procrustes	
	$n = 300$		$(n, p) = (12, 7)$	
	Vector	Parallel	Vector	Parallel
Time (sec.)	4.0	4.2	24.0	304.0
Iteration	97	95	83	175

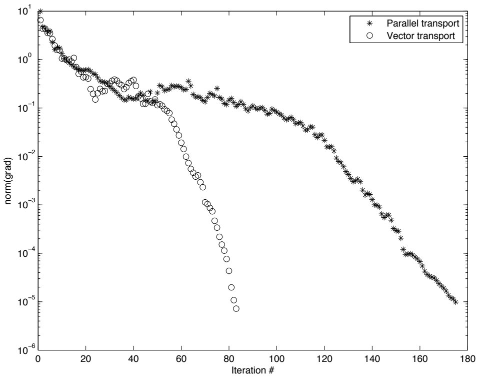

Fig. 2. Update of B^{-1}, Parallel and Vector Transport for Procrustes. n=12, p=7.

Since parallel transport and vector transport by projection have similar computational costs on S^{n-1}, the corresponding RBFGS versions have a similar computational cost per iteration. Therefore, we would expect any performance difference measured by time to reflect differences in rates of convergence. Columns 2 and 3 of Table 1 show that vector transport produces a convergence rate very close to parallel transport and the times are close as expected. This is encouraging from the point of view that the more flexible vector transport did not significantly degrade the convergence rate of RBFGS.

Given that vector transport by projection is significantly less expensive computationally than parallel transport on $\mathrm{St}(p, n)$, we would expect a significant improvement in performance as measured by time if the vector transport version manages to achieve a convergence rate similar to parallel transport. The times in columns 4 and 5 of Table 1 show an advantage to the vector transport version larger than the computational complexity predicts. The iteration counts provide an explanation. Encouragingly, the use of vector transport actually improves convergence compared to parallel transport. We note that the parallel transport version performs the required numerical integration of a differential equation with a stepsize sufficiently small so that decreasing it does not improve the convergence rate of RBFGS but no smaller to avoid unnecessary computations. Figure 2 illustrates in more detail the significant improvement in convergence rate achieved for vector transport. It provides strong evidence that a careful consideration of the choice of vector transport may have significant beneficial effects on both cost per step and overall convergence. More detailed consideration of this observation and the convergence theory for RBFGS will be presented in a future paper.

References

1. P.-A. Absil, R. Mahony, and R. Sepulchre (2010) Optimization on manifolds: methods and applications. In: Diehl M., Glineur F., Michiels W. (eds) Recent Trends in Optimization and its Applications in Engineering.
2. P.-A. Absil, C. G. Baker, and K. A. Gallivan (2007) Trust-region methods on Riemannian manifolds. Found. Comput. Math., 7(3):303–330
3. P.-A. Absil, R. Mahony, and R. Sepulchre (2008) Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton, NJ
4. Roy L. Adler, Jean-Pierre Dedieu, Joseph Y. Margulies, Marco Martens, and Mike Shub (2002) Newton's method on Riemannian manifolds and a geometric model for the human spine. IMA J. Numer. Anal., 22(3):359–390
5. N. Del Buono and C. Elia (2003) Computation of few Lyapunov exponents by geodesic based algorithms. Future Generation Computer systems, 19: 425-430
6. Ian Brace and Jonathan H. Manton (2006) An improved BFGS-on-manifold algorithm for computing weighted low rank approximations. In Proceedings of the 17h International Symposium on Mathematical Theory of Networks and Systems, pages 1735–1738
7. John E. Dennis, Jr. and Robert B. Schnabel (1983) Numerical methods for unconstrained optimization and nonlinear equations. Prentice Hall Series in Computational Mathematics, Prentice Hall Inc., Englewood Cliffs, NJ
8. Alan Edelman, Tomás A. Arias, and Steven T. Smith (1998) The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl., 20(2):303–353
9. D. Gabay (1982) Minimizing a differentiable function over a differential manifold. J. Optim. Theory Appl., 37(2):177–219
10. Jorge Nocedal and Stephen J. Wright (2006) Numerical optimization. Springer Series in Operations Research and Financial Engineering, Springer, New York, second edition
11. Berkant Savas and Lek-Heng Lim (2008) Best multilinear rank approximation of tensors with quasi-Newton methods on Grassmannians. Technical Report LITH-MAT-R-2008-01-SE, Department of Mathematics, Linköpings University

Acknowledgements

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors. This work was supported in part by “Communauté française de Belgique - Actions de Recherche Concertées”.