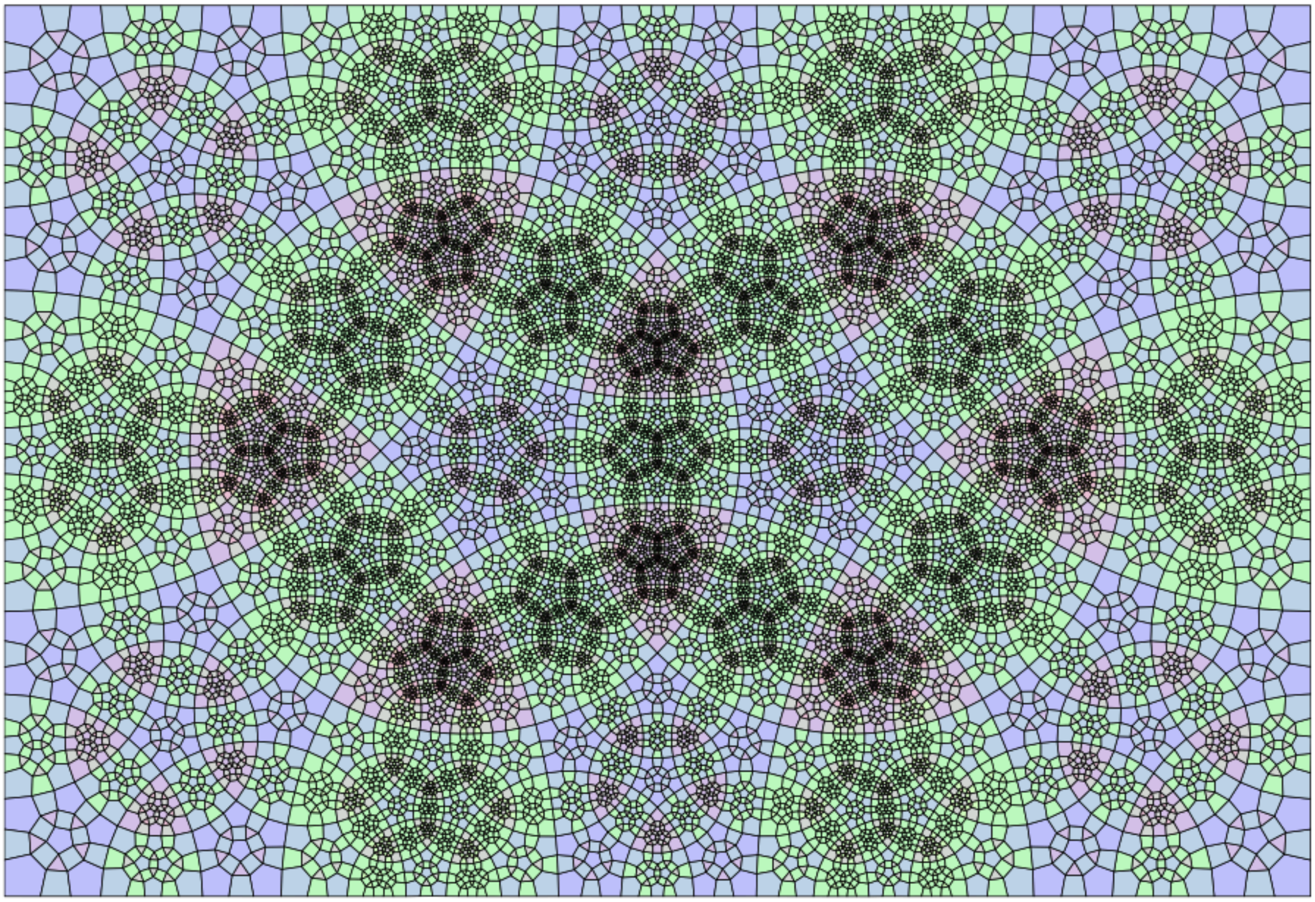
A Broyden Class of Quasi-Newton Methods for Riemannian Optimization

Wen Huang, K. A. Gallivan, P.-A. Absil



FSU15-04 — Florida State University
Mathematics Department

A BROYDEN CLASS OF QUASI-NEWTON METHODS FOR RIEMANNIAN OPTIMIZATION

WEN HUANG [†§], K. A. GALLIVAN [‡], AND P.-A. ABSIL [†]

Abstract. This paper develops and analyzes a generalization of the Broyden class of quasi-Newton methods to the problem of minimizing a smooth objective function f on a Riemannian manifold. A condition on vector transport and retraction that guarantees convergence and facilitates efficient computation is derived. Experimental evidence is presented demonstrating the value of the extension to the Riemannian Broyden class through superior performance for some problems compared to existing Riemannian BFGS methods, in particular those that depend on differentiated retraction.

Key words. Riemannian optimization; manifold optimization; Quasi-Newton methods; Broyden methods; Stiefel manifold;

AMS subject classifications. 65K05, 90C48, 90C53

1. Introduction. In the classical Euclidean setting, the Broyden class (see, e.g., [22, §6.3]) is a family of quasi-Newton methods that depend on a real parameter, ϕ. Its Hessian approximation update formula is $B_{k+1} = (1 - \phi_k)B_{k+1}^{\mathrm{BFGS}} + \phi_k B_{k+1}^{\mathrm{DFP}}$, where B_{k+1}^{BFGS} stands for the update obtained by the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method and B_{k+1}^{DFP} for the update of the Davidon–Fletcher–Powell (DFP) method. Therefore, all members of the Broyden class satisfy the well-known *secant equation*, central to many quasi-Newton methods. For many years, BFGS, $\phi = 0$, was the preferred member of the family, as it tends to perform better in numerical experiments. Analyzing the entire Broyden class was nevertheless a topic of interest in view of the insight that it gives into the properties of quasi-Newton methods; see [11] and the many references therein. Subsequently, it was found that negative values of ϕ are desirable [34, 10] and recent results reported in [19] indicate that a significant improvement can be obtained by exploiting the freedom offered by ϕ.

The problem of minimizing a smooth objective function f on a Riemannian manifold has been a topic of much interest over the past few years due to several important applications. Recently considered applications include matrix completion problems [8, 21, 12, 33], truss optimization [26], finite-element discretization of Cosserat rods [27], matrix mean computation [6, 4], and independent component analysis [31, 30]. Research efforts to develop and analyze optimization methods on manifolds can be traced back to the work of Luenberger [20]; they concern, among others, steepest-descent methods [20], conjugate gradients [32], Newton's method [32, 3], and trust-region methods [1, 5]; see also [2] for an overview.

The idea of quasi-Newton methods on manifolds is not new, however, the literature of which we are aware restricts consideration to the BFGS quasi-Newton method. Gabay [15] discussed a version using parallel transport on submanifolds of $\mathbb{R}^n$. Brace and Manton [9] applied a version on the Grassmann manifold to the problem of weighted low-rank approximations. Qi [25] compared the performance of different

[†] Department of Mathematical Engineering, ICTEAM Institute, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium.

[‡] Department of Mathematics, 208 Love Building, 1017 Academic Way, Florida State University, Tallahassee FL 32306-4510, USA.

[§] Corresponding author. E-mail: huwst08@gmail.com.

vector transports for a version of BFGS on Riemannian manifolds. Savas and Lim [28] proposed a BFGS and limited memory BFGS methods for problems with cost functions defined on a Grassmannian and applied the methods to the best multilinear rank approximation problem. Ring and Wirth [26] systematically analyzed a version of BFGS on Riemannian manifolds which requires differentiated retraction. Seibert et al. [29] discussed the freedom available when generalizing BFGS to Riemannian manifolds and analyzed one generalization of BFGS method on Riemannian manifolds that are isometric to $\mathbb{R}^n$.

In view of the above considerations, generalizing the Broyden family to manifolds is an appealing endeavor, which we undertake in this paper. For $\phi = 0$ (BFGS) the proposed algorithm is quite similar to the BFGS method of Ring and Wirth [26]. Notably, both methods rely on the framework of retraction and vector transport developed in [3, 2]. The BFGS method of [26] is more general in the sense that it also considers infinite-dimensional manifolds. On the other hand, a characteristic of our work is that we strive to resort as little as possible to the derivative of the retraction. Specifically, the definition of y_k (which corresponds to the usual difference of gradients) in [26] involves $\mathrm{D} f_{R_{x_k}}(s_k)$, whose Riesz representation is $(\mathrm{D} R_{x_k}(s_k))^* \nabla f|_{R_{x_k}(s_k)}$; here we use the notation of [26], i.e., R is the retraction, $f_R = f \circ R$, $s_k = R_{x_k}^{-1}(x_{k+1})$, $*$ represents the Riemannian adjoint operator, and ∇f is the Riemannian gradient. In contrast, our definition of y_k relies on the same isometric vector transport as the one that appears in the Hessian approximation update formula. This can be advantageous in situations where R is defined by means of a constraint restoration procedure that does not admit a closed-form expression. It may also be the case that the chosen R admits a closed-form expression but that its derivative is unknown to the user. The price to pay for using the isometric vector transport in y_k is satisfying a novel "locking condition". Fortunately, we show simple procedures that can produce a retraction or an isometric vector transport such that the pair satisfies the locking condition. As a result, efficient and convergent algorithms can be developed. Another contribution with respect to [26] is that we propose a limited-memory version of the quasi-Newton algorithm for large-scale problems.

The paper is organized as follows. The Riemannian Broyden (RBroyden) family of algorithms and the locking condition are defined in Section 2. A convergence analysis is presented in Section 3. Two methods of constructing an isometric vector transport and a method of constructing a retraction related to the locking condition are derived in Section 4. The limited-memory RBFGS is described in Section 5. The inverse Hessian approximation formula of Ring and Wirth's RBFGS is derived in Section 6 and numerical experiments are reported in Section 7 for all versions of the methods. Conclusions are presented and future work suggested in Section 8.

2. RBroyden family of methods.

The RBroyden family of methods is stated in Algorithm 1 and the details, in particular the Hessian approximation update formulas in Steps 6 and 7, are explained next. A basic background in differential geometry is assumed; such a background can be found, e.g., in [7, 2].

The concepts of retraction and vector transport can be found in [2] or [25]. Practically, the input statement of Algorithm 1 means the following. The retraction R is a C^2 mapping [2] from the tangent bundle $\mathrm{T}\mathcal{M}$ onto $\mathcal{M}$ such that (i) $R(0_x) = x$ for all $x \in \mathcal{M}$ (where 0_x denotes the origin of $\mathrm{T}_x\mathcal{M}$) and (ii) $\frac{d}{dt} R(t\xi_x)|_{t=0} = \xi_x$ for

[1] It can be shown using the Cauchy-Schwarz inequality that $u_k \geq 1$ and $u_k = 1$ if and only if there exists a constant κ such that $y_k = \kappa \mathcal{B}_k s_k$.

[2] The C^2 assumption is used in Lemma 3.5.

Algorithm 1 RBroyden family

```
Input: Riemannian manifold $\mathcal{M}$ with Riemannian metric $g$; retraction $R$; isometric vector transport $\mathcal{T}_{\mathrm{S}}$, with $R$ as associated retraction, that satisfies (2.8); continuously differentiable real-valued function $f$ on $\mathcal{M}$; initial iterate $x_0 \in \mathcal{M}$; initial Hessian approximation $\mathcal{B}_0$ which is a linear transformation of the tangent space $\mathrm{T}_{x_0}\mathcal{M}$ that is symmetric positive definite with respect to the metric $g$; convergence tolerance $\varepsilon > 0$; Wolfe condition constants $0 < c_1 < \frac{1}{2} < c_2 < 1$;
$k \leftarrow 0$;
while $\|\operatorname{grad} f(x_k)\| > \varepsilon$ do
    Obtain $\eta_k \in \mathrm{T}_{x_k}\mathcal{M}$ by solving $\mathcal{B}_k \eta_k = -\operatorname{grad} f(x_k)$;
    Set $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$ where $\alpha_k > 0$ is computed from a line search procedure to satisfy the Wolfe conditions
        (2.1)    $f(x_{k+1}) \leq f(x_k) + c_1 \alpha_k g(\operatorname{grad} f(x_k), \eta_k),$
        (2.2)    $\frac{d}{dt} f(R(t\eta_k))|_{t=\alpha_k} \geq c_2 \frac{d}{dt} f(R(t\eta_k)|_{t=0};$
    Set $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$;
    Define $s_k = \mathcal{T}_{\mathrm{S}_{\alpha_k \eta_k}} \alpha_k \eta_k$ and $y_k = \beta_k^{-1} \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\mathrm{S}_{\alpha_k \eta_k}} \operatorname{grad} f(x_k)$, where $\beta_k = \frac{\|\alpha_k \eta_k\|}{\|\mathcal{T}_{R_{\alpha_k \eta_k}} \alpha_k \eta_k\|}$ and $\mathcal{T}_R$ is the differentiated retraction of $R$;
    Define the linear operator $\mathcal{B}_{k+1} : \mathrm{T}_{x_{k+1}}\mathcal{M} \to \mathrm{T}_{x_{k+1}}\mathcal{M}$ by
        $\mathcal{B}_{k+1} p = \tilde{\mathcal{B}}_k p - \frac{g(s_k, \tilde{\mathcal{B}}_k p)}{g(s_k, \tilde{\mathcal{B}}_k s_k)} \tilde{\mathcal{B}}_k s_k + \frac{g(y_k, p)}{g(y_k, s_k)} y_k + \phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) g(v_k, p) v_k,$
    for all $p \in \mathrm{T}_{x_{k+1}}\mathcal{M}$ or equivalently
        (2.3)    $\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k - \frac{\tilde{\mathcal{B}}_k s_k (\tilde{\mathcal{B}}_k^* s_k)^\flat}{(\tilde{\mathcal{B}}_k^* s_k)^\flat s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k} + \phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) v_k v_k^\flat,$
    where $v_k = y_k / g(y_k, s_k) - \tilde{\mathcal{B}}_k s_k / g(s_k, \tilde{\mathcal{B}}_k s_k)$, $\phi_k$ is any number in the open interval $(\phi_k^c, \infty)$, $\tilde{\mathcal{B}}_k = \mathcal{T}_{\mathrm{S}_{\alpha_k \eta_k}} \circ \mathcal{B}_k \circ \mathcal{T}_{\mathrm{S}_{\alpha_k \eta_k}}^{-1}$, $\phi_k^c = 1/(1 - u_k)$, $u_k = (g(y_k, \tilde{\mathcal{B}}_k^{-1} y_k) g(s_k, \tilde{\mathcal{B}}_k s_k)) / g(y_k, s_k)^2$, $g(\cdot, \cdot)$ denotes the Riemannian metric and $a^\flat$ represents the flat of $a$, i.e., $a^\flat : \mathrm{T}_x\mathcal{M} \to \mathbb{R} : v \to g(a, v)$, $*$ denotes the adjoint with respect to $g$;[1]
    $k \leftarrow k + 1$;
end while
```

all $\xi_x \in \mathrm{T}_x\mathcal{M}$. The restriction of R to $\mathrm{T}_x\mathcal{M}$ is denoted by R_x. The domain of R does not need to be the whole tangent bundle, but in practice it often is, and in this paper we make the blanket assumption that R is defined wherever needed. A vector transport $\mathcal{T} : \mathrm{T}\mathcal{M} \oplus \mathrm{T}\mathcal{M} \to \mathrm{T}\mathcal{M}, (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}\xi_x$ with associated retraction R is a smooth mapping such that, for all (x, η_x) in the domain of R and all $\xi_x, \zeta_x \in \mathrm{T}_x\mathcal{M}$, it holds that (i) $\mathcal{T}_{\eta_x}\xi_x \in \mathrm{T}_{R(\eta_x)}\mathcal{M}$, (ii) $\mathcal{T}_{0_x}\xi_x = \xi_x$, (iii) $\mathcal{T}_{\eta_x}$ is a linear map. In Algorithm 1, the vector transport $\mathcal{T}_{\mathrm{S}}$ is isometric, i.e., it additionally satisfies (iv)

$$g_{R(\eta_x)}(\mathcal{T}_{\mathrm{S}_{\eta_x}}\xi_x, \mathcal{T}_{\mathrm{S}_{\eta_x}}\zeta_x) = g_x(\xi_x, \zeta_x). \tag{2.4}$$

In most practical cases, $\mathcal{T}_{\mathrm{S}_\eta}$ exists for all η, and we make this assumption throughout. In fact, we do not require that the isometric vector transport is smooth. The required properties are $\mathcal{T}_{\mathrm{S}} \in C^0$ and for any $\bar{x} \in \mathcal{M}$, there exists a neighborhood $\mathcal{U}$ of $\bar{x}$ and a constant c_0 such that for all $x, y \in \mathcal{U}$

$$\|\mathcal{T}_{\mathrm{S}_\eta} - \mathcal{T}_{R_\eta}\| \leq c_0 \|\eta\| \tag{2.5}$$

$$\|\mathcal{T}_{\mathrm{S}_\eta}^{-1} - \mathcal{T}_{R_\eta}^{-1}\| \leq c_0 \|\eta\| \tag{2.6}$$

where $\eta = R_x^{-1}(y)$, $\mathcal{T}_R$ denotes the differentiated retraction, i.e.,

$$\mathcal{T}_{R_{\eta_x}} \xi_x = \mathrm{D}\, R(\eta_x)[\xi_x] = \frac{d}{dt} R_x(\eta_x + t\xi_x)|_{t=0} \tag{2.7}$$

and $\| \cdot \|$ denotes the induced norm of the Riemannian metric g. In the following analysis, we use only these two properties of isometric vector transport.

In Algorithm 1, we require the isometric vector transport $\mathcal{T}_{\mathrm{S}}$ to satisfy the *locking condition*

$$\mathcal{T}_{\mathrm{S}_\xi} \xi = \beta \mathcal{T}_{R_\xi} \xi, \quad \beta = \frac{\|\xi\|}{\|\mathcal{T}_{R_\xi} \xi\|}, \tag{2.8}$$

for all $\xi \in \mathrm{T}_x \mathcal{M}$ and all $x \in \mathcal{M}$. Practical ways of building such a $\mathcal{T}_{\mathrm{S}}$ are discussed in Section 4. Observe that, throughout Algorithm 1, the differentiated retraction $\mathcal{T}_R$ only appears in the form $\mathcal{T}_{R_\xi} \xi$, which is equal to $\frac{d}{dt} R(t\xi)|_{t=1}$. Hence $\mathcal{T}_{R_{\alpha_k \eta_k}} \alpha_k \eta_k$ is just the velocity vector of the line search curve $\alpha \mapsto R(\alpha \eta_k)$ at time α_k and we are only required to be able to evaluate the differentiated retraction in the direction transported. The computational efficiency that results is also discussed in Section 4.

The isometry condition (2.4) and the locking condition (2.8) are imposed on $\mathcal{T}_{\mathrm{S}}$ notably because, as shown in Lemma 2.1, they ensure that the second Wolfe condition (2.2) implies $g(s_k, y_k) > 0$. Much as in the Euclidean case, it is essential that $g(s_k, y_k) > 0$, otherwise the secant condition $\mathcal{B}_{k+1} s_k = y_k$ cannot hold with $\mathcal{B}_{k+1}$ positive definite, whereas positive definiteness of the $\mathcal{B}_k$'s is key to guaranteeing that the search directions η_k are descent directions. It is possible to state Algorithm 1 without imposing the isometry and locking conditions, but then it becomes an open question whether the main convergence results would still hold. Clearly, some intermediate results would fail to hold and, assuming that the main results still hold, a completely different approach would probably be required to prove them.

When $\phi = 0$, the updating formula (2.3) reduces to the Riemannian BFGS formula of [25]. However, a crucial difference between Algorithm 1 and the Riemannian BFGS of [25] lies in the definition of y_k. Its definition in [25] corresponds to setting β_k to 1 instead of $\frac{\|\alpha_k \eta_k\|}{\|\mathcal{T}_{R_{\alpha_k \eta_k}} \alpha_k \eta_k\|}$. Our choice of β_k allows for a convergence analysis under more general assumptions than those of the convergence analysis of Qi [23]. Indeed, the convergence analysis of the Riemannian BFGS of [25], found in [23], assumes that retraction R is set to the exponential mapping and that vector transport $\mathcal{T}_{\mathrm{S}}$ is set to the parallel transport. These specific choices remain legitimate in Algorithm 1, hence the convergence analysis given here subsumes the one in [23]; however, several other choices become possible, as discussed in more detail in Section 4.

Lemma 2.1 proves that Algorithm 1 is well-defined for $\phi_k \in (\phi_k^c, \infty)$, where ϕ_k^c is defined in Step 7 of Algorithm 1.

Lemma 2.1. *Algorithm 1 constructs infinite sequences* $\{x_k\}$, $\{\mathcal{B}_k\}$, $\{\tilde{\mathcal{B}}_k\}$, $\{\alpha_k\}$, $\{s_k\}$, *and* $\{y_k\}$, *unless the stopping criterion in Step 2 is satisfied at some iteration.*

For all k, the Hessian approximation $\mathcal{B}_k$ is symmetric positive definite with respect to metric g, $\eta_k \neq 0$, and

$$g(s_k, y_k) \geq (c_2 - 1)\alpha_k g(\operatorname{grad} f(x_k), \eta_k). \tag{2.9}$$

Proof. We first show that (2.9) holds when all the involved quantities exist and $\eta_k \neq 0$. Define $m_k(t) = f(R_{x_k}(t\eta_k/\|\eta_k\|))$. We have

$$\begin{aligned}
g(s_k, y_k) &= g(\mathcal{T}_{\mathrm{S}_{\alpha_k\eta_k}} \alpha_k\eta_k, \beta_k^{-1} \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\mathrm{S}_{\alpha_k\eta_k}} \operatorname{grad} f(x_k)) \\
&= g(\mathcal{T}_{\mathrm{S}_{\alpha_k\eta_k}} \alpha_k\eta_k, \beta_k^{-1} \operatorname{grad} f(x_{k+1})) - g(\mathcal{T}_{\mathrm{S}_{\alpha_k\eta_k}} \alpha_k\eta_k, \mathcal{T}_{\mathrm{S}_{\alpha_k\eta_k}} \operatorname{grad} f(x_k)) \\
&= g(\beta_k^{-1}\mathcal{T}_{\mathrm{S}_{\alpha_k\eta_k}} \alpha_k\eta_k, \operatorname{grad} f(x_{k+1})) - g(\alpha_k\eta_k, \operatorname{grad} f(x_k)) \qquad \text{(by isometry)} \\
&= g(\mathcal{T}_{R_{\alpha_k\eta_k}} \alpha_k\eta_k, \operatorname{grad} f(x_{k+1})) - g(\alpha_k\eta_k, \operatorname{grad} f(x_k)) \qquad \text{(by (2.8))} \\
&= \alpha_k\|\eta_k\| \left(\frac{dm_k(\alpha_k\|\eta_k\|)}{dt} - \frac{dm_k(0)}{dt}\right). \qquad (2.10)
\end{aligned}$$

Note that guaranteeing (2.10), which will be used frequently, is the key reason for imposing the locking condition (2.8). From the second Wolfe condition (2.2), we have

$$\frac{dm_k(\alpha_k\|\eta_k\|)}{dt} \geq c_2 \frac{dm_k(0)}{dt}. \tag{2.11}$$

Therefore, it follows that

$$\frac{dm_k(\alpha_k\|\eta_k\|)}{dt} - \frac{dm_k(0)}{dt} \geq (c_2 - 1)\frac{dm_k(0)}{dt} = (c_2 - 1)\frac{1}{\|\eta_k\|} g(\operatorname{grad} f(x_k), \eta_k). \tag{2.12}$$

The claim (2.9) follows from (2.10) and (2.12).

When $\mathcal{B}_k$ is symmetric positive definite, η_k is a descent direction. Observe that the function $\alpha \mapsto f(R(\alpha\eta_k))$ remains a continuously differentiable function from $\mathbb{R}$ to $\mathbb{R}$ which is bounded below. Therefore, the classical result in [22, Lemma 3.1] guarantees the existence of a step size, α_k, that satisfies the Wolfe conditions.

The claims are proved by induction. They hold for $k = 0$ in view of the assumptions on $\mathcal{B}_0$ of Step 3 and of the results above. Assume now that the claims hold for some k. From (2.9), we have

$$\begin{aligned}
g(s_k, y_k) &\geq (1 - c_2)\alpha_k g(\operatorname{grad} f(x_k), -\eta_k) \\
&= (1 - c_2)\alpha_k g(\operatorname{grad} f(x_k), \mathcal{B}_k^{-1} \operatorname{grad} f(x_k)) > 0.
\end{aligned}$$

Recall that, in the Euclidean case, $s_k^T y_k > 0$ is a necessary and sufficient condition for the existence of a positive-definite secant update (see [14, Lemma 9.2.1]), and that BFGS is such an update [14, (9.2.10)]. From the generalization of these results to the Riemannian case (see [23, Lemmas 2.4.1 and 2.4.2]), it follows that $\mathcal{B}_{k+1}$ is symmetric positive definite when $\phi_k \equiv 0$.

Consider the function $h(\phi_k) : \mathbb{R} \to \mathbb{R}^d$ which gives the eigenvalues of $\mathcal{B}_{k+1}$. Since $\mathcal{B}_{k+1}$ is symmetric positive definite when $\phi_k \equiv 0$, we know all entries of $h(0)$ are greater than 0. By calculations similar to those for the Euclidean case ([10]), we have $\det(\mathcal{B}_{k+1}) = \det(\mathcal{B}_k)\frac{g(y_k, s_k)}{g(s_k, \mathcal{B}_k s_k)}(1 + \phi_k(u_k - 1))$, where ϕ_k and u_k are defined in Step 7 of Algorithm 1. So $\det(\mathcal{B}_{k+1}) = 0$ if and only if $\phi_k = \phi_k^c < 0$. In other words, $h(\phi_k)$ has one or more 0 entries if and only if $\phi_k = \phi_k^c$. In addition, since all entries of $h(0)$ are greater than 0 and $h(\phi_k)$ is a continuous function, we have that all entries of $h(\phi_k)$ are greater than 0 if and only if $\phi_k > \phi_k^c$. Therefore, the operator $\mathcal{B}_{k+1}$ is positive definite when $\phi_k > \phi_k^c$. Noting that the vector transport is isometric in (2.3), the symmetry of $\mathcal{B}_{k+1}$ is easily verified. □

3. Global convergence analysis.

In this section, global convergence is proven under a generalized convexity assumption and for $\phi_k \in [0, 1-\delta]$, where δ is any number in $(0, 1]$. The behavior of the Riemannian Broyden methods with ϕ_k not necessarily in this interval is explored in the experiments. Note the result derived in this section also guarantees local convergence to an isolated local minimizer.

3.1. Basic assumptions and definitions.

Throughout the convergence analysis, $\{x_k\}$, $\{\mathcal{B}_k\}$, $\{\tilde{\mathcal{B}}_k\}$, $\{\alpha_k\}$, $\{s_k\}$, $\{y_k\}$, and $\{\eta_k\}$, are infinite sequences generated by Algorithm 1, Ω denotes the sublevel set $\{x : f(x) \leq f(x_0)\}$, and x^* is a local minimizer of f in the level set Ω. The existence of such an x^* is guaranteed if Ω is compact, which happens, in particular, whenever the manifold $\mathcal{M}$ is compact.

Coordinate expressions in a neighborhood and in tangent spaces are used when appropriate. For an element of the manifold, $v \in \mathcal{M}$, $\hat{v} \in \mathbb{R}^d$ denotes the coordinates defined by a chart φ over a neighborhood $\mathcal{U}$, i.e., $\hat{v} = \varphi(v)$ for $v \in \mathcal{U}$. Coordinate expressions, $\hat{F}(x)$, for elements, $F(x)$, of a vector field F on $\mathcal{M}$ are written in terms of the canonical basis of the associated tangent space, $\mathrm{T}_x \mathcal{M}$, via the coordinate vector fields defined by the chart φ.

The convergence analysis depends on the property of (strong) retraction-convexity formalized in Definition 3.1 and the following three additional assumptions.

Definition 3.1. *For a function $f : \mathcal{M} \to \mathbb{R} : x \mapsto f(x)$ on a Riemannian manifold $\mathcal{M}$ with retraction R, define $m_{x,\eta}(t) = f(R_x(t\eta))$ for $x \in \mathcal{M}$ and $\eta \in \mathrm{T}_x \mathcal{M}$. The function f is retraction-convex with respect to the retraction R in a set $\mathcal{S}$ if for all $x \in \mathcal{S}$, all $\eta \in \mathrm{T}_x \mathcal{M}$ and $\|\eta\| = 1$, $m_{x,\eta}(t)$ is convex for all t which satisfy $R_x(\tau\eta) \in \mathcal{S}$ for all $\tau \in [0, t]$. Moreover, f is strongly retraction-convex in $\mathcal{S}$ if $m_{x,\eta}(t)$ is strongly convex, i.e., there exist two constants $0 < a_0 < a_1$ such that $a_0 \leq \frac{d^2 m_{x,\eta}}{dt^2}(t) \leq a_1$, for all $x \in \mathcal{S}$, all $\|\eta\| = 1$ and all t such that $R_x(\tau\eta) \in \mathcal{S}$ for all $\tau \in [0, t]$.*

Assumption 3.1. *The objective function f is twice continuously differentiable.*

Let $\tilde{\Omega}$ be a neighborhood of x^* and ρ be a positive constant such that, for all $y \in \tilde{\Omega}$, $\tilde{\Omega} \subset R_y(\mathbb{B}(0_y, \rho))$ and $R_y(\cdot)$ is a diffeomorphism on $\mathbb{B}(0_y, \rho)$. The existence of $\tilde{\Omega}$, termed ρ-totally retractive neighborhood of x^*, is guaranteed [18, §3.3]. Shrinking $\tilde{\Omega}$ if necessary, further assume that it is an R-star shaped neighborhood of x^*, i.e., $R_{x^*}(tR_{x^*}^{-1}(x)) \in \tilde{\Omega}$ for all $x \in \tilde{\Omega}$ and $t \in [0, 1]$. The next two assumptions are a Riemannian generalization of a weakened version of [11, Assumption 2.1]: in the Euclidean setting ($\mathcal{M}$ is the Euclidean space $\mathbb{R}^n$ and the retraction R is the standard one, i.e., $R_x(\eta) = x + \eta$), if [11, Assumption 2.1] holds, then the next two assumptions are satisfied by letting $\tilde{\Omega}$ be the sublevel set Ω.

Assumption 3.2. *The iterates x_k stay continuously in $\tilde{\Omega}$, meaning that $R_{x_k}(t\eta_k) \in \tilde{\Omega}$ for all $t \in [0, \alpha_k]$.*

Observe that, in view of Lemma 2.1 ($\mathcal{B}_k$ remains symmetric positive definite with respect to g), for all $K > 0$, the sequences $\{x_k\}$, $\{\mathcal{B}_k\}$, $\{\tilde{\mathcal{B}}_k\}$, $\{\alpha_k\}$, $\{s_k\}$, $\{y_k\}$, and $\{\eta_k\}$, for $k \geq K$, are still generated by Algorithm 1. Hence, Assumption 3.2 amounts to requiring that the iterates x_k eventually stay continuously in $\tilde{\Omega}$. Note that Assumption 3.2 can not be removed. To see this, consider for example the unit sphere with the exponential retraction, where we can have $x_k = x_{k+1}$ with $\|\alpha_k \eta_k\| = 2\pi$. (A similar comment was made in [18] before Lemma 3.6.)

Assumption 3.3. *f is strongly retraction-convex (Definition 3.1) with respect to the retraction R in $\tilde{\Omega}$.*

The definition of retraction-convexity generalizes standard Euclidean and Riemannian concepts. It is easily seen that (strong) retraction-convexity reduces to (strong) convexity when the function is defined on Euclidean space. It can be shown that when R is the exponential mapping, (strong) retraction-convexity is ordinary (strong) convexity for a C^2 function based on the definiteness of its Hessian . It also can be shown that a set $\mathcal{S}$ as in Definition 3.1 always exists around a nondegenerate local minimizer of a C^2 function (Lemma 3.1).

Lemma 3.1. *Suppose Assumption 3.1 holds and* $\operatorname{Hess} f(x^*)$ *is positive definite. Define $m_{x,\eta}(t) = f(R_x(t\eta))$. Then there exists a ϱ-totally retractive neighborhood $\mathcal{N}$ of x^* and two constants $0 < \tilde{a}_0 < \tilde{a}_1$ such that*

$$\tilde{a}_0 \leq \frac{d^2 m_{x,\eta}}{dt^2}(t) \leq \tilde{a}_1, \tag{3.1}$$

for all $x \in \mathcal{N}$, $\eta \in \mathrm{T}_x \mathcal{M}$, $\|\eta\| = 1$ and $t < \varrho$.

Proof. By definition, we have $\frac{d}{dt} f(R_x(t\eta)) = g(\operatorname{grad} f(R_x(t\eta)), \mathrm{D}\, R_x(t\eta)[\eta])$ and

$$\begin{aligned}\frac{d^2}{dt^2} f(R_x(t\eta)) =& g(\operatorname{Hess} f(R_x(t\eta)))[\mathrm{D}\, R_x(t\eta)[\eta]], \mathrm{D}\, R_x(t\eta)[\eta]) \\ &+ g(\operatorname{grad} f(R_x(t\eta)), \frac{\mathrm{D}}{dt} \mathrm{D}\, R_x(t\eta)[\eta]),\end{aligned}$$

where the definition of $\frac{\mathrm{D}}{dt}$ can be found in [2, §5.4]. Since $\mathrm{D}\, R_x(t\eta)[\eta]|_{t=0} = \eta$ and $\operatorname{grad} f(R_x(t\eta))|_{t=0,x=x^*} = 0_x$, it holds that $\frac{d^2}{dt^2} f(R_x(t\eta))|_{t=0,x=x^*} = g(\operatorname{Hess} f(x^*)[\eta], \eta)$. In addition, since $\operatorname{Hess} f(x^*)$ is positive definite, there exist two constants $0 < b_0 < b_1$ such that inequalities $b_0 < \frac{d^2}{dt^2} f(R_x(t\eta))|_{t=0,x=x^*} < b_1$ hold for all $\eta \in \mathrm{T}_{x^*} \mathcal{M}$ and $\|\eta\| = 1$. Note that $\frac{d^2}{dt^2} f(R_x(t\eta))$ is continuous with respect to (x, t, η). Therefore, there exist a neighborhood $\mathcal{U}$ of x^* and a neighborhood $\mathcal{V}$ of 0 such that $b_0/2 < \frac{d^2}{dt^2} f(R_x(t\eta)) < 2b_1$ for all $(x, t) \in \mathcal{U} \times \mathcal{V}$, $\eta \in \mathrm{T}_x \mathcal{M}$ and $\|\eta\| = 1$. Choose $\varrho > 0$ such that $[-\varrho, \varrho] \subset \mathcal{V}$ and let $\mathcal{N} \subset \mathcal{U}$ be a ϱ-totally retractive neighborhood of x^*. □

3.2. Preliminary lemmas. The lemmas in this section provide the results needed to show global convergence stated in Theorem 3.1. The strategy generalizes that for the Euclidean case in [11]. Where appropriate, comments are included indicating important adaptations of the reasoning to Riemannian manifolds. The main adaptations originate from the fact that the Euclidean analysis exploits the relationship $y_k = \bar{G}_k s_k$ ([11, (2.3)]), where $\bar{G}$ is a average Hessian of f, and this relationship does not gracefully generalize to Riemannian manifolds (unless the isometric vector transport $\mathcal{T}_{\mathrm{S}}$ in Algorithm 1 is chosen as the parallel transport, a choice we often want to avoid in view of its computational cost). This difficulty requires alternate approaches in several proofs. The key idea of the approaches is to make use of the scaled function $m_{x,\eta}(t)$ rather than f, a well-known strategy in the Riemannian setting.

The first result, Lemma 3.2, is used to prove Lemma 3.4.

Lemma 3.2. *If Assumptions 3.1, 3.2 and 3.3 hold then there exists a constant $a_0 > 0$ such that*

$$\frac{1}{2} a_0 \|s_k\|^2 \leq (c_1 - 1)\alpha_k g(\operatorname{grad} f(x_k), \eta_k). \tag{3.2}$$

Constant a_0 *can be chosen as in Definition 3.1.*

Proof. In Euclidean space, Taylor's Theorem is used to characterize a function around a point. A generalization of Taylor's formula to Riemannian manifolds was proposed in [32, Remark 3.2], but it is restricted to the exponential mapping rather than allowing for an arbitrary retraction. This difficulty is overcome by defining a function on a curve on the manifold and applying Taylor's Theorem. Define $m_k(t) = f(R_{x_k}(t\eta_k/\|\eta_k\|))$. Since $f \in C^2$ is strongly retraction-convex on $\tilde{\Omega}$ by Assumption 3.3, there exist constants $0 < a_0 < a_1$ such that $a_0 \leq \frac{d^2 m_{x,\eta}(t)}{dt^2} \leq a_1$ for all $t \in [0, \alpha_k\|\eta_k\|]$. From Taylor's theorem, we know

$$\begin{aligned} f(x_{k+1}) - f(x_k) = m_k(\alpha_k\|\eta_k\|) - m_k(0) &= \frac{dm_k(0)}{dt}\alpha_k\|\eta_k\| + \frac{1}{2}\frac{d^2 m_k(p)}{dt^2}(\alpha_k\|\eta_k\|)^2 \\ &= g(\operatorname{grad} f(x_k), \alpha_k\eta_k) + \frac{1}{2}\frac{d^2 m_k(p)}{dt^2}(\alpha_k\|\eta_k\|)^2 \\ &\geq g(\operatorname{grad} f(x_k), \alpha_k\eta_k) + \frac{1}{2}a_0(\alpha_k\|\eta_k\|)^2, \end{aligned} \tag{3.3}$$

where $0 \leq p \leq \alpha_k\|\eta_k\|$. Using (3.3), the first Wolfe condition (2.1) and that $\|s_k\| = \alpha_k\|\eta_k\|$, we obtain $(c_1 - 1)g(\operatorname{grad} f(x_k), \alpha_k\eta_k) \geq a_0\|s_k\|^2/2$. □

Lemma 3.3 generalizes [11, (2.4)].

Lemma 3.3. *If Assumptions 3.1, 3.2 and 3.3 hold then there exist two constants* $0 < a_0 \leq a_1$ *such that*

$$a_0 g(s_k, s_k) \leq g(s_k, y_k) \leq a_1 g(s_k, s_k), \tag{3.4}$$

for all k*. Constants* a_0 *and* a_1 *can be chosen as in Definition 3.1.*

Proof. In the Euclidean case of [11, (2.4)], the proof follows easily from the convexity of the cost function and the resulting positive definiteness of the Hessian over the entire relevant set. The Euclidean proof exploits the relationship $y_k = \bar{G}_k s_k$, where $\bar{G}_k$ is the average Hessian and that $\bar{G}_k$ must be positive definite to bound the inner product $s_k^T y_k$ using the largest and smallest eigenvalues that can in turn be bounded on the relevant set. We do not have this property on a Riemannian manifold but the locking condition, retraction-convexity and replacing the average Hessian with a quantity derived from a function defined on a curve on the manifold allows the generalization.

Define $m_k(t) = f(R_{x_k}(t\eta_k/\|\eta_k\|))$. Using the locking condition (2.10) and Taylor's Theorem yields

$$g(s_k, y_k) = \alpha_k\|\eta_k\|(\frac{dm(\alpha_k\|\eta_k\|)}{dt} - \frac{dm(0)}{dt}) = \alpha_k\|\eta_k\|\int_0^{\alpha_k\|\eta_k\|}\frac{d^2 m}{dt^2}(s)ds$$

and since $g(s_k, s_k) = \alpha_k^2\|\eta_k\|^2$, we have

$$\frac{g(s_k, y_k)}{g(s_k, s_k)} = \frac{1}{\alpha_k\|\eta_k\|}\int_0^{\alpha_k\|\eta_k\|}\frac{d^2 m}{dt^2}(s)ds.$$

By Assumption 3.3, it follows that $a_0 \leq \frac{g(s_k, y_k)}{g(s_k, s_k)} \leq a_1$. □

Lemma 3.4 generalizes [11, Lemma 2.1].

Lemma 3.4. *Suppose Assumptions 3.1, 3.2 and 3.3 hold. Then there exist two constants* $0 < a_2 < a_3$ *such that*

$$a_2\|\operatorname{grad} f(x_k)\|\cos\theta_k \leq \|s_k\| \leq a_3\|\operatorname{grad} f(x_k)\|\cos\theta_k, \tag{3.5}$$

for all k, where $\cos\theta_k = \frac{-g(\operatorname{grad} f(x_k),\eta_k)}{\|\operatorname{grad} f(x_k)\|\|\eta_k\|}$, i.e., θ_k is the angle between the search direction η_k and the steepest descent direction, $-\operatorname{grad} f(x_k)$.

Proof. Define $m_k(t) = f(R_{x_k}(t\eta_k/\|\eta_k\|))$. By (2.9) of Lemma 2.1, we have

$$g(s_k, y_k) \geq \alpha_k(c_2-1)g(\operatorname{grad} f(x_k),\eta_k) = \alpha_k(1-c_2)\|\operatorname{grad} f(x_k)\|\|\eta_k\|\cos\theta_k.$$

Using (3.4) and noticing $\|\alpha_k\eta_k\| = \|s_k\|$, we know $\|s_k\| \geq a_2\|\operatorname{grad} f(x_k)\|\cos\theta_k$, where $a_2 = (1-c_2)/a_1$, proving the left inequality.

By (3.2) of Lemma 3.2, we have $(c_1-1)g(\operatorname{grad} f(x_k),\alpha_k\eta_k) \geq a_0\|s_k\|^2/2$. Noting $\|s_k\| = \alpha_k\|\eta_k\|$ and by the definition of $\cos\theta_k$, we have $\|s_k\| \leq a_3\|\operatorname{grad} f(x_k)\|\cos\theta_k$, where $a_3 = 2(1-c_1)/a_0$. □

Lemma 3.5 is needed to prove Lemmas 3.6 and 3.9. Lemma 3.5 gives a Lipschitz-like relationship between two related vector transports applied to the same tangent vector. $\mathcal{T}_1$ below plays the same role as $\mathcal{T}_{\mathrm{S}}$ in Algorithm 1.

Lemma 3.5. *Let $\mathcal{M}$ be a Riemannian manifold endowed with two vector transports $\mathcal{T}_1 \in C^0$ and $\mathcal{T}_2 \in C^\infty$ where $\mathcal{T}_1$ satisfies (2.5) and (2.6) and both transports are associated with a same retraction R. Then for any $\bar{x} \in \mathcal{M}$ there exists a constant $a_4 > 0$ and a neighborhood of $\bar{x}$, $\mathcal{U}$, such that for all $x, y \in \mathcal{U}$*

$$\|\mathcal{T}_{1_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \leq a_4\|\xi\|\|\eta\|,$$

where $\eta = R_x^{-1}y$ and $\xi \in T_x\mathcal{M}$.

Proof. $L_R(\hat{x},\hat{\eta})$ and $L_2(\hat{x},\hat{\eta})$ denote coordinate form of $\mathcal{T}_{R_\eta}$ and $\mathcal{T}_{2_\eta}$ respectively. Since all norms are equivalent for a finite dimensional space, there exists $0 < b_1(x) < b_2(x)$ such that $b_1(x)\|\xi_x\| \leq \|\hat{\xi}_x\|_2 \leq b_2(x)\|\xi_x\|$ for all $x \in \mathcal{U}$, where $\|\cdot\|_2$ denotes the Euclidean norm, i.e., 2-norm. By choosing $\mathcal{U}$ compact, it follows that $b_1\|\xi_x\| \leq \|\hat{\xi}_x\|_2 \leq b_2\|\xi_x\|$ for all $x \in \mathcal{U}$ where $0 < b_1 < b_2$, $b_1 = \min_{x\in\mathcal{U}} b_1(x)$ and $b_2 = \max_{x\in\mathcal{U}} b_2(x)$. It follows that

$$\begin{aligned}
&\|\mathcal{T}_{1_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \\
&= \|\mathcal{T}_{1_\eta}\xi - \mathcal{T}_{R_\eta}\xi + \mathcal{T}_{R_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \leq b_0\|\eta\|\|\xi\| + \|\mathcal{T}_{R_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \\
&\leq b_0\|\eta\|\|\xi\| + \frac{1}{b_1}\|(L_R(\hat{x},\hat{\eta}) - L_2(\hat{x},\hat{\eta}))\hat{\xi}\|_2 \\
&\leq b_0\|\eta\|\|\xi\| + \frac{1}{b_1}\|\hat{\xi}\|_2\|L_R(\hat{x},\hat{\eta}) - L_2(\hat{x},\hat{\eta})\|_2 \\
&\leq b_0\|\eta\|\|\xi\| + b_3\|\hat{\xi}\|_2\|\hat{\eta}\|_2 \\
&\qquad \text{(since } L_R(\hat{x},0) = L_2(\hat{x},0) \text{ and } L_2 \text{ is smooth and } L_R \in C^1 \text{ because } R \in C^2.\text{)} \\
&= b_4\|\xi\|\|\eta\|
\end{aligned}$$

where b_0, b_3, b_4 are positive constants. □

Lemma 3.6 is a consequence of Lemma 3.5.

Lemma 3.6. *Let $\mathcal{M}$ be a Riemannian manifold endowed with a retraction R whose differentiated retraction is denoted $\mathcal{T}_R$. Let $\bar{x} \in \mathcal{M}$. Then there is a neighborhood $\mathcal{U}$ of $\bar{x}$ and constant $\tilde{a}_4 > 0$ such that for all $x, y \in \mathcal{U}$, any $\xi \in \mathrm{T}_x\mathcal{M}$ with $\|\xi\| = 1$, the effect of the differentiated retraction is bounded with*

$$|\|\mathcal{T}_{R_\eta}\xi\| - 1| \leq \tilde{a}_4\|\eta\|,$$

where $\eta = R_x^{-1}y$.

Proof. Applying Lemma 3.5 with $\mathcal{T}_1 = \mathcal{T}_R$ and $\mathcal{T}_2$ isometric, we have $\|\mathcal{T}_{R_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \leq b_0\|\xi\|\|\eta\|$, where b_0 is a positive constant. Noticing $\|\xi\| = 1$ and that $\|\cdot\|$ is the induced norm, we have, by an application of the triangle inequality, $b_0\|\eta\| \geq \|\mathcal{T}_{R_\eta}\xi - \mathcal{T}_{2_\eta}\xi\| \geq \|\mathcal{T}_{R_\eta}\xi\| - \|\mathcal{T}_{2_\eta}\xi\| = \|\mathcal{T}_{R_\eta}\xi\| - 1$. Similarly, we have $b_0\|\eta\| \geq \|\mathcal{T}_{2_\eta}\xi - \mathcal{T}_{R_\eta}\xi\| \geq \|\mathcal{T}_{2_\eta}\xi\| - \|\mathcal{T}_{R_\eta}\xi\| = 1 - \|\mathcal{T}_{R_\eta}\xi\|$ to complete the proof. □

Lemma 3.7 generalizes [11, (2.13)] and implies a generalization of the Zoutendijk Condition [22, Theorem 3.2], i.e., if $\cos\theta_k$ does not approach 0, then according to this lemma, the algorithm is convergent.

Lemma 3.7. *Suppose Assumptions 3.1, 3.2 and 3.3 hold. Then there exists a constant* $a_5 > 0$ *such that for all* k

$$f(x_{k+1}) - f(x^*) \leq (1 - a_5\cos^2\theta_k)(f(x_k) - f(x^*)),$$

where $\cos\theta_k = \frac{-g(\operatorname{grad} f(x_k),\eta_k)}{\|\operatorname{grad} f(x_k)\|\|\eta_k\|}$.

Proof. The original proof in [11, (2.13)] uses the average Hessian. As when proving Lemma 3.2, this is replaced by considering a function defined on a curve on the manifold. Let $z_k = \|R_{x^*}^{-1}x_k\|$ and $\zeta_k = (R_{x^*}^{-1}x_k)/z_k$. Define $m_k(t) = f(R_{x^*}(t\zeta_k))$. From Taylor's Theorem, we have

$$m_k(0) - m_k(z_k) = \frac{dm_k(z_k)}{dt}(0 - z_k) + \frac{1}{2}\frac{d^2m_k(p)}{dt^2}(0 - z_k)^2, \tag{3.6}$$

where p is some number between 0 and z_k. Notice that x^* is the minimizer, so $m_k(0) - m_k(z_k) \leq 0$. Also note that $a_0 \leq \frac{d^2m_k(p)}{dt^2} \leq a_1$ by Assumption 3.3 and Definition 3.1. We have

$$\frac{dm_k(z_k)}{dt} \geq \frac{1}{2}a_0z_k. \tag{3.7}$$

Still using (3.6) and noticing that $\frac{d^2m_k(p)}{dt^2}(0 - z_k)^2 \geq 0$, we have

$$f(x_k) - f(x^*) \leq \frac{dm_k(z_k)}{dt}z_k. \tag{3.8}$$

Combining (3.7) and (3.8) and noticing that $\frac{dm_k(z_k)}{dt} = g(\operatorname{grad} f(x_k), \mathcal{T}_{R_{z_k\zeta_k}}\zeta_k)$, we have $f(x_k) - f(x^*) \leq \frac{2}{a_0}g^2(\operatorname{grad} f(x_k), \mathcal{T}_{R_{z_k\zeta_k}}\zeta_k)$ and

$$f(x_k) - f(x^*) \leq \frac{2}{a_0}\|\operatorname{grad} f(x_k)\|^2\|\mathcal{T}_{R_{z_k\zeta_k}}\zeta_k\|^2 \leq b_0\|\operatorname{grad} f(x_k)\|^2, \tag{3.9}$$

by Lemma 3.6 and the assumptions on $\tilde{\Omega}$, where b_0 is a positive constant. Using (3.5), the first Wolfe condition (2.1) and the definition of $\cos\theta_k$, we obtain $f(x_{k+1}) - f(x_k) \leq -b_1\|\operatorname{grad} f(x_k)\|^2\cos^2\theta_k$, where b_1 is some positive constant. Using (3.9), we obtain $f(x_{k+1}) - f(x^*) \leq f(x_{k+1}) - f(x_k) + f(x_k) - f(x^*) \leq -b_1\|\operatorname{grad} f(x_k)\|^2\cos^2\theta_k + f(x_k) - f(x^*) \leq -\frac{b_1}{b_0}\cos^2\theta_k(f(x_k) - f(x^*)) + f(x_k) - f(x^*) = (1 - a_5\cos^2\theta_k)(f(x_k) - f(x^*))$, where $a_5 = b_1/b_0$ is a positive constant. □

Lemma 3.8 generalizes [11, Lemma 2.2].

Lemma 3.8. *Suppose Assumptions 3.1, 3.2 and 3.3 hold. Then there exist two constants* $0 < a_6 < a_7$ *such that*

$$a_6\frac{g(s_k, \tilde{\mathcal{B}}_ks_k)}{\|s_k\|^2} \leq \alpha_k \leq a_7\frac{g(s_k, \tilde{\mathcal{B}}_ks_k)}{\|s_k\|^2}, \tag{3.10}$$

for all k.

Proof. We have

$$\begin{aligned}(1-c_2)g(s_k,\tilde{\mathcal{B}}_k s_k) &= (1-c_2)g(\alpha_k\eta_k,\alpha_k\mathcal{B}_k\eta_k)\\ &= (c_2-1)\alpha_k^2 g(\eta_k,\operatorname{grad} f(x_k)) \text{ (by Step 3 of Algorithm 1)}\\ &\leq \alpha_k g(s_k,y_k) \text{ (by (2.9) of Lemma 2.1)}\\ &\leq \alpha_k a_1\|s_k\|^2 \text{ (by (3.4))}.\end{aligned}$$

Therefore, we have $\alpha_k \geq a_6\frac{g(s_k,\tilde{\mathcal{B}}_k s_k)}{\|s_k\|^2}$, where $a_6=(1-c_2)/a_1$, giving the left inequality.

By (3.2) of Lemma 3.2 and $\|s_k\|=\alpha_k\|\eta_k\|$, we have $(c_1-1)\alpha_k g(\operatorname{grad} f(x_k),\eta_k)\geq \frac{1}{2}a_0\|s_k\|^2$. It follows that

$$(c_1-1)\alpha_k g(\operatorname{grad} f(x_k),\eta_k) = (1-c_1)g(\mathcal{B}_k\eta_k,\alpha_k\eta_k) = \frac{1-c_1}{\alpha_k}g(s_k,\tilde{\mathcal{B}}_k s_k).$$

Therefore, we have $\alpha_k \leq a_7\frac{g(s_k,\tilde{\mathcal{B}}_k s_k)}{\|s_k\|^2}$, where $a_7=2(1-c_1)/a_0$, giving the right inequality. □

Lemma 3.9 generalizes [11, Lemma 3.1, Equation (3.3)].

Lemma 3.9. *Suppose Assumption 3.1 holds. Then, for all k there exists a constant $a_9>0$ such that*

$$g(y_k,y_k)\leq a_9 g(s_k,y_k). \tag{3.11}$$

Proof. Define $y_k^{\mathrm{P}}=\operatorname{grad} f(x_{k+1})-P_{\gamma_k}^{1\leftarrow 0}\operatorname{grad} f(x_k)$, where $\gamma_k(t)=R_{x_k}(t\alpha_k\eta_k)$, i.e., the retraction line from x_k to x_{k+1} and P is the parallel transport along $\gamma_k(t)$. Details about the definition of parallel transport can be found, e.g., in [2, §5.4]. From [18, Lemma 8], we have $\|P_{\gamma_k}^{0\leftarrow 1}y_k^{\mathrm{P}}-\bar{H}_k\alpha_k\eta_k\|\leq b_0\|\alpha_k\eta_k\|^2=b_0\|s_k\|^2$, where $\bar{H}_k=\int_0^1 P_{\gamma_k}^{0\leftarrow t}\operatorname{Hess} f(\gamma_k(t))P_{\gamma_k}^{t\leftarrow 0}dt$ and $b_0>0$. It follows that

$$\begin{aligned}\|y_k\| &\leq \|y_k-y_k^{\mathrm{P}}\|+\|y_k^{\mathrm{P}}\| = \|y_k-y_k^{\mathrm{P}}\|+\|P_{\gamma_k}^{0\leftarrow 1}y_k^{\mathrm{P}}\|\\ &\leq \|y_k-y_k^{\mathrm{P}}\|+\|P_{\gamma_k}^{0\leftarrow 1}y_k^{\mathrm{P}}-\bar{H}_k\alpha_k\eta_k\|+\|\bar{H}_k\alpha_k\eta_k\|\\ &\leq \|\operatorname{grad} f(x_{k+1})/\beta_k-\mathcal{T}_{S_{\alpha_k\eta_k}}\operatorname{grad} f(x_k)-\operatorname{grad} f(x_{k+1})+P_{\gamma_k}^{1\leftarrow 0}\operatorname{grad} f(x_k)\|\\ &\quad +\|\bar{H}_k\alpha_k\eta_k\|+b_0\|s_k\|^2\\ &\leq \|\operatorname{grad} f(x_{k+1})/\beta_k-\operatorname{grad} f(x_{k+1})\|+\|P_{\gamma_k}^{1\leftarrow 0}\operatorname{grad} f(x_k)-\mathcal{T}_{S_{\alpha_k\eta_k}}\operatorname{grad} f(x_k)\|\\ &\quad +\|\bar{H}_k\alpha_k\eta_k\|+b_0\|s_k\|^2\\ &\leq b_1\|\operatorname{grad} f(x_{k+1})\|+b_2\|s_k\|\|\operatorname{grad} f(x_k)\| \text{ (by Lemmas 3.5 and 3.6)}\\ &\quad +b_3\|s_k\|+b_0\|s_k\| \text{ (since } \|\operatorname{Hess} f\| \text{ is bounded above in a compact set)}\\ &\leq b_4\|s_k\| \text{ ([16, Lemma 14.5] and [18, Lemma 3.3])}\end{aligned}$$

where b_1, b_2, b_3 and $b_4>0$. Therefore, by Lemma 3.3, we have $\frac{g(y_k,y_k)}{g(s_k,y_k)}\leq\frac{g(y_k,y_k)}{a_0 g(s_k,s_k)}\leq\frac{b_1^2}{a_0}$ giving the desired result. □

Lemma 3.10 generalizes [11, Lemma 3.1] and as with the earlier lemmas the proof does not use an average Hessian.

Lemma 3.10. *Suppose Assumptions 3.1, 3.2 and 3.3 hold. Then there exist* *constants* $a_{10} > 0,\ a_{11} > 0,\ a_{12} > 0$ *such that*

$$\frac{g(s_k, \tilde{\mathcal{B}}_k s_k)}{g(s_k, y_k)} \leq a_{10}\alpha_k \tag{3.12}$$

$$\frac{\|\tilde{\mathcal{B}}_k s_k\|^2}{g(s_k, \tilde{\mathcal{B}}_k s_k)} \geq a_{11}\frac{\alpha_k}{\cos^2\theta_k} \tag{3.13}$$

$$\frac{|g(y_k, \tilde{\mathcal{B}}_k s_k)|}{g(y_k, s_k)} \leq a_{12}\frac{\alpha_k}{\cos\theta_k} \tag{3.14}$$

for all k.

Proof. By (2.9) of Lemma 2.1, we have $g(s_k, y_k) \geq (c_2 - 1) g(\operatorname{grad} f(x_k), \alpha_k \eta_k)$. So by the Step 3 of Algorithm 1, we obtain $g(s_k, y_k) \geq \frac{(1-c_2)}{\alpha_k} g(s_k, \tilde{\mathcal{B}} s_k)$ and therefore $\frac{g(s_k, \tilde{\mathcal{B}}_k s_k)}{g(s_k, y_k)} \leq a_{10}\alpha_k$, where $a_{10} = 1/(1 - c_2)$, proving (3.12).

Inequality (3.13) follows from

$$\begin{aligned}\frac{\|\tilde{\mathcal{B}}_k s_k\|^2}{g(s_k, \tilde{\mathcal{B}}_k s_k)} &= \frac{\alpha_k^2 \|\operatorname{grad} f(x_k)\|^2}{\alpha_k \|s_k\| \|\operatorname{grad} f(x_k)\| \cos\theta_k} \\ &\qquad \text{(by Step 3 of Algorithm 1 and the definition of } \cos\theta_k) \\ &= \frac{\alpha_k \|\operatorname{grad} f(x_k)\|}{\|s_k\| \cos\theta_k} \geq a_{11}\frac{\alpha_k}{\cos^2\theta_k}, \text{ (by (3.5))}\end{aligned}$$

where $a_{11} > 0$.

Finally, inequality (3.14) follows from

$$\begin{aligned}\frac{|g(y_k, \tilde{\mathcal{B}}_k s_k)|}{g(s_k, y_k)} &\leq \frac{\alpha_k \|y_k\| \|\operatorname{grad} f(x_k)\|}{g(s_k, y_k)} \text{ (by Step 3 of Algorithm 1)} \\ &\leq \frac{a_9^{1/2} \alpha_k \|\operatorname{grad} f(x_k)\|}{g^{1/2}(s_k, y_k)} \text{ (by (3.11))} \\ &\leq \frac{a_9^{1/2} \alpha_k \|\operatorname{grad} f(x_k)\|}{a_0^{1/2} \|s_k\|} \text{ (by (3.4))} \\ &\leq a_{12}\frac{\alpha_k}{\cos\theta_k}, \text{ (by (3.5))}\end{aligned}$$

where a_{12} is a positive constant. □

Lemma 3.11 generalizes [11, Lemma 3.2].

LEMMA 3.11. *Suppose Assumptions 3.1, 3.2 and 3.3 hold.* $\phi_k \in [0, 1]$. *Then there exists a constant* $a_{13} > 0$ *such that*

$$\prod_{j=1}^{k} \alpha_j \geq a_{13}^k, \tag{3.15}$$

for all $k \geq 1$.

Proof. The major difference between the Euclidean and Riemannian proofs is that in the Riemannian case, we have two operators $\mathcal{B}_k$ and $\tilde{\mathcal{B}}_k$ as opposed to a single operator in the Euclidean case. Once we have proven that they have the same trace and determinant, the proof unfolds similarly to the Euclidean proof. The details are given for the reader's convenience.

Use hat to denote the coordinates expression of the operators $\mathcal{B}_k$ and $\tilde{\mathcal{B}}_k$ in Algorithm 1 and consider $\mathrm{trace}(\hat{\mathcal{B}})$ and $\det(\hat{\mathcal{B}})$. Note that $\mathrm{trace}(\hat{\mathcal{B}})$ and $\det(\hat{\mathcal{B}})$ are independent of a basis. Since $\mathcal{T}_{\mathrm{S}}$ is an isometric vector transport, we have that $\mathcal{T}_{\mathrm{S}_{\alpha_k\eta_k}}$ is invertible for all k, and thus

$$\begin{aligned}\mathrm{trace}(\hat{\tilde{\mathcal{B}}}_k) &= \mathrm{trace}(\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}}\hat{\mathcal{B}}_k\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}}^{-1}) = \mathrm{trace}(\hat{\mathcal{B}}_k),\\ \det(\hat{\tilde{\mathcal{B}}}_k) &= \det(\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}}\hat{\mathcal{B}}_k\hat{\mathcal{T}}_{S_{\alpha_k\eta_k}}^{-1}) = \det(\hat{\mathcal{B}}_k).\end{aligned}$$

From the update formula of $\mathcal{B}_k$ in Algorithm 1, the trace of the update formula is

$$\begin{aligned}\mathrm{trace}(\hat{\mathcal{B}}_{k+1}) = \mathrm{trace}(\hat{\mathcal{B}}_k) &+ \frac{\|y_k\|^2}{g(y_k,s_k)} + \phi_k\frac{\|y_k\|^2}{g(y_k,s_k)}\frac{g(s_k,\tilde{\mathcal{B}}_ks_k)}{g(y_k,s_k)}\\ &- (1-\phi_k)\frac{\|\tilde{\mathcal{B}}_ks_k\|^2}{g(s_k,\tilde{\mathcal{B}}_ks_k)} - 2\phi_k\frac{g(y_k,\tilde{\mathcal{B}}_ks_k)}{g(y_k,s_k)}.\end{aligned}\tag{3.16}$$

Recall that $\phi_k g(s_k,\tilde{\mathcal{B}}_ks_k)\geq 0$. If we choose a particular basis such that the expression of the metric is the identity, then the Broyden update equation (2.3) is exactly the classical Broyden update equation, except that B_k is replaced by $\hat{\tilde{\mathcal{B}}}_k$, and by [11, (3.9)] we have

$$\det(\hat{\mathcal{B}}_{k+1}) \geq \det(\hat{\mathcal{B}}_k)\frac{g(y_k,s_k)}{g(s_k,\tilde{\mathcal{B}}_ks_k)}.\tag{3.17}$$

Since det and $g(\cdot,\cdot)$ are independent of the basis, it follows that (3.17) holds regardless of the chosen basis. Using (3.11), (3.12), (3.13) and (3.14) for (3.16), we obtain

$$\mathrm{trace}(\hat{\mathcal{B}}_{k+1}) \leq \mathrm{trace}(\hat{\mathcal{B}}_k) + a_9 + \phi_k a_9 a_{10}\alpha_k - \frac{a_{11}(1-\phi_k)\alpha_k}{\cos^2\theta_k} + \frac{2\phi_k a_{12}\alpha_k}{\cos\theta_k}\tag{3.18}$$

Notice that

$$\begin{aligned}\frac{\alpha_k}{\cos\theta_k} &= \frac{\alpha_k\|\,\mathrm{grad}\, f(x_k)\|\|\eta_k\|}{-g(\mathrm{grad}\, f(x_k),\eta_k)} = \frac{\alpha_k\|\tilde{\mathcal{B}}_ks_k\|\|s_k\|}{g(s_k,\tilde{\mathcal{B}}_ks_k)}\\ &= \frac{\|\tilde{\mathcal{B}}_ks_k\|}{\|s_k\|}\frac{\alpha_k\|s_k\|^2}{g(s_k,\tilde{\mathcal{B}}_ks_k)} \leq a_7\frac{\|\tilde{\mathcal{B}}_ks_k\|}{\|s_k\|}\ \text{(by (3.10))}.\end{aligned}\tag{3.19}$$

Since the fourth term in (3.18) is always negative, $\cos\theta_k\leq 1$, $\phi_k\geq 0$, (3.19) and (3.18) imply that

$$\mathrm{trace}(\hat{\mathcal{B}}_{k+1}) \leq \mathrm{trace}(\hat{\mathcal{B}}_k) + a_9 + (\phi_k a_9 a_{10} + 2\phi_k a_{12} a_7)\frac{\|\tilde{\mathcal{B}}_ks_k\|}{\|s_k\|}.$$

Consider $\|\hat{\mathcal{B}}_k\|_g$, where $\|\cdot\|_g$ denotes the induced norm from the vector norm $\|\hat{u}\|_g = \sqrt{\hat{u}^T G_k\hat{u}}$, where G_k is the matrix expression of the inner product of $\mathrm{T}_{x_k}\mathcal{M}$.

It follows that

$$\begin{aligned}
&\frac{\|\tilde{\mathcal{B}}_k s_k\|}{\|s_k\|} \leq \|\tilde{\mathcal{B}}_k\| = \|\mathcal{B}_k\| \\
&= \|\hat{\mathcal{B}}_k\|_g = \frac{\|\hat{\mathcal{B}}_k v\|_g}{\|v\|_g} \text{ (there exists a } v \text{ such that this equality holds)} \\
&= \sqrt{\frac{v^T \hat{\mathcal{B}}_k^T G_k \hat{\mathcal{B}}_k v}{v^T G_k v}} = \sqrt{\frac{v^T G_k \hat{\mathcal{B}}_k \hat{\mathcal{B}}_k v}{v^T G_k v}} \text{ (since } \mathcal{B}_k \text{ is self-adjoint, then } \hat{\mathcal{B}}_k^T G_k = G_k \hat{\mathcal{B}}_k) \\
&= \sqrt{\frac{\tilde{v}^T G_k^{1/2} \hat{\mathcal{B}}_k G_k^{-1/2} G_k^{1/2} \hat{\mathcal{B}}_k G_k^{-1/2} \tilde{v}}{\tilde{v}^T \tilde{v}}} = \sqrt{\frac{\tilde{v}^T G_k^{-1/2} G_k \hat{\mathcal{B}}_k G_k^{-1/2} G_k^{1/2} \hat{\mathcal{B}}_k G_k^{-1/2} \tilde{v}}{\tilde{v}^T \tilde{v}}} \\
&= \sqrt{\frac{\tilde{v}^T G_k^{-1/2} \hat{\mathcal{B}}_k^T G_k^{1/2} G_k^{1/2} \hat{\mathcal{B}}_k G_k^{-1/2} \tilde{v}}{\tilde{v}^T \tilde{v}}} = \sqrt{\frac{\tilde{v}^T M^T M \tilde{v}}{\tilde{v}^T \tilde{v}}} \\
&\leq \|M\|_2 \leq \operatorname{trace}(M) = \operatorname{trace}(\hat{\mathcal{B}}_k),
\end{aligned}$$

where $\tilde{v} = G_k^{1/2} v$, $M = G_k^{1/2} \hat{\mathcal{B}}_k G_k^{-1/2}$. Therefore, $\operatorname{trace}(\hat{\mathcal{B}}_{k+1}) \leq a_9 + (1 + \phi_k a_9 a_{10} + 2\phi_k a_{12} a_7) \operatorname{trace}(\hat{\mathcal{B}}_k)$. This inequality implies that there exists a constant $b_0 > 0$ such that

$$\operatorname{trace}(\hat{\mathcal{B}}_{k+1}) \leq b_0^k. \tag{3.20}$$

Using (3.12) and (3.17), we have

$$\det(\hat{\mathcal{B}}_{k+1}) \geq \det(\hat{\mathcal{B}}_k) \frac{1}{a_{10}\alpha_k} \geq \det(\hat{\mathcal{B}}_1) \prod_{j=1}^{k} \frac{1}{a_{10}\alpha_j}. \tag{3.21}$$

From the geometric/arithmetic mean inequality[3] applied to the eigenvalues of $\hat{\mathcal{B}}_{k+1}$, we know $\det(\hat{\mathcal{B}}_{k+1}) \leq (\frac{\operatorname{trace}(\hat{\mathcal{B}}_{k+1})}{d})^d$, where d is the dimension of manifold $\mathcal{M}$. Therefore, by (3.20) and (3.21),

$$\prod_{j=1}^{k} \frac{1}{a_{10}\alpha_j} \leq \frac{1}{\det(\hat{\mathcal{B}}_1)} (\frac{\operatorname{trace}(\hat{\mathcal{B}}_{k+1})}{d})^d \leq \frac{1}{\det(\hat{\mathcal{B}}_1) d^d} (b_0^d)^k.$$

Thus there exists a constant $a_{13} > 0$ such that $\prod_{j=1}^{k} \alpha_j \geq a_{13}^k$, for all $k \geq 1$. □

3.3. Main convergence result.

With the preliminary lemmas in place, the main convergence result can be stated and proven in a manner that closely follows the Euclidean proof of [11, Theorem 3.1].

Theorem 3.1. *Suppose Assumptions 3.1, 3.2 and 3.3 hold and $\phi_k \in [0, 1-\delta]$. Then the sequence $\{x_k\}$ generated by Algorithm 1 converges to a minimizer x^* of f.*

Proof. Inequality (3.18) can be written as

$$\operatorname{trace}(\hat{\mathcal{B}}_{k+1}) \leq \operatorname{trace}(\hat{\mathcal{B}}_k) + a_9 + t_k \alpha_k, \tag{3.22}$$

where $t_k = \phi_k a_9 a_{10} - a_{11}(1-\phi_k)/\cos^2\theta_k + 2\phi_k a_{12}/\cos\theta_k$. The proof is by contradiction. Assume $\cos\theta_k \to 0$, then $t_k \to -\infty$ since ϕ_k is bounded away from 1, i.e.,

[3] For $x_i \geq 0$, $(\prod_{i=1}^{d} x_i)^{1/d} \leq \sum_{i=1}^{d} x_i/d$.

$\phi_k \leq 1-\delta$. So there exists a constant $K_0 > 0$ such that $t_k < -2a_9/a_{13}$ for all $k \geq K_0$. Using (3.22) and that $\hat{\mathcal{B}}_{k+1}$ is positive definite, we have

$$\begin{aligned} 0 < \operatorname{trace}(\hat{\mathcal{B}}_{k+1}) &\leq \operatorname{trace}(\hat{\mathcal{B}}_{K_0}) + a_9(k+1-K_0) + \sum_{j=K_0}^{k} t_j \alpha_j \\ &< \operatorname{trace}(\hat{\mathcal{B}}_{K_0}) + a_9(k+1-K_0) - \frac{2a_9}{a_{13}} \sum_{j=K_0}^{k} \alpha_j. \end{aligned} \tag{3.23}$$

Applying the geometric/arithmetic mean inequality to (3.15), we get $\sum_{j=1}^{k} \alpha_j \geq k a_{13}$ and therefore

$$\sum_{j=K_0}^{k} \alpha_j \geq k a_{13} - \sum_{j=1}^{K_0-1} \alpha_j. \tag{3.24}$$

Plugging (3.24) into (3.23), we obtain

$$\begin{aligned} 0 &< \operatorname{trace}(\hat{\mathcal{B}}_{K_0}) + a_9(k+1-K_0) - \frac{2a_9}{a_{13}} k a_{13} + \frac{2a_9}{a_{13}} \sum_{j=1}^{K_0-1} \alpha_j \\ &= \operatorname{trace}(\hat{\mathcal{B}}_{K_0}) + a_9(1-k-K_0) + \frac{2a_9}{a_{13}} \sum_{j=1}^{K_0-1} \alpha_j. \end{aligned}$$

For large enough k, the right-hand side of the inequality is negative, which contradicts the assumption that $\cos\theta_k \to 0$. Therefore there exists a constant δ and a subsequence such that $\cos\theta_{k_j} > \delta > 0$ for all j, i.e., there is a subsequence that does not converge to 0. Applying Lemma 3.7 completes the proof. □

4. Ensuring the locking condition. In order to apply an algorithm in the RBroyden family, we must specify a retraction R and an isometric vector transport $\mathcal{T}_S$ that satisfy the locking condition (2.8). Exponential mapping and parallel transport satisfy condition (2.8) with $\beta = 1$. However, for some manifolds, we do not have the analytical form of exponential mapping and parallel transport. Even if a form is known its evaluation may be unacceptably expensive. Two methods of constructing an isometric vector given a retraction and a method for constructing a retraction and an isometric vector transport simultaneously are discussed in this section. In practice, the choice of the pair must also consider if an efficient implementation is possible.

In Sections 4.1 and 4.2, the differentiated retraction $\mathcal{T}_{R_\eta}\xi$ is only needed for η and ξ in a same direction, where $\eta, \xi \in \mathrm{T}_x \mathcal{M}$ and $x \in \mathcal{M}$. A reduction of complexity can follow from this restricted usage of $\mathcal{T}_R$. We illustrate this point for the Stiefel manifold $\mathrm{St}(p,n) = \{X \in \mathbb{R}^{n\times p} | X^T X = I_p\}$ as it is used in our experiments. Consider the retraction by polar decomposition [2, (4.8)],

$$R_x(\eta) = (x+\eta)(I+\eta^T\eta)^{-1/2}. \tag{4.1}$$

For retraction (4.1), as shown in [17, (10.2.7)], the differentiated retraction is $\mathcal{T}_{R_\eta}\xi = y\Lambda + (I_n - yy^T)\xi(y^T(x+\eta))^{-1}$, where $y = R_x(\eta)$, Λ is the unique solution of the Sylvester equation

$$y^T\xi - \xi^T y = \Lambda P + P\Lambda \tag{4.2}$$

and $P = y^T(x+\eta) = (I_p+\eta^T\eta)^{1/2}$. Now consider the more specific task of computing $\mathcal{T}_{R_\eta}\xi$ for η and ξ in a same direction. Since $\mathcal{T}_{R_\eta}\xi$ is linear in ξ, we can assume without loss of generality that $\xi = \eta$. Then the solution of (4.2) has a closed form, i.e., $\Lambda = P^{-1}x^T\eta P^{-1}$. This can be seen from

$$\begin{aligned}
&y^T\eta - \eta^T y = y^T(\eta + x - x) - (\eta + x - x)^T y = x^T y - y^T x + y^T(\eta + x) - (\eta + x)^T y\\
=&x^T y - y^T x = x^T(x+\eta)(I_p+\eta^T\eta)^{-1/2} - (I_p+\eta^T\eta)^{-1/2}(x+\eta)^T x\\
=&x^T\eta P^{-1} - P^{-1}\eta^T x = x^T\eta P^{-1} + P^{-1}x^T\eta \;(x^T\eta \text{ is a skew symmetric matrix})\\
=&P(P^{-1}x^T\eta P^{-1}) + (P^{-1}x^T\eta P^{-1})P.
\end{aligned}$$

The closed form solution of Λ yields a form, $\mathcal{T}_{R_\eta}\eta = (I_n - yP^{-1}\eta^T)\eta P^{-1}$, with lower complexity. We are not aware of such a low-complexity closed-form solution for $\mathcal{T}_{R_\eta}\xi$ when η and ξ are not in a same direction.

4.1. Method 1: from a retraction and an isometric vector transport. Given a retraction R, if an associated isometric vector transport,$\mathcal{T}_{\mathrm{I}}$, for which there is an efficient implementation, is known then $\mathcal{T}_{\mathrm{I}}$ can be modified so that it satisfies condition (2.8). Consider $x \in \mathcal{M}$, $\eta \in \mathrm{T}_x\mathcal{M}$, $y = R_x(\eta)$ and define the tangent vectors $\xi_1 = \mathcal{T}_{\mathrm{I}_\eta}\eta$ and $\xi_2 = \beta\mathcal{T}_{R_\eta}\eta$ with the normalizing scalar $\beta = \frac{\|\eta\|}{\|\mathcal{T}_{R_\eta}\eta\|}$. The desired isometric vector transport is

$$\mathcal{T}_{\mathrm{S}_\eta}\xi = (\mathrm{id} - \frac{2\nu_2\nu_2^\flat}{\nu_2^\flat\nu_2})(\mathrm{id} - \frac{2\nu_1\nu_1^\flat}{\nu_1^\flat\nu_1})\mathcal{T}_{I_\eta}\xi,$$

where $\nu_1 = \xi_1 - \omega$ and $\nu_2 = \omega - \xi_2$. ω could be any tangent vector in $\mathrm{T}_y\mathcal{M}$ which satisfies $\|\omega\| = \|\xi_1\| = \|\xi_2\|$. If ω is any unit-norm vector in the space spanned by $\{\xi_1, \xi_2\}$, e.g., $\omega = -\xi_1$ or $-\xi_2$, then P_y is the well-known direct rotation from ξ_1 to ξ_2 in the inner product that defines $\cdot^\flat$. The use of the negative sign avoids numerical cancelation as ξ_1 approaches ξ_2, i.e., near convergence. Two Householder reflectors are used to preserve the orientation, which is sufficient to make $\mathcal{T}_{\mathrm{S}}$ satisfy the consistency condition (ii) of vector transport. It can be shown that this $\mathcal{T}_{\mathrm{S}}$ satisfies conditions (2.5) and (2.6) [17, Theorem 4.4.1].

4.2. Method 2: from a retraction and bases of tangent spaces. Method 1 modifies a given isometric vector transport. In this section, a method to construct directly an isometric vector transport that satisfies the locking condition (2.8) is presented. Method 2 requires a function that constructs an orthonormal basis of $\mathrm{T}_x\mathcal{M}$. Let d denote the dimension of manifold $\mathcal{M}$ and let the function giving a basis of $\mathrm{T}_x\mathcal{M}$ be $B : x \to B(x) = (b_1, b_2, \dots b_d)$, where $b_i, 1 \leq i \leq d$ form an orthonormal basis of $\mathrm{T}_x\mathcal{M}$. For the results above that relax the smoothness requirement on vector transport, B need only be C^1.

Consider $x \in \mathcal{M}$, $\eta, \xi \in \mathrm{T}_x\mathcal{M}$, $y = R_x(\eta)$, $B_1 = B(x)$ and $B_2 = B(y)$. Define $B_1^\flat : \mathrm{T}_x\mathcal{M} \to \mathbb{R}^d : \eta_x \mapsto \begin{bmatrix} g_x(b_1^{(1)}, \eta_x) & \dots & g_x(b_d^{(1)}, \eta_x)\end{bmatrix}^T$, where $b_i^{(1)}$ is the i-th element of B_1 and likewise for $B_2^\flat$. It follows that the desired isometric vector transport is

$$\mathcal{T}_{\mathrm{S}_\eta}\xi = B_2(I - \frac{2v_2v_2^T}{v_2^Tv_2})(I - \frac{2v_1v_1^T}{v_1^Tv_1})B_1^\flat\xi,$$

where $v_1 = B_1^\flat\eta - w$, $v_2 = w - \beta B_2^\flat\mathcal{T}_{R_\eta}\eta$. w can be any vector such that $\|w\| = \|B_1^\flat\eta\| = \|\beta B_2^\flat\mathcal{T}_{R_\eta}\eta\|$, and choosing $w = -B_1^\flat\eta$ or $-\beta B_2^\flat\mathcal{T}_{R_\eta}\eta$ yields a direct rotation.

This vector transport also satisfies conditions (2.5) and (2.6); this can be shown using the same idea as in [17, Theorem 4.4.1].

The problem therefore becomes how to build the function B. Absil et al. [2, p. 37] give an approach based on $(\mathcal{U}, \varphi)$, a chart of the manifold $\mathcal{M}$, that yields a smooth B defined in the chart domain. E_i, the i-th coordinate vector field of $(\mathcal{U}, \varphi)$ on $\mathcal{U}$, is defined by $(E_i f)(x) := \partial_i(f \circ \varphi^{-1})(\varphi(x)) = D(f \circ \varphi^{-1})(\varphi(x))[e_i]$. These coordinate vector fields are smooth and every vector field ξ admits the decomposition $\xi = \sum_i (\xi \varphi_i) E_i$ on $\mathcal{U}$. The function $\tilde{B} : x \to \tilde{B}(x) = \{E_2, E_2, \dots, E_d\}$ is a smooth function that builds a basis on $\mathcal{U}$. Finally, any orthogonalization method, such as the Gram-Schmidt algorithm or a QR decomposition, can be used to get an orthonormal basis giving the function $B : x \to B(x) = \tilde{B}(x) Q(x)$, where $Q(x)$ is an upper triangle matrix with positive diagonal terms.

4.3. Method 3: from a transporter.

Let $L(\mathrm{T}\,\mathcal{M}, \mathrm{T}\,\mathcal{M})$ denote the fiber bundle with base space $\mathcal{M} \times \mathcal{M}$ such that the fiber over $(x, y) \in \mathcal{M} \times \mathcal{M}$ is $L(\mathrm{T}_x\,\mathcal{M}, \mathrm{T}_y\,\mathcal{M})$, the set of all linear maps from $\mathrm{T}_x\,\mathcal{M}$ to $\mathrm{T}_y\,\mathcal{M}$. We define a *transporter* $\mathcal{L}$ on $\mathcal{M}$ to be a smooth section of the bundle $L(\mathrm{T}\,\mathcal{M}, \mathrm{T}\,\mathcal{M})$—that is, for $(x, y) \in \mathcal{M} \times \mathcal{M}$, $\mathcal{L}(x, y) \in L(\mathrm{T}_x\,\mathcal{M}, \mathrm{T}_y\,\mathcal{M})$—such that, for all $x \in \mathcal{M}$, $\mathcal{L}(x, x) = \mathrm{id}$. Given a retraction R, it can be shown that $\mathcal{T}$ defined by

$$\mathcal{T}_{\eta_x} \xi_x = \mathcal{L}(x, R_x(\eta_x))\, \xi_x \tag{4.3}$$

is a vector transport with associated retraction R. Moreover, if $\mathcal{L}(x, y)$ is isometric from $\mathrm{T}_x\,\mathcal{M}$ to $\mathrm{T}_y\,\mathcal{M}$, then the vector transport (4.3) is isometric. (The term transporter was used previously in [24] in the context of embedded submanifolds.)

In this section, it is assumed that an efficient transporter $\mathcal{L}$ is given, and we show that the retraction R defined by the differential equation

$$\frac{d}{dt} R_x(t\eta) = \mathcal{L}(x, R_x(t\eta))\eta, \quad R_x(0) = x \tag{4.4}$$

and the resulting vector transport $\mathcal{T}$ defined by (4.3) satisfy the locking condition (2.8). To this end, let η be an arbitrary vector in $\mathrm{T}_x\,\mathcal{M}$. We have $\mathcal{T}_\eta \eta = \mathcal{L}(x, R_x(\eta))\eta = \frac{d}{dt} R_x(t\eta)|_{t=1} = \frac{d}{d\tau} R_x(\eta + \tau\eta))|_{\tau=0} = \mathcal{T}_{R_\eta} \eta$, where we have used (4.3), (4.4) and (2.7). That is, R and $\mathcal{T}$ satisfy the locking condition (2.8) with $\beta = 1$. For some manifolds, there exist transporters such that (4.4) has a closed solution, e.g., the Stiefel manifold [17, §10.2.3] and the Grassmann manifold [17, §10.6.3].

5. Limited-memory RBFGS.

In the form of RBroyden methods discussed above, explicit representations are needed for the operators $\mathcal{B}_k$, $\tilde{\mathcal{B}}_k$, $\mathcal{T}_{\mathrm{S}_{\alpha_k \eta_k}}$, and $\mathcal{T}^{-1}_{\mathrm{S}_{\alpha_k \eta_k}}$. These may not be available. Furthermore, even if explicit expressions are known, applying them may be unacceptably expensive computationally, e.g., the matrix multiplication required in the update of $\mathcal{B}_k$. Generalizations of the Euclidean limited-memory BFGS method can solve this problem for RBFGS. The idea of limited-memory RBFGS (LRBFGS) is to store some number of the most recent s_k and y_k and to transport those vectors to the new tangent space rather than the entire matrix $\mathcal{H}_k$.

For RBFGS, the inverse update formula is $\mathcal{H}_{k+1} = \mathcal{V}_k^\flat \tilde{\mathcal{H}}_k \mathcal{V}_k + \rho_k s_k s_k^\flat$, where $\tilde{\mathcal{H}}_k = \mathcal{T}_{\mathrm{S}_{\alpha_k \eta_k}} \circ \mathcal{H}_k \circ \mathcal{T}^{-1}_{\mathrm{S}_{\alpha_k \eta_k}}$, $\rho_k = \frac{1}{g(y_k, s_k)}$ and $\mathcal{V}_k = \mathrm{id} - \rho_k y_k s_k^\flat$. If the $\ell + 1$ most recent

s_k and y_k are stored then we have

$$\begin{aligned}\mathcal{H}_{k+1} &= \tilde{\mathcal{V}}_k^\flat \tilde{\mathcal{V}}_{k-1}^\flat \cdots \tilde{\mathcal{V}}_{k-\ell}^\flat \tilde{\mathcal{H}}_{k+1}^0 \tilde{\mathcal{V}}_{k-\ell} \cdots \tilde{\mathcal{V}}_{k-1} \tilde{\mathcal{V}}_k \\ &\quad + \rho_{k-\ell} \tilde{\mathcal{V}}_k^\flat \tilde{\mathcal{V}}_{k-1}^\flat \cdots \tilde{\mathcal{V}}_{k-\ell+1}^\flat s_{k-\ell}^{(k+1)} {s_{k-\ell}^{(k+1)}}^\flat \tilde{\mathcal{V}}_{k-\ell+1} \cdots \tilde{\mathcal{V}}_{k-1} \tilde{\mathcal{V}}_k \\ &\quad + \cdots + \rho_k s_k^{(k+1)} s_k^{(k+1)},\end{aligned}$$

where $\tilde{\mathcal{V}}_i = \operatorname{id} - \rho_i y_i^{(k+1)} {s_i^{(k+1)}}^\flat$, $\tilde{\mathcal{H}}_{k+1}^0$ is the initial Hessian approximation for step $k+1$, $s_i^{(k+1)}$ represents a tangent vector in $\mathrm{T}_{x_{k+1}} \mathcal{M}$ given by transporting s_i and likewise for $y_i^{(k+1)}$. The details of transporting s_i, y_i to $s_i^{(k+1)}, y_i^{(k+1)}$ are given later. Note that $\tilde{\mathcal{H}}_{k+1}^0$ is not necessarily $\tilde{\mathcal{H}}_{k-\ell}$. It can be any positive definite self-adjoint operator. Similar to the Euclidean case, we use

$$\mathcal{H}_{k+1}^0 = \frac{g(s_k, y_k)}{g(y_k, y_k)} \operatorname{id}. \tag{5.1}$$

It is easily seen that $\eta_{k+2} = -\mathcal{H}_{k+1} \operatorname{grad} f(x_{k+1})$ which yields the Step 4 to Step 13 of Algorithm 2 (generalized from the two-loop recursion, see e.g., [22, Algorithm 7.4]), and the explicit form of $\mathcal{T}_{\mathrm{S}}$ is not required.

Algorithm 2 is a limited-memory algorithm based on this idea.

Note Step 18 of Algorithm 2. The vector $s_{k-m}^{(k+1)}$ is obtained by transporting $s_{k-m}^{(k-m+1)}$ m times. If the vector transport is insensitive to finite precision then the approach is acceptable. Otherwise, $s_{k-m}^{(k+1)}$ may be not in $\mathrm{T}_{x_{k+1}} \mathcal{M}$. Care must be taken to avoid this situation. One possibility is to project $s_i^{(k+1)}, y_i^{(k+1)}, i = l, l+1, \ldots, k-1$ to tangent space $T_{x_{k+1}}\mathcal{M}$ after every transport. The other possibility is to store $x_{l+1}, x_{l+2}, \ldots, x_k$, $s_l^{(l+1)}, s_{l+1}^{(l+2)}, \ldots, s_k^{(k+1)}$ and $y_l^{(l+1)}, y_{l+1}^{(l+2)}, \ldots, y_k^{(k+1)}$ and transport $s_i^{(i+1)}, y_i^{(i+1)}$ from $T_{x_{i+1}}\mathcal{M}$ to $T_{x_{k+1}}\mathcal{M}$ during Steps 4 to 13. The former approach is used in our experiments.

6. Ring and Wirth's RBFGS update formula. In Ring and Wirth's RBFGS [26] for infinite dimensional Riemannian manifolds, the direction vector η_k is chosen to define a function on $\mathrm{T}_{x_k} \mathcal{M}$ that satisfies the equation $\mathcal{B}_k(\eta_k, \xi) = \mathrm{D} f_{R_{x_k}}(0)[\xi] = g(\operatorname{grad} f(x_k), \xi)$ for all $\xi \in \mathrm{T}_{x_k} \mathcal{M}$.

The update in [26] is

$$\mathcal{B}_{k+1}(\mathcal{T}_{\mathrm{S}_{\alpha_k \eta_k}} \zeta, \mathcal{T}_{\mathrm{S}_{\alpha_k \eta_k}} \xi) = \mathcal{B}_k(\zeta, \xi) - \frac{\mathcal{B}_k(s_k, \zeta) \mathcal{B}_k(s_k, \xi)}{\mathcal{B}_k(s_k, s_k)} + \frac{y_k(\zeta) y_k(\xi)}{y_k(s_k)},$$

where $s_k = R_{x_k}^{-1}(x_{k+1}) \in \mathrm{T}_{x_k} \mathcal{M}$ and $y_k = \mathrm{D} f_{R_{x_k}}(s_k) - \mathrm{D} f_{R_{x_k}}(0)$ is a cotangent vector at x_k, i.e., $\mathrm{D} f_{R_{x_k}}(s_k)[\xi] = g(\operatorname{grad} f(R_{x_k}(s_k)), \mathcal{T}_{R_{s_k}} \xi)$ for all $\xi \in \mathrm{T}_{x_k} \mathcal{M}$. If the dimension of the manifold is finite then using coordinates yields the following form of the update to the matrix that defines the bilinear function

$$\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^T \hat{\mathcal{B}}_{k+1} \hat{\mathcal{T}}_{S_{\alpha_k \eta_k}} = \hat{\mathcal{B}}_k - \frac{\hat{\mathcal{B}}_k \hat{s}_k \hat{s}_k^T \hat{\mathcal{B}}_k}{\hat{s}_k^T \hat{\mathcal{B}}_k \hat{s}_k} + \frac{\hat{y}_k \hat{y}_k^T}{\hat{y}_k^T \hat{s}_k}, \tag{6.1}$$

where $\hat{y}_k$ satisfies $y_k(\eta) = \hat{y}_k^T \hat{\eta}$ for all $\eta \in \mathrm{T}_{x_k} \mathcal{M}$ and $\mathcal{B}_k(\zeta, \xi) = \hat{\zeta}^T \hat{\mathcal{B}}_k \hat{\xi}$ for all $\zeta, \xi \in \mathrm{T}_{x_k} \mathcal{M}$. In contrast, the bilinear function defined by Algorithm 1 is $g(\zeta, \mathcal{B}_k \xi) = \hat{\zeta}^T G_k \hat{\mathcal{B}}_k \hat{\xi}$ where G_k is the matrix expression of the metric in $\mathrm{T}_{x_k} \mathcal{M}$. Note that $\hat{\mathcal{B}}_k$

Algorithm 2 LRBFGS

```
Input: Riemannian manifold $\mathcal{M}$ with Riemannian metric $g$; a retraction $R$; isometric
    vector transport $\mathcal{T}_{\mathrm{S}}$ that satisfies (2.8); smooth function $f$ on $\mathcal{M}$; initial iterate
    $x_0 \in \mathcal{M}$; an integer $\ell > 0$.
$k = 0$, $\varepsilon > 0$, $0 < c_1 < \frac{1}{2} < c_2 < 1$, $\gamma_0 = 1$, $l = 0$;
while $\|\operatorname{grad} f(x_k)\| > \varepsilon$ do
    $\mathcal{H}_k^0 = \gamma_k \operatorname{id}$. Obtain $\eta_k \in \mathrm{T}_{x_k} \mathcal{M}$ by the following algorithm:
        $q \leftarrow \operatorname{grad} f(x_k)$
        for $i = k-1, k-2, \dots, l$ do
            $\xi_i \leftarrow \rho_i g(s_i^{(k)}, q)$;
            $q \leftarrow q - \xi_i y_i^{(k)}$;
        end for
        $r \leftarrow \mathcal{H}_k^0 q$;
        for $i = l, l+1, \dots, k-1$ do
            $\omega \leftarrow \rho_i g(y_i^{(k)}, r)$;
            $r \leftarrow r + s_i^{(k)}(\xi_i - \omega)$;
        end for
    set $\eta_k = -r$;
    find $\alpha_k$ that satisfies Wolfe conditions

        $f(x_{k+1}) \leq f(x_k) + c_1 \alpha_k g(\operatorname{grad} f(x_k), \eta_k)$
        $\frac{d}{dt} f(R_x(t\eta_k))|_{t=\alpha_k} \geq c_2 \frac{d}{dt} f(R_x(t\eta_k)|_{t=0}$

    Set $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$;
    Define $s_k^{(k+1)} = \mathcal{T}_{\mathrm{S}_{\alpha_k \eta_k}} \alpha_k \eta_k$, $y_k^{(k+1)} = \operatorname{grad} f(x_{k+1})/\beta_k - \mathcal{T}_{\mathrm{S}_{\alpha_k \eta_k}} \operatorname{grad} f(x_k)$, $\rho_k =$
        $1/g(s_k^{(k+1)}, y_k^{(k+1)})$, $\gamma_{k+1} = g(s_k^{(k+1)}, y_k^{(k+1)})/\|y_k^{(k+1)}\|^2$, and $\beta_k = \frac{\|\alpha_k \eta_k\|}{\|\mathcal{T}_{R_{\alpha_k \eta_k}} \alpha_k \eta_k\|}$;
    Let $l = \max\{k - \ell, 0\}$. Add $s_k^{(k+1)}$, $y_k^{(k+1)}$ and $\rho_k$ into storage and if $k > \ell$,
        then discard vector pair $\{s_{l-1}^{(k)}, y_{l-1}^{(k)}\}$ and scalar $\rho_{l-1}$ from storage; Transport
        $s_l^{(k)}, s_{l+1}^{(k)}, \dots, s_{k-1}^{(k)}$ and $y_l^{(k)}, y_{l+1}^{(k)}, \dots, y_{k-1}^{(k)}$ from $\mathrm{T}_{x_k} \mathcal{M}$ to $\mathrm{T}_{x_{k+1}} \mathcal{M}$ by $\mathcal{T}_{\mathrm{S}}$, then
        get $s_l^{(k+1)}, s_{l+1}^{(k+1)}, \dots, s_{k-1}^{(k+1)}$ and $y_l^{(k+1)}, y_{l+1}^{(k+1)}, \dots, y_{k-1}^{(k+1)}$;
    $k = k + 1$;
end while
```

in the update of [26] plays the same role as $G_k \hat{\mathcal{B}}_k$ in Algorithm 1. The expressions in [26] absorb the matrix G_k, implicit in the definition of the inner product, into the definitions of $\hat{\mathcal{B}}_k$ and y_k. Therefore, the RBFGS update of Algorithm 1 and the RBFGS update of [26] are the same. The difference between the algorithms is the definitions of s_k and y_k.

An inverse Hessian approximation update formula is not derived in [26]. However, the following coordinate expression of the inverse Hessian approximation update can be derived from (6.1)

$$\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^{-1} \hat{\mathcal{H}}_{k+1} (\hat{\mathcal{T}}_{S_{\alpha_k \eta_k}}^T)^{-1} = (I - \frac{\hat{s}_k \hat{y}_k^T}{\hat{y}_k^T \hat{s}_k}) \hat{\mathcal{H}}_k (I - \frac{\hat{y}_k \hat{s}_k^T}{\hat{y}_k^T \hat{s}_k}) + \frac{\hat{s}_k \hat{s}_k^T}{\hat{y}_k^T \hat{s}_k}. \tag{6.2}$$

Finite dimensional versions of the RBFGS in [26] based on these two updates are easily implemented. Our experiments include the more efficient inverse Hessian approximation form.

7. Experiments.

To investigate the performance of the RBroyden family of methods, we consider the minimization of the Brockett cost function

$$f : \mathrm{St}(p, n) \to \mathbb{R} : X \mapsto \mathrm{trace}(X^T AXN),$$

where $N = \mathrm{diag}(\mu_1, \ldots, \mu_p)$ with $\mu_1 > \cdots > \mu_p > 0$, $A \in \mathbb{R}^{n\times n}$ and $A = A^T$. It has been shown that the columns of a global minimizer, $X^* e_i$, are eigenvectors for the p smallest eigenvalues, λ_i, ordered so that $\lambda_1 \leq \cdots \leq \lambda_p$ [2, §4.8].

The Brockett cost function is not a retraction-convex function on the entire domain. However, for generic choices of A, the cost function is strongly retraction-convex in a sublevel set around any global minimizer. RBroyden algorithms do not require retraction-convexity to be well-defined and all converge to a global minimizer of the Brockett cost function if started sufficiently close.

For our experiments we view $\mathrm{St}(p, n)$ as an embedded submanifold of the Euclidean space $\mathbb{R}^{n\times p}$ and Riemannian metric is endowed with the Euclidean metric $\langle A, B\rangle = \mathrm{trace}(A^T B)$. The corresponding gradient can be found in [2, p. 80].

7.1. Methods tested.

For problems with sizes moderate enough to make dense matrix operations acceptable in the complexity of a single step, we tested algorithms in the RBroyden family using the inverse Hessian approximation update

$$\mathcal{H}_{k+1} = \tilde{\mathcal{H}}_k - \frac{\tilde{\mathcal{H}}_k y_k (\tilde{\mathcal{H}}_k^* y_k)^\flat}{(\tilde{\mathcal{H}}_k^* y_k)^\flat y_k} + \frac{s_k s_k^\flat}{s_k^\flat y_k} + \tilde{\phi}_k g(y_k, \tilde{\mathcal{H}}_k y_k) u_k u_k^\flat, \tag{7.1}$$

where $u_k = s_k/g(s_k, y_k) - \tilde{\mathcal{H}}_k y_k/g(y_k, \tilde{\mathcal{H}}_k y_k)$. In our tests, we set variable $\tilde{\phi}_k$ to Davidon's value $\tilde{\phi}_k^D$ defined in (7.5) or we set $\tilde{\phi}_k = \tilde{\phi} = 1.0, 0.8, 0.6, 0.4, 0.2, 0.1, 0.01, 0$. For these problem sizes the inverse Hessian approximation update tends to be preferred to the Hessian approximation update since it avoids solving a linear system. Our experiments on problems of moderate size also include the RBFGS in [26] with the inverse Hessian update formula (6.2) we derived earlier.

Finally, we investigate the performance of the LRBFGS for problems of moderate size and problems with much larger sizes. In the latter cases, since dense matrix operations are too expensive computationally and spatially, LRBFGS is compared to a Riemannian conjugate gradient method, RCG, described in [2]. All experiments are performed in Matlab R2014a on a 64 bit Ubuntu platform with 3.6 GHz CPU(Intel(R) Core(TM) i7-4790).

7.2. Vector transport and retraction.

Sections 2.2 and 2.3 in [18] provide methods for representing a tangent vector and constructing isometric vector transports when the d-dimensional manifold $\mathcal{M}$ is a submanifold of $\mathbb{R}^w$. If the codimension $w - d$ is not much smaller than w, then a d-dimensional representation for a tangent vector and an isometric transporter by parallelization

$$\mathcal{L}(x, y) = B_y B_x^\flat$$

are preferred, where $B : \mathcal{V} \to \mathbb{R}^{w\times d} : z \mapsto B_z$ is a smooth function defined on an open set $\mathcal{V}$ of $\mathcal{M}$ and the columns of B_z form an orthonormal basis of $\mathrm{T}_z\,\mathcal{M}$. Otherwise, a w-dimensional representation for a tangent vector and isometric transporter by rigging

$$\mathcal{L}(x, y) = G_y^{-\frac{1}{2}}(I - Q_x Q_x^T - Q_y Q_x^T) G_x^{\frac{1}{2}}$$

are preferred, where G_z denotes a matrix expression of g_z, i.e., $g_z(\eta_z, \xi_z) = \eta_z^T G_z \eta_z$, Q_x is given by orthonormalizing $(I - N_x(N_x^T N_x)^{-1} N_x^T) N_y$, likewise with x and y interchanged for Q_y, and $N : \mathcal{V} \to \mathbb{R}^{(w-d)\times d} : z \mapsto N_z$ is a smooth function defined on an open set $\mathcal{V}$ of $\mathcal{M}$ and the elements of N_z form an orthonormal basis of the normal space at z. In particular for $\mathrm{St}(p, n)$, w and d are np and $np - p(p+1)/2$ respectively. Methods of obtaining smooth functions to build smooth bases of the tangent and normal spaces for $\mathrm{St}(p, n)$ are discussed in [18, Section 5].

We use the ideas in Section 4.3 applied to the isometric transporter derived from the parallelization B introduced in [18, Section 5] to define a retraction R. The details, worked out in [17, §10.2.3], yield the following result. Let $X \in \mathrm{St}(p, n)$. We have

$$(R_X(\eta) \;\; R_X(\eta)_\perp) = (X \;\; X_\perp) \left(\exp \begin{pmatrix} \Omega & -K^T \\ K & 0_{(n-p)\times(n-p)} \end{pmatrix} \right), \tag{7.2}$$

where $\Omega = X^T \eta$, $K = X_\perp^T \eta$ and given $M \in \mathbb{R}^{n\times p}$, $M_\perp \in \mathbb{R}^{n\times(n-p)}$ denotes a matrix that satisfies $M_\perp^T M_\perp = I_{(n-p)\times(n-p)}$ and $M^T M_\perp = I_{p\times(n-p)}$. The function

$$Y = R_X(\eta) \tag{7.3}$$

is the desired retraction. The matrix $Y_\perp$ required in the definition of the basis B_Y is set to be $R_X(\eta)_\perp$ given by (7.2).

Since the RBFGS in [26] does not require the locking condition, retractions other than (7.3) may be used. For example, the qf retraction [2, (4.7)],

$$R_x(\eta) = \mathrm{qf}(x + \eta), \tag{7.4}$$

where qf denotes the Q factor of the QR decomposition with nonnegative elements on the diagonal of R, does not satisfy the locking condition and is less computationally expensive than (4.1) and (7.3). We have verified experimentally that using the qf retraction rather than (7.3) in the RBFGS of [26] produces smaller computational times. Therefore, the RBFGS of [26] experiments use the qf retraction and vector transport by parallelization.

The closed form of the differentiated qf retraction exists and the cotangent vector $\mathrm{D}\, f_{R_x}(s)$ required by Ring and Wirth is computationally inexpensive for the Stiefel manifold. The computational details can be found in [17, §10.2.4] and we give the result:

$$\mathrm{D}\, f_{R_x}(s) = ((y^T(x+s))^{-1}(2\,\mathrm{triu}(\mathrm{grad}\, f(R_x(s))^T y) y^T + \mathrm{grad}\, f(R_x(s))^T (I - yy^T))),$$

where $y = R_x(s)$ and $(\mathrm{triu}(M))_{i,j} = M_{i,j}$ if $i < j$ and $(\mathrm{triu}(M))_{i,j} = 0$ otherwise.

7.3. Notation, algorithm parameters and test data parameters. Given a search direction for a RBroyden algorithm, the step size α_k is set using the line search algorithm in [14, Algorithm A6.3.1mod] for optimizing a smooth function of a scalar. The constants c_1 and c_2 in Wolfe conditions are taken to be 10^{-4} and 0.999 respectively. The initial step size for the line search algorithm is given by the approach in [22, Page 60].

Unless otherwise indicated in the description of the experiments, the problems are defined by setting $A = Z + Z^T$ where the elements of Z are drawn from the standard normal distribution using Matlab's RANDN with seed 1, N is a diagonal matrix whose diagonal elements are integers from 1 to p, i.e., $N = \operatorname{diag}(p, p-1, \dots, 1)$. The initial iterate X_0 is given by applying Matlab's function ORTH to a matrix whose elements are drawn from the standard normal distribution. The identity is used as the initial Hessian inverse approximation, the intrinsic dimension representation is used for a tangent vector and vector transport is defined by parallelization. The stopping criterion requires that the ratio of the norm of initial gradient and the norm of final gradient is less than 10^{-6}.

To obtain sufficiently stable timing results, an average time is taken of several runs with identical parameters for a total runtime of at least 1 minute.

7.4. Performance for different ϕ. Most of the existing literature investigates the effects of the coefficient ϕ_k in the Hessian approximation update formula. In [11], Byrd et al. claim that in Euclidean space, the ability to correct eigenvalues of the Hessian approximation that are much larger than the eigenvalues of the true Hessian degrades for larger ϕ values. Our experiments show the same trend on manifolds (see Table 1) and RBFGS is seen to be the best at such a correction among the restricted RBroyden family methods.

Strategies for choosing ϕ_k and allowing it to be outside $[0, 1]$ have been investigated. Davidon [13] defines an update for ϕ_k by minimizing the condition number of $\mathcal{B}_k^{-1}\mathcal{B}_{k+1}$, subject to preserving positive definiteness. We have generalized this update to Riemannian manifolds for both the Hessian approximation (2.3) and inverse Hessian approximation (7.1) forms to obtain

$$\phi_k^D = \begin{cases} \frac{g(y_k,s_k)(g(y_k,\tilde{\mathcal{B}}_k^{-1}y_k)-g(y_k,s_k))}{g(s_k,\tilde{\mathcal{B}}_k s_k)g(y_k,\tilde{\mathcal{B}}_k^{-1}y_k)-g(y_k,s_k)^2}, & \text{if } g(y_k, s_k) \leq \frac{2g(s_k,\tilde{\mathcal{B}}_k s_k)g(y_k,\tilde{\mathcal{B}}_k^{-1}y_k)}{g(s_k,\tilde{\mathcal{B}}_k s_k)+g(y_k,\tilde{\mathcal{B}}_k^{-1}y_k)}; \\ \frac{g(y_k,s_k)}{g(y_k,s_k)-g(s_k,\tilde{\mathcal{B}}_k s_k)}, & \text{otherwise} \end{cases}$$

$$\tilde{\phi}_k^D = \begin{cases} \frac{g(y_k,s_k)(g(s_k,\tilde{\mathcal{H}}_k^{-1}s_k)-g(y_k,s_k))}{g(y_k,\tilde{\mathcal{H}}_k y_k)g(s_k,\tilde{\mathcal{H}}_k^{-1}s_k)-g(y_k,s_k)^2}, & \text{if } g(y_k, s_k) \leq \frac{2g(s_k,\tilde{\mathcal{H}}_k^{-1}s_k)g(y_k,\tilde{\mathcal{H}}_k y_k)}{g(s_k,\tilde{\mathcal{H}}_k^{-1}s_k)+g(y_k,\tilde{\mathcal{H}}_k y_k)}; \\ \frac{g(y_k,s_k)}{g(y_k,s_k)-g(y_k,\tilde{\mathcal{H}}_k y_k)}, & \text{otherwise.} \end{cases} \tag{7.5}$$

When the "if" conditions are satisfied in these definitions the Hessian and inverse Hessian approximations are symmetric positive definite. The "otherwise" clauses in the definitions correspond to the two forms of the Riemannian SR-1 method [18].

We point out that Byrd et al. [10] use negative values of ϕ to improve the performance of the Hessian approximation form. However, their experiments require solving a linear system to find $z_k = \operatorname{Hess} f(x_k)^{-1} v_k$. Their purpose was, of course, to demonstrate a theoretical value of ϕ_k and not to recommend the specific form for computation which by involving the Hessian is inconsistent with the goal of quasi-Newton methods. In the Riemannian setting, the action of the Hessian is often known rather than the Hessian itself, i.e., given $\eta \in \mathrm{T}_x \mathcal{M}$, $\operatorname{Hess} f(x)[\eta]$ is known. So z_k could be approximated by applying a few steps of an iterative method such as CG to the system of equations. Also, the Hessian could be recovered given a basis for $\mathrm{T}_x \mathcal{M}$ and the linear system solved but this is an excessive amount of work. Therefore, we test only the generalization of Davidon's update, $\tilde{\phi}_k^D$.

Since we work on the inverse Hessian approximation update, $\tilde{\phi}_k \equiv 1$ corresponds

$\tilde{\phi}_k$	$\tilde{\phi}_k^D$	1.0	0.8	0.6	0.4	0.2	0.1	0.01	0
$iter$	1.63_2	1.84_2	1.98_2	2.23_2	2.64_2	3.55_2	4.71_2	1.49_3	1.23_5
nf	1.64_2	1.85_2	1.99_2	2.24_2	2.65_2	3.57_2	4.72_2	1.49_3	1.23_5
ng	1.64_2	1.85_2	1.99_2	2.24_2	2.65_2	3.57_2	4.72_2	1.49_3	1.23_5
nH	3.24_2	3.66_2	3.94_2	4.43_2	5.26_2	7.09_2	9.39_2	2.98_3	2.45_5
nV	4.87_2	5.50_2	5.92_2	6.66_2	7.91_2	1.06_3	1.41_3	4.47_3	3.68_5
nR	1.63_2	1.84_2	1.98_2	2.23_2	2.64_2	3.56_2	4.71_2	1.49_3	1.23_5
gf_f	3.45_{-8}	3.29_{-8}	4.43_{-8}	4.39_{-8}	4.90_{-8}	5.07_{-8}	5.27_{-8}	5.31_{-8}	5.52_{-8}
$\frac{gf_f}{gf_0}$	6.26_{-7}	6.07_{-7}	7.87_{-7}	7.87_{-7}	8.80_{-7}	9.08_{-7}	9.40_{-7}	9.48_{-7}	9.85_{-7}
t	1.47_{-1}	1.51_{-1}	1.65_{-1}	1.82_{-1}	2.15_{-1}	2.84_{-1}	3.73_{-1}	1.19	9.73_1

TABLE 1

Comparison of RBroyden family for $\tilde{\phi}_k^D$ and several constant $\tilde{\phi}_k$. As mentioned in Section 7.4, $\tilde{\phi}_k \equiv 1$ corresponds to RBFGS and $\tilde{\phi}_k \equiv 0$ to RDFP. The average number of Riemannian SR1 updates in Davidon's update is 5.1. The subscript $-k$ indicates a scale of 10^{-k}. $iter, nf, ng, nV, nR$ denote the number of iterations, the number of function evaluations, the number of gradient evaluations, the number of actions of a vector transport and the number of actions of a retraction respectively. nH denote the number of operations of the form $\operatorname{Hess} f(x)\eta$ or $\mathcal{B}\eta$. t denotes the run time (seconds). gf_0 and gf_f denote the initial and final norm of the gradient.

to RBFGS and $\tilde{\phi}_k \equiv 0$ corresponds to RDFP. Also note we are testing the restricted RBroyden family since $0 \leq \tilde{\phi}_k \leq 1$ implies $0 \leq \phi_k \leq 1$. The parameters n and p are chosen to be 12 and 8 respectively. To show the differences among RBroyden family with different ϕ_k, the initial inverse Hessian approximation $\mathcal{H}_0$ is set to be $\operatorname{diag}(1,\ 1,\ \dots\ ,\ 1,\ 1/50,\ 1/10000) \in \mathbb{R}^{d \times d}$. This kind of choice for $\mathcal{H}_0$ has been used in [10]. The matrix A is set to be QDQ^T where Q is obtained by applying matlab ORTH command on a matrix whose entries are drawn from the standard normal distribution, $D = \operatorname{diag}(0,\ 0, \dots, 0, 0.01/(n-p), 0.02/(n-p), \dots, 0.01(n-p)/(n-p))$, the number of 0 is p. Note that minimizers are not isolated with this D. The experiments thus illustrate the fact, well know in the Euclidean case, that the method may converge when the strong convexity assumption (Assumption 3.3) is not satisfied.

Table 1 shows the average results of 10 runs with random Q and initial iterate. There is a clear preference in performance for choosing the constant $\tilde{\phi}$ near 1.0 to yield RBFGS or a nearby method. Davidon's update does not perform better than RBFGS in general. However, for the Brockett cost function with this particular eigenvalue distribution, i.e., the form of D, Davidon's update performs better than RBFGS for Brockett's cost function in the sense of number of iterations. How to efficiently and effectively choose $\tilde{\phi}_k$ or ϕ_k for general problems is still an open question in Riemannian optimization research.

7.5. Comparison of RBFGS, Ring and Wirth's RBFGS and LRBFGS. Ring and Wirth's RBFGS [26] is a potentially competitive alternative to the family of RBroyden methods. This is particularly true on the Stiefel manifold, since the qf retraction, transport using the differentiated qf retraction and the intrinsic dimension form of a cotangent vector all have relatively efficient computational forms described in Section 7.2.

Table 2 contains the results for Brockett's cost function with multiple sizes of the Stiefel manifold for the efficient Ring and Wirth algorithm (RW) and RBFGS using isometric vector transport by parallelization and retraction (7.3). RBFGS is a competitive method even though RW was run with the retraction (qf) that makes it most efficient. The smaller time advantage on the largest problem indicates that the dense matrix computations are beginning to mask the effects of other algorithmic

$(n,\ p)$	$(12,\ 6)$		$(12,\ 12)$		$(24,\ 12)$		$(24,\ 24)$	
method	RBFGS	RW	RBFGS	RW	RBFGS	RW	RBFGS	RW
iter	6.63_{1}	8.54_{1}	7.93_{1}	7.98_{1}	2.05_{2}	2.47_{2}	2.34_{2}	2.31_{2}
nf	7.44_{1}	9.63_{1}	8.72_{1}	9.17_{1}	2.11_{2}	2.53_{2}	2.37_{2}	2.35_{2}
ng	6.64_{1}	8.56_{1}	7.93_{1}	8.00_{1}	2.05_{2}	2.47_{2}	2.35_{2}	2.31_{2}
nV	1.96_{2}	1.70_{2}	2.35_{2}	1.58_{2}	6.13_{2}	4.92_{2}	7.00_{2}	4.60_{2}
nR	7.34_{1}	9.53_{1}	8.62_{1}	9.07_{1}	2.10_{2}	2.52_{2}	2.36_{2}	2.34_{2}
gf_f	4.34_{-5}	5.17_{-5}	5.11_{-5}	6.15_{-5}	2.23_{-4}	2.15_{-4}	3.04_{-4}	3.08_{-4}
gf_f/gf_0	6.39_{-7}	7.53_{-7}	5.57_{-7}	6.76_{-7}	8.40_{-7}	8.09_{-7}	9.18_{-7}	9.26_{-7}
t	5.30_{-2}	6.97_{-2}	6.64_{-2}	6.59_{-2}	2.39_{-1}	2.78_{-1}	3.21_{-1}	3.25_{-1}

Table 2

Comparison of RBFGS and RW for average of 10 random runs. The subscript $-k$ indicates a scale of 10^{-k}.

method	RBFGS	LRBFGS					
ℓ		1	2	4	8	16	32
iter	3.40_{2}	7.60_{2}	6.78_{2}	6.09_{2}	5.84_{2}	5.38_{2}	4.91_{2}
nf	3.43_{2}	8.01_{2}	6.91_{2}	6.14_{2}	5.87_{2}	5.42_{2}	4.94_{2}
ng	3.40_{2}	7.60_{2}	6.78_{2}	6.09_{2}	5.84_{2}	5.38_{2}	4.91_{2}
nV	1.02_{3}	2.28_{3}	3.39_{3}	5.47_{3}	9.86_{3}	1.75_{4}	3.09_{4}
nR	3.42_{2}	8.00_{2}	6.90_{2}	6.13_{2}	5.86_{2}	5.41_{2}	4.93_{2}
gf_f	5.72_{-4}	5.28_{-4}	5.25_{-4}	5.21_{-4}	5.35_{-4}	5.69_{-4}	5.44_{-4}
gf_f/gf_0	9.62_{-7}	8.90_{-7}	8.82_{-7}	8.74_{-7}	8.98_{-7}	9.57_{-7}	9.15_{-7}
t	1.02	6.53_{-1}	6.62_{-1}	7.41_{-1}	9.73_{-1}	1.36	2.08

Table 3

Comparison of LRBFGS and RBFGS for average of 10 random runs. The subscript $-k$ indicates a scale of 10^{-k}.

choices. This motivates a comparison with the LRBFGS method intended to limit the use of dense matrices with the full dimension of the problem.

The performance results for RBFGS and LRBFGS with different values of the parameter ℓ are given in Table 3 for Brockett's cost function with $n = p = 32$. As expected, the number of iterations required by LRBFGS to achieve a reduction in the norm of the gradient comparable to RBFGS decreases as ℓ increases but remains higher than the number required by RBFGS. The benefit of LRBFGS is seen in computation times that are superior or similar to that of RBFGS for $\ell \leq 8$. This clearly indicates that, for this range of ℓ, the approximation of the inverse of the Hessian is of suitable quality in LRBFGS so that the number of less complex iterations is kept sufficiently small to solve the problem in an efficient manner. The advantage is lost, as expected, once ℓ becomes too large for the size of the given problem. In practice, for moderately sized problems, exploiting the potential benefits of LRBFGS requires an efficient method of choosing ℓ which depends strongly on the problem. The results are encouraging in the sense of potential for problems large enough to preclude the use of RW, RBFGS, or other RBroyden family members.

7.6. A large-scale problem. In the final set of experiments illustrating the potential of the methods, LRBFGS is applied to Brockett's cost function for several sufficiently large values of n and p. The qf retraction is used and isometric vector transport is defined by applying ideas of Section 4.1 to the vector transport by rigging. The parameter ℓ in LRBFGS is set to 4. The performance of LRBFGS is compared to that of a Riemannian conjugate gradient algorithm (RCG) defined in [2] that is suitable for large-scale problems. RCG uses a modified Polak-Ribiére formula (see [22, (5.45)]) and imposes the strong Wolfe conditions with $c_1 = 10^{-4}$ and $c_2 = 10^{-2}$

$(n,\ p)$	$(1000,\ 2)$		$(1000,\ 3)$		$(1000,\ 4)$		$(1000,\ 5)$	
	LRBFGS	RCG	LRBFGS	RCG	LRBFGS	RCG	LRBFGS	RCG
iter	2.33_2	2.38_2	3.68_2	4.41_2	4.49_2	4.78_2	5.26_2	5.44_2
nf	2.36_2	7.53_2	3.74_2	1.38_3	4.54_2	1.48_3	5.31_2	1.66_3
ng	2.33_2	7.50_2	3.69_2	1.38_3	4.49_2	1.48_3	5.26_2	1.66_3
nV	2.09_3	9.86_2	3.30_3	1.81_3	4.03_3	1.95_3	4.72_3	2.20_3
nR	2.35_2	7.52_2	3.73_2	1.38_3	4.53_2	1.48_3	5.30_2	1.66_3
gf_f	1.69_{-4}	1.90_{-4}	2.98_{-4}	3.21_{-4}	4.33_{-4}	4.76_{-4}	5.98_{-4}	6.49_{-4}
$\frac{gf_f}{gf_0}$	8.53_{-7}	9.57_{-7}	8.95_{-7}	9.65_{-7}	8.89_{-7}	9.76_{-7}	9.04_{-7}	9.81_{-7}
t	8.07_{-1}	1.02	1.70	2.47	2.70	3.75	4.48	6.52

Table 4

Comparison of LRBFGS and RCG for average of 10 random runs. The subscript $-k$ indicates a scale of 10^{-k}.

by using [22, Algorithm 3.5].

The performance results for RCG and LRBFGS for several values of $(n,\ p)$ are shown in Table 4. The reductions of the norm of the initial gradient are comparable so both algorithms provide similar optimization performance. However, the computation time required by LRBFGS to achieve the reduction is notably smaller than the computation time required by RCG. The number of iterations for LRBFGS is smaller or comparable to RCG but the main source of the difference in computation time is seen in the much larger numbers of function and gradient evaluations required by RCG. This is due to the line search having difficulty satisfying the Wolfe conditions and we conclude that LRBFGS is a viable approach for large-scale problems characterized by Brockett's cost function.

8. Conclusion.

We have developed a new generalization of the Broyden family of optimization algorithms to solve problems on a Riemannian manifold. The Riemannian locking condition was defined to provide a simpler approach to guaranteeing convergence of the RBroyden family while allowing a relatively efficient implementation for many manifolds. Global convergence for a retraction-convex function was also shown. Superlinear convergence of the RBroyden family is known to hold and will be discussed in a forthcoming paper.

Methods of deriving efficient isometric vector transport and an associated retraction were given and a limited-memory version of the RBFGS was described for storage and computational efficiency for large-scale problems. The potential of the methods was illustrated using a range of problem sizes for Brockett's cost function and comparison to Ring and Wirth's RBFGS and RCG.

Future work will address more comprehensive analysis of the choice of the parameters $\tilde{\phi}_k$ and ϕ_k. Even though the explicit computation of the differentiated retraction is not required by our new methods, some information about it is required. A critical future task is the development and analysis of additional methods to derive computationally efficient retractions and isometric vector transports that require even less information about the associated differentiated retraction or avoid it completely.

9. Acknowledgements. The authors appreciate the careful reading and helpful comments and suggestions of two anonymous referees.

REFERENCES

[1] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, NJ, 2008.

[3] R. L. Adler, J.-P. Dedieu, and J. Y. Margulies. Newton's method on Riemannian manifolds and a geometric model for the human spine. *IMA Journal of Numerical Analysis*, 22(3):359–390, 2002.

[4] B. Afsari, R. Tron, and R. Vidal. On the convergence of gradient descent for finding the Riemannian center of mass. *SIAM Journal on Control and Optimization*, 51(3):2230–2260, 2013. arXiv:1201.0925v1.

[5] C. G. Baker, P.-A. Absil, and K. A. Gallivan. An implicit trust-region method on Riemannian manifolds. *IMA Journal of Numerical Analysis*, 28(4):665–689, 2008.

[6] D. A. Bini and B. Iannazzo. Computing the Karcher mean of symmetric positive definite matrices. *Linear Algebra and its Applications*, 438(4):1700–1710, February 2013. doi:10.1016/j.laa.2011.08.052.

[7] W. M. Boothby. *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press, second edition, 1986.

[8] N. Boumal and P.-A. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 406–414, 2011.

[9] I. Brace and J. H. Manton. An improved BFGS-on-manifold algorithm for computing weighted low rank approximations. *Proceedings of 17th international Symposium on Mathematical Theory of Networks and Systems*, pages 1735–1738, 2006.

[10] R. H. Byrd, D. C. Liu, and J. Nocedal. On the behavior of Broyden's class of quasi-Newton methods. *SIAM Journal on Optimization*, 2(4):533–557, 1992.

[11] R. H. Byrd, J. Nocedal, and Y.-X. Yuan. Global convergence of a class of quasi-Newton methods on convex problems. *SIAM Journal on Numerical Analysis*, 24(5):1171–1190, 1987.

[12] W. Dai, E. Kerman, and O. Milenkovic. A geometric approach to low-rank matrix completion. *IEEE Transactions on Information Theory*, 58(1):237–247, 2012. arXiv:1006.2086v1.

[13] W. C. Davidon. Optimally conditioned optimization algorithms without line searches. *Mathematical Programming*, 9(1):1–30, 1975.

[14] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Springer, New Jersey, 1983.

[15] D Gabay. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219, 1982.

[16] K. A. Gallivan, C. Qi, and P.-A. Absil. A Riemannian Dennis-Moré condition. In Michael W. Berry, Kyle A. Gallivan, Efstratios Gallopoulos, Ananth Grama, Bernard Philippe, Yousef Saad, and Faisal Saied, editors, *High-Performance Scientific Computing*, pages 281–293. Springer London, 2012. doi:10.1007/978-1-4471-2437-5_14.

[17] W. Huang. *Optimization algorithms on Riemannian manifolds with applications*. PhD thesis, Florida State University, Department of Mathematics, 2013.

[18] W. Huang, P.-A. Absil, and K. A. Gallivan. A Riemannian symmetric rank-one trust-region method. *Mathematical Programming*, February 2014. doi:10.1007/s10107-014-0765-1.

[19] C. Liu and S. A. Vander Wiel. Statistical quasi-Newton: a new look at least change. *SIAM Journal on Optimization*, 18(4):1266–1285, 2007.

[20] D. G. Luenberger. The gradient projection method along geodesics. *Management Science*, 18(11):620–631, 1972.

[21] B. Mishra, G. Meyer, and R. Sepulchre. Low-rank optimization for distance matrix completion. *In Proceeding of 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4455–4460, December 2011. doi:10.1109/CDC.2011.6160810.

[22] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, second edition, 2006.

[23] C. Qi. *Numerical optimization methods on Riemannian manifolds*. PhD thesis, Florida State University, Department of Mathematics, 2011.

[24] C. Qi, K. A. Gallivan, and P.-A. Absil. An efficient BFGS algorithm for Riemannian optimization. *Proceedings of the 19th International Symposium on Mathematical Theory of Network and Systems (MTNS 2010)*, (1):2221–2227, 2010.

[25] C. Qi, K. A. Gallivan, and P.-A. Absil. Riemannian BFGS algorithm with applications. *Recent Advances in Optimization and its Applications in Engineering*, pages 183–192, 2010.

[26] W. Ring and B. Wirth. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627, January 2012. doi:10.1137/11082885X.

[27] O. Sander. Geodesic finite elements for Cosserat rods. *International Journal for Numerical Methods in Engineering*, 82(13):1645–1670, 2010. doi:10.1002/nme.2814.

[28] B. Savas and L. H. Lim. Quasi-Newton methods on Grassmannians and multilinear approxi-

mations of tensors. *SIAM Journal on Scientific Computing*, 32(6):3352–3393, 2010.

[29] M. Seibert, M. Kleinsteuber, and K. Hüper. Properties of the BFGS method on Riemannian manifolds. *Mathematical System Theory C Festschrift in Honor of Uwe Helmke on the Occasion of his Sixtieth Birthday*, pages 395–412, 2013.

[30] S. E. Selvan, U. Amato, K. A. Gallivan, and C. Qi. Descent algorithms on oblique manifold for source-adaptive ICA contrast. *IEEE Transactions on Neural Networks and Learning Systems*, 23(12):1930–1947, 2012.

[31] H. Shen, K. Hüper, and M. Kleinsteuber. On fastICA algorithms and some generalisations. *Numerical Linear Algebra in Signals, Systems and Control*, 80:403–432, 2011. doi:10.1007/978-94-007-0602-6_19.

[32] S. T. Smith. Optimization techniques on Riemannian manifolds. *Hamiltonian and Gradient Flows, Algorithms and Control*, 3:113–136, 1994.

[33] B. Vandereycken. Low-rank matrix completion by Riemannian optimization—extended version. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

[34] Y. Zhang and R. P. Tewarson. Quasi-Newton algorithms with updates from the preconvex part of Broyden's family. *IMA Journal of Numerical Analysis*, 8(4):487–509, 1988.