Found. Comput. Math. 303–330 (2007)
© 2007 SFoCM
DOI: 10.1007/s10208-005-0179-9

FOUNDATIONS OF
COMPUTATIONAL
MATHEMATICS
The Journal of the Society for the Foundations of Computational Mathematics

Trust-Region Methods on Riemannian Manifolds

P.-A. Absil,[1] C. G. Baker,[2] and K. A. Gallivan[2]

[1]Department of Mathematical Engineering
Université catholique de Louvain
1348 Louvain-la-Neuve, Belgium
URL: http://www.inma.ucl.ac.be/∼absil/
and
University of Cambridge
Cambridge CB2 1RD
England

[2]School of Computational Science
Florida State University
Tallahassee, FL 32306-4120, USA
URLs: http://www.csit.fsu.edu/{∼cbaker,∼gallivan}

Abstract. A general scheme for trust-region methods on Riemannian manifolds is proposed and analyzed. Among the various approaches available to (approximately) solve the trust-region subproblems, particular attention is paid to the truncated conjugate-gradient technique. The method is illustrated on problems from numerical linear algebra.

1. Introduction

Several problems related to numerical linear algebra can be expressed as optimizing a smooth function whose domain is a differentiable manifold. Applications

Date received: March 24, 2005. First version received: December 20, 2005. Date accepted: June 26, 2006. Communicated by Mike Todd. Online publication: February 7, 2007.
AMS classification: 65K05, 65J05, 90C26, 90C48, 58C05.
Key words and phrases: Numerical optimization on manifolds, Trust-region, Truncated conjugate-gradient, Steihaug–Toint, Global convergence, Local Convergence, Superlinear convergence, Symmetric eigenvalue problem.

appear in various areas, including computer vision [**MKS01**], machine learning [**NA05**], maximum likelihood estimation [**Smi05**], [**XB05**], electronic structure computation [**LE00**], system balancing [**HM94**], model reduction [**YL99**], and robot manipulation [**HHM02**].

The simplest algorithms for solving optimization problems on manifolds are arguably those based on the idea of steepest descent; see, e.g., [**HM94**], [**Udr94**] and references therein. These algorithms have good global convergence properties but slow (linear) local convergence. Other methods achieve superlinear convergence by using second-order information on the cost function. Among these methods, Newton's method is conceptually the simplest. The history of Newton's method on manifolds can be traced back to Luenberger [**Lue72**], if not earlier. Gabay [**Gab82**] proposed a Newton method on embedded submanifolds of $\mathbb{R}^n$. Smith [**Smi93**], [**Smi94**] and Udrişte [**Udr94**] formulated and analyzed the method on general Riemannian manifolds. Related work includes [**Shu86**], [**EAS98**], [**OW00**], [**Man02**], [**MM02**], [**ADM$^+$02**], [**DPM03**], [**HT04**].

A plain Newton method, on general manifolds as well as in $\mathbb{R}^n$, has major drawbacks as a numerical optimization method. The computational cost is often prohibitive, as a linear system has to be solved at each iteration. Moreover, the method is locally attracted to any stationary point, be it a local minimizer, local maxizer, or saddle point. Finally, the method may not even converge to stationary points, unless some strong conditions are statisfied (such as convexity of the cost function).

In the case of cost functions on $\mathbb{R}^n$, several techniques exist to improve the convergence properties of Newton's method. Most of these techniques fall into two categories: line-search methods and trust-region methods; see [**NW99**]. The advantages of a trust-region method over the pure Newton method are multiple. First, under mild conditions, trust-region schemes are provably convergent to a set of stationary points of the cost functions for all initial conditions. Moreover, the cost function is nonincreasing at each iterate which favors convergence to a local minimizer. Finally, the presence of a trust-region gives an additional guideline to stop the inner iteration early, hence reducing the computational cost, while preserving the fast local convergence of the exact scheme. Line-search techniques have been considered on Riemannian manifolds by Udrişte [**Udr94**] and Yang [**Yan06**].

The main purpose of this paper is to provide a theoretical and algorithmic framework for trust-region methods on Riemannian manifolds, applicable to multiple problems. The Riemannian trust-region (RTR) approach we propose works along the following lines. First, much as in the work of Shub [**Shu86**], [**ADM$^+$02**], a *retraction* R (Definition 2.1) is chosen on the Riemannian manifold M that defines, for any point $x \in M$, a one-to-one correspondence R_x between a neighborhood of x in M and a neighborhood of 0_x in the tangent space T_xM (see Figure 1). Using this retraction, the cost function f on M is lifted to a cost function $\hat{f}_x = f \circ R_x$ on T_xM. Since T_xM is a Euclidean space, it is possible to define a quadratic model of $\hat{f}_x$ and adapt classical methods in $\mathbb{R}^n$ to compute (in general, approximately) a minimizer of the model within a trust-region around $0_x \in T_xM$. This minimizer is then retracted back from T_xM to M using the retraction R_x. This point is a candidate for the new iterate, which will be accepted or rejected depending on the quality of the agreement between the quadratic model and the function f itself.

It is this "lift-solve-retract" procedure that distinguishes the proposed RTR approach from the standard trust-region methods; the standard methods, since they live in $\mathbb{R}^n$, only require the "solve" part. On a manifold, lifting the cost function makes it possible to locally fall back to a friendly Euclidean world (the tangent space $T_x M$) where classical techniques can be applied, and the retraction brings the result back to the manifold. A difficulty, from an analysis perspective, is that the RTR method does not deal with a unique cost function (as in the classical case), but rather with a succession of different lifted cost functions $\hat{f}_{x_k}$, where x_k is the kth iterate. A main contribution of this paper is to show that, under reasonable conditions, the nice properties of the standard trust-region schemes are preserved in their Riemannian generalizations (see Section 4).

Notice that it is theoretically possible to choose once and for all the retraction as the Riemannian exponential mapping. This corresponds to a strategy used by numerous Riemannian optimization methods when they compute the exponential of a tangent update vector in order to obtain a new iterate on the manifold; see [**Smi94**], [**Udr94**], [**EAS98**], [**Yan06**]. However, as pointed out by Manton [**Man02**, Sec. IX], the exponential map may not always be the most appropriate or computationally efficient way of performing the update. Our convergence analysis shows that the good properties of the algorithms hold for all suitably defined retractions (Definition 2.1) and not only for the exponential mapping.

The "lift-solve-retract" technique, as an aside, is not specific to trust-region methods, and can be applied to generalize a wide variety of classical optimization methods to manifolds. This approach, which finds its roots in the work of Shub [**Shu86**], seems to have received little attention in the literature until recently [**ADM$^+$02**], [**ABG04**]. Note that coordinate-based approaches follow a similar pattern; in practice, however, lifting to the tangent space tends to lead to more streamlined and computationally efficient algorithms (see the discussion in Section 2.1).

There is clearly a link between techniques of optimization on manifolds and standard constrained optimization approaches. However, there are manifolds that are not defined as constrained sets in $\mathbb{R}^n$; an important example is the Grassmann manifold (see Section 5.3). (Clearly, by Nash's embedding theorem [**Nas56**], every Riemannian manifold can be smoothly isometrically embedded in a Euclidean space; but this is only an existence theorem, and such an embedding may be elusive or computationally intractable.) Also, there are constrained sets that do not admit a regular manifold structure; a simple example is $\{x \in \mathbb{R}^n : \|x\|_\infty = 1\}$. The application areas thus overlap, but are not identical. On the problems that can be tackled by both approaches, an interesting feature of manifold optimization schemes is that they are *feasible*: each iterate belongs to the constrained set. Feasibility is advantageous, or even essential, in several situations (see, e.g., [**LT01**]). For example, the cost function is sometimes not defined outside the feasible set; or, the value of the cost function may have little if any relevance outside the feasible set; moreover, if the algorithm runs out of time or computing resources, it should be able to terminate and return a feasible point. Note that a crucial requirement in this feasible approach is the availability of a computationally tractable retraction. In the absence of such a retraction, and assuming that the manifold admits a description as the level set of a (vector-valued) function in $\mathbb{R}^n$, a remedy is to fall back to an infeasible method, such as the trust-region SQP algorithms considered in [**CGT00**], [**HV01**], [**Rid06**] and references therein.

We assume throughout that it is computationally impractical to determine whether the Hessian of the cost function is positive definite; trust-region subproblems are thus solved using inner iterations, such as the truncated conjugate-gradient method, that improve on the so-called Cauchy point by only using the Hessian of the model through its application to a vector. As a consequence, convergence of the trust-region algorithm to stationary points that are not local minima (i.e., saddle points and local maxima) cannot be ruled out. However, because trust-region methods are descent methods (the value of the cost function never increases), the situation is fundamentally different from the pure Newton case: convergence to saddle points and local minima of the cost function is numerically unstable and is thus not expected to occur in practice; and indeed, convergence to saddle points and local minima is only observed on very specifically crafted numerical experiments.

The theory and algorithms can be adapted to exploit the properties of specific manifolds and problems in several disciplines. Numerical linear algebra considers several problems that can be analyzed and solved using this approach. A particularly illustrative and computationally efficient application is the computation of the rightmost or leftmost eigenvalue and associated eigenvector of a symmetric/positive-definite matrix pencil (A, B). In this case, the manifold can be chosen as the projective space and a possible choice for the cost function is the Rayleigh quotient. The resulting trust-region algorithm can be interpreted as an inexact Rayleigh quotient iteration; we refer to [**ABG06b**] for details.

This paper makes use of basic notions of Riemannian geometry and numerical optimization; background can be found in [**dC92**] and [**NW99**]. The general concept of trust-region methods on Riemannian manifolds is presented in Section 2. Methods for (approximately) solving the trust-region subproblems are considered in Section 3. Convergence properties are investigated in Section 4. The theory is illustrated on practical examples in Section 5. In particular, a block algorithm for the computation of extreme eigenpairs, which evolves on the Grassmann manifold, is derived in detail in Section 5.3. Conclusions are presented in Section 6.

A preliminary version of the results presented in this paper appeared in the Proceedings of the 16th MTNS Conference [**ABG04**].

2. The General Algorithm

We follow the usual conventions of matrix computations and view $\mathbb{R}^n$ as the set of column vectors with n real components. The basic trust-region method in $\mathbb{R}^n$ for a cost function f consists of adding to the current iterate $x \in \mathbb{R}^n$ the update vector $\eta \in \mathbb{R}^n$ solving *the trust-region subproblem*

$$\min_{\eta \in \mathbb{R}^n} m(\eta) = f(x) + \partial f(x)\eta + \tfrac{1}{2}\eta^T \partial^2 f(x)\eta, \qquad \|\eta\| \le \Delta, \tag{1}$$

where $\partial f = (\partial_1 f, \ldots, \partial_n f)$ is the differential of f, $(\partial^2 f)_{ij} = \partial^2_{ij} f$ is the Hessian matrix—some convergence results allow for $\partial^2 f(x)$ in (1) to be replaced by any symmetric matrix, but we postpone this relaxation until later in the development—and Δ is the trust-region radius. The quality of the model m is assessed by forming the quotient

$$\rho = \frac{f(x) - f(x+\eta)}{m(0) - m(\eta)}. \tag{2}$$

Depending on the value of ρ, the new iterate will be accepted or discarded and the trust-region radius Δ will be updated. More details will be given later in this paper. Major references on trust-region methods include the early work of Powell [**Pow74**] and Moré and Sorensen [**Sor82**], [**MS84**], and the textbook [**CGT00**]; see also [**WD05**] and references therein for recent work on trust-region update strategies.

With a view toward extending the concept of trust-region subproblems to manifolds, we first consider the case of an abstract *Euclidean space*, i.e., a vector space endowed with an inner product (i.e., a symmetric, bilinear, positive-definite form). This generalization to a Euclidean space E of dimension d requires little effort since E may be identified with $\mathbb{R}^d$ once a basis of E is chosen (we refer to [**Boo75**, Sec. I.2] for a discussion on the distinction between $\mathbb{R}^n$ and abstract Euclidean spaces). Let $g(\cdot,\cdot)$ denote the inner product on E. Given a function $f : E \to \mathbb{R}$ and a current iterate $x \in E$, one can choose a basis $(e_i)_{i=1,\ldots,d}$ of E (not necessarily orthonormal with respect to the inner product) and write a classical G-norm trust-region subproblem (see, e.g., [**GLRT99**, Sec. 2])

$$\min_{\bar\eta \in \mathbb{R}^d} m(\bar\eta) := \bar f(\bar x) + \partial \bar f(\bar x)\bar\eta + \tfrac{1}{2}\bar\eta^T \partial^2 \bar f(\bar x)\bar\eta, \qquad \bar\eta^T G \bar\eta \le \Delta_x^2, \tag{3}$$

where $x = \sum_i \bar x_i e_i$, $\eta = \sum_i \bar\eta_i e_i$, $\bar f(\bar x) = f\left(\sum_i \bar x_i e_i\right)$, and $G_{ij} = g(e_i, e_j)$. It can be shown that $m(\eta)$ does not depend on the choice of basis $(e_i)_{i=1,\ldots,d}$; therefore (3) can be written as a coordinate-free expression

$$\begin{aligned}\min_{\eta \in E} m(\eta) &= f(x) + Df(x)[\eta] + \tfrac{1}{2}D^2 f(x)[\eta,\eta] \\ &= f(x) + g(\operatorname{grad} f(x), \eta) + \tfrac{1}{2}g(\operatorname{Hess} f[\eta], \eta) \quad \text{s.t. } g(\eta,\eta) \le \Delta_x^2\end{aligned} \tag{4}$$

for the trust-region subproblem in the Euclidean space E.

Now let M be a *manifold* of dimension d. Intuitively, this means that M looks locally like $\mathbb{R}^d$. Local correspondences between M and $\mathbb{R}^d$ are given by coordinate charts $\varphi_\alpha : \Omega_\alpha \subset M \to \mathbb{R}^n$; see, e.g., [**dC92**] for details. How can we define a trust-region method for a cost function f on M? Given a current iterate x, it is tempting to choose a coordinate neighborhood Ω_α containing x, translate the problem to $\mathbb{R}^d$ through the chart φ_α, build a quadratic model m, solve the trust-region problem in $\mathbb{R}^d$, and bring back the solution to M through φ_α^{-1}. The difficulty is that there are in general infinitely many α's such that $x \in \Omega_\alpha$. Each choice will yield a different model function $m \circ \varphi_\alpha$ and a different trust region $\{y \in M : \|\varphi_\alpha(y)\| \leq \Delta\}$, hence a different next iterate x_+. This kind of situation is pervasive in numerics on manifolds; it is usually addressed, assuming that M is a Riemannian manifold, by working in so-called *normal coordinates*.

In order to explain the concept of normal coordinates, we now present a condensed overview of Riemannian geometric concepts; we refer to [**dC92**], [**O'N83**] for details. In what follows, M will be a (C^∞) Riemannian manifold, i.e., M is endowed with a correspondence, called a Riemannian metric, which associates to each point x of M an inner product $g_x(\cdot, \cdot)$ on the tangent space T_xM and which varies differentiably. The Riemannian metric induces a norm $\|\xi\| = \sqrt{g_x(\xi, \xi)}$ on the tangent spaces T_xM. Also associated with a Riemannian manifold are the notions of Levi-Civita (or Riemannian) connection ∇, parallel transport, geodesic (which, intuitively, generalizes the notion of straight line), and associated exponential map defined by $\mathrm{Exp}_x\, \xi = \gamma(1)$ where γ is the geodesic satisfying $\gamma(0) = x$ and $\gamma'(0) = \xi$, with $\gamma'(0)$ denoting the tangent vector to the curve γ at $t = 0$. Given a point x in M, there is a ball $B_\varepsilon(0_x)$ in T_xM of radius ε around the origin 0_x of T_xM such that Exp_x is a diffeomorphism of $B_\varepsilon(0_x)$ onto an open subset of M. Then $\mathrm{Exp}_x(B_\varepsilon(0_x)) = U$ is called a *normal neighborhood* of x, and Exp_x defines a diffeomorphism between the Euclidean space T_xM and U. The supremum of these ε's is the *injectivity radius* $i_x(M)$ at x, and $i(M) := \inf_{x\in M} i_x$ is the *injectivity radius* of M. Finally, normal coordinates are defined in a normal neighborhood U by considering an orthonormal basis $\{e_i\}$ of T_xM and taking $(u_1, \dots, u_d)$ as the coordinates of $y = \mathrm{Exp}_x(\sum_{i=1}^n u_i e_i)$.

An important observation is that, for the purpose of defining a trust-region method, the choice of a basis $\{e_i\}$ in T_xM is immaterial, since trust-region subproblems on a Euclidean space like T_xM admit a coordinate-free expression (4). Therefore, the exponential mapping makes it possible to uniquely define trust-region subproblems on Riemannian manifolds by locally mapping the manifold to the Euclidean space T_xM.

However, as pointed out in [**Man02**], the systematic use of the exponential mapping may not be desirable in all cases: other local mappings to T_xM may reduce the computational cost while preserving the useful convergence properties of the considered method. Therefore, in this paper, we relax the exponential to a class of mappings called *retractions*, a concept that we borrow from [**Shu86**], [**ADM$^+$02**] with some modifications (see also the illustration in Figure 1).

Definition 2.1 (Retraction). A *retraction* on a manifold M is a mapping R on the tangent bundle TM into M with the following properties. Let R_x denote the

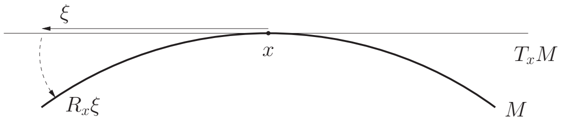

Fig. 1. Illustration of retractions.

restriction of R to T_xM.

(1) R is continuously differentiable.
(2) $R_x(0_x) = x$, where 0_x denotes the zero element of T_xM.
(3) $DR_x(0_x) = \mathrm{id}_{T_xM}$, the identity mapping on T_xM, with the canonical identification $T_{0_x}T_xM \simeq T_xM$.

It follows from the inverse function theorem (see [**dC92**, Chap. 0, Theorem 2.10]) that R_x is a local diffeomorphism at 0_x, namely, R_x is not only C^1 but also bijective with differentiable inverse on a neighborhood V of 0_x in T_xM. In particular, the exponential mapping is a retraction (see Proposition 2.9 in [**dC92**, Chap. 3] and the proof thereof), and any other retraction can be thought of as a first-order approximation of the exponential mapping. Notice that no assumption is made on the second and higher derivatives of the retractions; in particular, $D^2(\mathrm{Exp}_x^{-1} \circ R_x)(0_x)$ need not vanish (see also the discussion in Section 2.1). Practical examples of retractions on specific Riemannian manifolds, that may be more tractable computationally than the exponential, are given in Section 5. We point out that the requirements in Definition 2.1 are stronger than needed to obtain the convergence results; in particular, we could allow R to be defined only in a certain subset of TM. However, weaker assumptions would make the forthcoming developments more complicated, and there is no evidence that they would be more relevant in practical applications. For the same reason, we assume throughout that the manifold M is complete, i.e., $\mathrm{Exp}\,\xi$ exists for all ξ in TM.

We can now lay out the structure of a trust-region method on a Riemannian manifold (M, g) with retraction R. Given a cost function $f : M \to \mathbb{R}$ and a current iterate $x_k \in M$, we use R_{x_k} to locally map the minimization problem for f on M into a minimization problem for the cost function

$$\hat{f}_{x_k} : T_{x_k}M \to \mathbb{R} : \xi \mapsto f(R_{x_k}(\xi)). \tag{5}$$

The Riemannian metric g turns $T_{x_k}M$ into a Euclidean space endowed with the inner product $g_{x_k}(\cdot,\cdot)$, and, following (4), the trust-region subproblem on $T_{x_k}M$ reads

$$\begin{aligned} \min_{\eta \in T_{x_k}M} m_{x_k}(\eta) &= \hat{f}_{x_k}(0_{x_k}) + D\hat{f}_{x_k}(0_{x_k})[\eta] + \tfrac{1}{2}D^2\hat{f}_{x_k}(0_{x_k})[\eta,\eta] \\ &= \hat{f}_{x_k}(0_{x_k}) + g_{x_k}(\operatorname{grad}\hat{f}_{x_k}(0_{x_k}),\eta) \\ &\quad + \tfrac{1}{2}g_{x_k}(\operatorname{Hess}\hat{f}_{x_k}(0_{x_k})[\eta],\eta) \quad \text{s.t. } g_{x_k}(\eta,\eta) \le \Delta_k^2. \end{aligned} \tag{6}$$

For the global convergence theory it is only required that the second-order term in the model be some symmetric form. Therefore, instead of (6), we will consider the following more general formulation

$$\min_{\eta \in T_{x_k} M} m_{x_k}(\eta) \;=\; f(x_k) + g_{x_k}(\operatorname{grad} f(x_k), \eta) + \tfrac{1}{2} g_{x_k}(\mathcal{H}_{x_k}\eta, \eta)$$
$$\text{s.t. } g_{x_k}(\eta, \eta) \leq \Delta_k^2, \tag{7}$$

where $\mathcal{H}_{x_k} : T_{x_k}M \to T_{x_k}M$ is some symmetric linear operator, i.e., $g_{x_k}(\mathcal{H}_{x_k}\xi, \chi) = g_{x_k}(\xi, \mathcal{H}_{x_k}\chi)$, $\xi, \chi \in T_x M$. This is called the *trust-region subproblem*.

Next, an (approximate) solution η_k of the Euclidean trust-region subproblem (7) is computed using any available method: if an iterative method is used, its iterations are called inner iterations of the overall algorithm (see Section 3). The candidate for the new iterate is then given by $x_+ = R_{x_k}(\eta_k)$.

The decision to accept or not the candidate and to update the trust-region radius is based on the quotient

$$\rho_k = \frac{f(x_k) - f(R_{x_k}(\eta_k))}{m_{x_k}(0_{x_k}) - m_{x_k}(\eta_k)} = \frac{\hat{f}_{x_k}(0_{x_k}) - \hat{f}_{x_k}(\eta_k)}{m_{x_k}(0_{x_k}) - m_{x_k}(\eta_k)}. \tag{8}$$

If ρ_k is exceedingly small, then the model is very inaccurate: the step must be rejected and the trust-region radius must be reduced. If ρ_k is small but less dramatically so, then the step is accepted but the trust-region radius is reduced. If ρ_k is close to 1, then there is good agreement between the model and the function over the step, and the trust-region radius can be expanded.

This procedure can be formalized as the following algorithm; it reduces to [**NW99**, Alg. 4.1] in the classical $\mathbb{R}^n$ case (see [**CGT00**, Chap. 10] for variants).

Algorithm 1 (RTR—basic Riemannian Trust-Region algorithm).
Data: *Complete Riemannian manifold* (M, g); *real function* f *on* M; *retraction* R *from* TM *to* M *as in Definition* 2.1.
Parameters: $\bar{\Delta} > 0$, $\Delta_0 \in (0, \bar{\Delta})$, *and* $\rho' \in [0, \frac{1}{4})$.
Input: *initial iterate* $x_0 \in M$.
Output: *sequence of iterates* $\{x_k\}$.

```
for $k = 0, 1, 2, \ldots$
    Obtain $\eta_k$ by (approximately) solving (7);
    Evaluate $\rho_k$ from (8);
    if $\rho_k < \frac{1}{4}$
        $\Delta_{k+1} = \frac{1}{4}\Delta_k$
    else if $\rho_k > \frac{3}{4}$ and $\|\eta_k\| = \Delta_k$
        $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$
    else
        $\Delta_{k+1} = \Delta_k$;
    if $\rho_k > \rho'$
        $x_{k+1} = R_x(\eta_k)$
    else
        $x_{k+1} = x_k$;
end (for).
```

In the sequel we will sometimes drop the subscript "k" and denote x_{k+1} by x_+.

2.1. *Discussion*

The concept of retraction (Definition 2.1) is related to the notion of a smooth family of parametrizations [**HT04**]. To any smooth family of parametrizations $\{\mu_x\}_{x\in M}$, one can associate a retraction defined by $R_x := \mu_x \circ (D\mu_x(0_x))^{-1}$. Conversely, given smooth vector fields e_i, $i = 1, \ldots, d$, such that $\{e_i(x)\}_{i=1,\ldots,d}$ is a basis of T_xM for all x in a neighbourhood U of x^*, to a retraction R one can associate the mappings $\mu_x : (u_1, \ldots, u_d) \mapsto R_x(\sum u_i e_i(x))$ which form a locally smooth family of parametrizations around x^*. Notice that in general this technique cannot be applied globally: while a smooth basis $\{e_i(x)\}_{i=1,\ldots,d}$ always exists locally, there are manifolds for which such a basis fails to exist globally. (An example is a sphere embedded in a space of odd dimension: a result informally known as the "hairy ball theorem" states that there is no nonvanishing smooth vector field on such a sphere, hence there is no globally smooth basis of vector fields.) Moreover, from a computational viewpoint, such a basis of vector fields may be difficult or even impossible to deal with. The use of retractions in the context of trust-region methods is thus particularly convenient, as it takes advantage of the fact that trust-region methods (and optimization methods in general) can be defined on general Euclidean spaces, devoid of a particular basis. We refer to Section 5 for illustrations on concrete examples.

From the "first-order rigidity condition" $DR_x(0_x) = \mathrm{id}_{T_xM}$ satisfied by retractions (Definition 2.1), it follows that $\operatorname{grad} \hat{f}_{x_k}(0_{x_k}) = \operatorname{grad} f(x_k)$, where $\operatorname{grad} f(x)$, the gradient of f at x, is defined by $g_x(\operatorname{grad} f(x), \xi) = \mathrm{d}f_x(\xi)$, $\xi \in T_xM$ (see [**dC92**, Chap. 3, Ex. 8]). Since no assumption is made on the second- and higher-order derivatives of the retraction (such conditions are not necessary in the local and global convergence analyses carried out in Section 4), it follows that in general $\operatorname{Hess} \hat{f}(0_x) \neq \operatorname{Hess} f(x)$, where

$$\operatorname{Hess} f(x) : T_xM \mapsto T_xM : \xi \mapsto \operatorname{Hess} f(x)[\xi] := \nabla_\xi \operatorname{grad} f(x) \qquad (9)$$

is the Hessian operator of f at x, as defined in [**dC92**, Chap. 6, Ex. 11]. In Lemma 4.11, second-order conditions are given on the retraction such that $\operatorname{Hess} \hat{f}_{x_k}(0_x) = \operatorname{Hess} f(x)$. In the case of embedded submanifolds, the Levi-Civita connection ∇ reduces to a directional derivative in the embedding space followed by a projection onto the tangent space to the manifold [**O'N83**, §4.3]; this facilitates the derivation of a formula for $\operatorname{Hess} f(x)[\xi]$. (Notice that in the literature, the word *Hessian* is sometimes used for the second covariant differential $D^2 f$, which is related to the Hessian operator (9) by the identity $D^2 f(\xi, \chi) = g_x(\operatorname{Hess} f(x)[\xi], \chi)$ [**O'N83**].)

In general, there is no assumption on the operator $\mathcal{H}_{x_k}$ in (7) other than being a symmetric linear operator. Consequently, even though m_{x_k} was initially presented as a model of $f \circ R_{x_k}$, the choice of the retraction R_{x_k} does not impose any constraint on m_{x_k}. In order to achieve superlinear convergence, however, $\mathcal{H}_{x_k}$ will be required to be an "approximate" Hessian (Theorem 4.13). Obtaining an appropriate approximate Hessian in practice is addressed in Section 5.1. A possible way of choosing $\mathcal{H}_{x_k}$ is to define m_x as the quadratic model of $f \circ \tilde{R}_{x_k}$, where $\tilde{R}_x$ is a retraction, not necessarily equal to R_{x_k}; a similar point of view was adopted in [**HT04**] in the framework of Newton's method.

We conclude this section by pointing out more explicitly the link between Algorithm 1 and the Riemannian Newton method. Assume that $\mathcal{H}_{x_k}$ in (7) is the exact Hessian of f at x_k, and assume that the exact solution η^* of the trust-region subproblem (7) lies in the interior of the trust region. Then η^* satisfies

$$\operatorname{grad} f + \nabla_{\eta^*} \operatorname{grad} f = 0,$$

which is the Riemannian Newton equation of Smith [**Smi93**], [**Smi94**] and Udrişte [**Udr94**, Chap. 7, §5]. Note that both authors propose to apply the update vector η^* using the Riemannian exponential retraction; namely, the new iterate is defined as $x_+ = \mathrm{Exp}_x\, \eta^*$. As shown by Smith [**Smi93**], [**Smi94**], the Riemannian Newton algorithm converges locally quadratically to the nondegenerate stationary points of f. A cubic rate of convergence is even observed in frequently encountered cases where some symmetry condition holds [**Smi93**], [**Smi94**], [**EAS98**], [**AMS04**]. We will see in Section 4 that the superlinear convergence property of Newton's method is preserved by the trust-region modification, while the global convergence properties are improved: the accumulation points are guaranteed to be stationary points regardless of the initial conditions, and among the stationary points only the local minima can be local attractors.

3. Computing a Trust-Region Step

We have seen in Section 2 that the use of retractions yields trust-region subproblems expressed in Euclidean spaces $T_x M$. Therefore, all the classical methods for solving the trust-region subproblem can be applied.

As mentioned in the Introduction, it is assumed here that for some reason, usually related to the large size of the problem under consideration or to the computational efficiency required to outperform alternative methods, it is impractical to check the positive-definiteness of $\mathcal{H}_{x_k}$; rather, $\mathcal{H}_{x_k}$ is only available via its application to a vector. The *truncated conjugate-gradient method* of Steihaug [**Ste83**] and Toint [**Toi81**] is particularly appropriate in these circumstances. The following algorithm is a straightforward adaptation of the method of [**Ste83**] to the trust-region subproblem (7). This algorithm is an *inner iteration* as it is an iteration used within the RTR framework (Algorithm 1) to compute an approximate solution of the trust-region subproblems. Note that we use indices in superscript to denote the evolution of η within the inner iteration, while subscripts are used in the outer iteration.

Algorithm 2 (tCG—truncated CG for the trust-region subproblem).

```
Set $\eta^0 = 0$, $r_0 = \operatorname{grad} f(x_k)$, $\delta_0 = -r_0$;
for $j = 0, 1, 2, \ldots$ until a stopping criterion is satisfied, perform the iteration:
  if $g_{x_k}(\delta_j, \mathcal{H}_{x_k}\delta_j) \leq 0$
    Compute $\tau$ such that $\eta = \eta^j + \tau\delta_j$ minimizes $m_{x_k}(\eta)$ in (7)
      and satisfies $\|\eta\|_{g_x} = \Delta$;
    return $\eta$;
  Set $\alpha_j = g_{x_k}(r_j, r_j)/g_{x_k}(\delta_j, \mathcal{H}_{x_k}\delta_j)$;
  Set $\eta^{j+1} = \eta^j + \alpha_j\delta_j$;
  if $\|\eta^{j+1}\|_{g_x} \geq \Delta$
    Compute $\tau \geq 0$ such that $\eta = \eta^j + \tau\delta_j$ satisfies $\|\eta\|_{g_x} = \Delta$;
    return $\eta$;
  Set $r_{j+1} = r_j + \alpha_j \mathcal{H}_{x_k}\delta_j$;
  Set $\beta_{j+1} = g_{x_k}(r_{j+1}, r_{j+1})/g_{x_k}(r_j, r_j)$;
  Set $\delta_{j+1} = -r_{j+1} + \beta_{j+1}\delta_j$;
end (for).
```

The simplest stopping criterion for Algorithm 2 is to truncate after a fixed number of iterations. In order to improve the convergence rate, a possibility is to stop as soon as an iteration j is reached for which

$$\|r_j\| \leq \|r_0\| \min(\|r_0\|^\theta, \kappa). \tag{10}$$

Concerning the computation of τ, it can be shown that when $g(\delta_j, \mathcal{H}_{x_k}\delta_j) \leq 0$, $\arg\min_{\tau\in\mathbb{R}} m_{x_k}(\eta^j + \tau\delta_j)$ is equal to the positive root of $\|\eta^j + \tau\delta_k\|_{g_x} = \Delta$, which is explicitly given by

$$\frac{-g_x(\eta^j, \delta_j) + \sqrt{g_x(\eta^j, \delta_j)^2 - (\Delta^2 - g_x(\eta^j, \eta^j))g_x(\delta_j, \delta_j)}}{g_x(\delta_j, \delta_j)}.$$

Notice that the tCG algorithm is “inverse-free,” as it uses $\mathcal{H}_{x_k}$ in the computation of $\mathcal{H}_{x_k}\delta_j$ only. The reader interested in the underlying principles of the Steihaug–Toint truncated CG method should refer to [**Ste83**], [**NW99**] or [**CGT00**].

Alternatives to tCG for approximately solving trust-region subproblems are mentioned in [**CGT00**, Sec. 7.5.4]; see also [**Hag01**], [**HP05**].

4. Convergence Analysis

In this section we first study the global convergence properties of the RTR scheme (Algorithm 1), without any assumption on the way the trust-region subproblems (7) are solved, except that the approximate solution η_k must produce a decrease in the model that is at least a fixed fraction of the so-called Cauchy decrease. Under mild additional assumptions on the retraction and the cost function, it is shown that the sequences $\{x_k\}$ produced by Algorithm 1 converge to the set of stationary points of the cost function. This result is well known in the $\mathbb{R}^n$ case; in the case of manifolds, the convergence analysis has to address the fact that a different lifted cost function $\hat{f}_{x_k}$ is considered at each iterate x_k.

We then analyze the local convergence of Algorithm 1-2 around nondegenerate local minima. Algorithm 1-2 refers to the RTR framework where the trust-region subproblems are approximately solved using the tCG algorithm with stopping criterion (10). It is shown that the iterates of the algorithm converge to nondegenerate stationary points with an order of convergence $\min\{\theta + 1, 2\}$ (at least).

4.1. *Global Convergence*

The objective of this section is to show that, under appropriate assumptions, the sequence $\{x_k\}$ generated by Algorithm 1 satisfies $\lim_{k\to\infty} \|\mathrm{grad}\, f(x_k)\| = 0$; this generalizes a classical convergence property of trust-region methods in $\mathbb{R}^n$, see [**NW99**, Theorem 4.8]. In what follows, (M, g) is a complete Riemannian manifold of dimension d, and R is a retraction on M (Definition 2.1). We define

$$\hat{f} : TM \mapsto \mathbb{R} : \xi \mapsto f(R(\xi)) \tag{11}$$

and, in accordance with (5), $\hat{f}_x$ denotes the restriction of $\hat{f}$ to T_xM. We denote by $B_\delta(0_x) = \{\xi \in T_xM : \|\xi\| < \delta\}$ the open ball in T_xM of radius δ centered at 0_x, and $B_\delta(x)$ stands for the set $\{y \in M : \mathrm{dist}(x, y) < \delta\}$ where dist denotes the Riemannian distance (i.e., the distance defined in terms of the Riemannian metric; see, e.g., [**O'N83**, §5.15]). We denote by $P_\gamma^{t\leftarrow t_0} v$ the vector of $T_{\gamma(t)}M$ obtained by parallel transporting the vector $v \in T_{\gamma(t_0)}M$ along a curve γ.

As in the classical $\mathbb{R}^n$ case, we first show that at least one accumulation point of $\{x_k\}$ is stationary. The convergence result requires that $m_{x_k}(\eta_k)$ be a sufficiently good approximation of $\hat{f}_{x_k}(\eta_k)$. In [**CGT00**, Theorem 6.4.5] this is guaranteed by the assumption that the Hessian of the cost function is bounded. It is however possible to weaken this assumption, which leads us to consider the following definition.

Definition 4.1 (Radially L-C^1 function). Let $\hat{f} : TM \to \mathbb{R}$ be as in (11). We say that $\hat{f}$ is *radially Lipschitz continuously differentiable* if there exist reals $\beta_{RL} > 0$ and $\delta_{RL} > 0$ such that, for all $x \in M$, for all $\xi \in T_xM$ with $\|\xi\| = 1$, and for all $t < \delta_{RL}$, it holds that

$$\left| \frac{d}{d\tau} \hat{f}_x(\tau\xi)|_{\tau=t} - \frac{d}{d\tau} \hat{f}_x(\tau\xi)|_{\tau=0} \right| \leq \beta_{RL} t. \tag{12}$$

For the purpose of Algorithm 1, which is a descent algorithm, this condition needs only to be imposed for all x in the level set

$$\{x \in M : f(x) \leq f(x_0)\}. \tag{13}$$

A key assumption in the classical global convergence result in $\mathbb{R}^n$ is that the approximate solution η_k of the trust-region subproblem (7) produces at least as much decrease in the model function as a fixed fraction of the Cauchy decrease; see [**NW99**, Sec. 4.3]. Since the trust-region subproblem (7) is expressed on a Euclidean space, the definition of the Cauchy point is adapted from $\mathbb{R}^n$ without difficulty, and the bound

$$m_{x_k}(0) - m_{x_k}(\eta_k) \geq c_1 \| \operatorname{grad} f(x_k)\| \min \left(\Delta_k, \frac{\|\operatorname{grad} f(x_k)\|}{\|\mathcal{H}_{x_k}\|} \right), \tag{14}$$

for some constant $c_1 > 0$, is readily obtained from the $\mathbb{R}^n$ case, where $\|\mathcal{H}_{x_k}\|$ is defined as

$$\|\mathcal{H}_{x_k}\| := \sup\{\|\mathcal{H}_{x_k}\zeta\| : \zeta \in T_{x_k}M, \ \|\zeta\| = 1\}. \tag{15}$$

In particular, the truncated CG method (Algorithm 2) satisfies this bound (with $c_1 = \frac{1}{2}$, see [**NW99**, Lemma 4.5]) since it first computes the Cauchy point and then attempts to improve the model decrease.

With these preliminaries in place, we can state and prove the first global convergence result. (We refer to [**ABG06a**] for the proof, which differs little from the Euclidean case.) Note that this theorem is presented under weak assumptions; stronger but arguably easier to check assumptions are given in Proposition 4.5.

Theorem 4.2. *Let* $\{x_k\}$ *be a sequence of iterates generated by Algorithm* 1 *with* $\rho' \in \left[0, \frac{1}{4}\right)$. *Suppose that* f *is* C^1 *and bounded below on the level set* (13), *that* $\hat{f}$ *is radially L-*C^1 (*Definition* 4.1), *and that* $\|\mathcal{H}_{x_k}\| \leq \beta$ *for some constant* β. *Further suppose that all approximate solutions* η_k *of* (7) *satisfy the Cauchy decrease inequality* (14) *for some positive constant* c_1. *We then have*

$$\liminf_{k\to\infty} \|\operatorname{grad} f(x_k)\| = 0.$$

To further show that all accumulation points of $\{x_k\}$ are stationary points, we need to make an additional regularity assumption on the cost function f. The global convergence result in $\mathbb{R}^n$, as stated in [**NW99**, Theorem 4.8], requires that f be Lipschitz continuously differentiable. That is to say, for any $x, y \in \mathbb{R}^n$,

$$\| \operatorname{grad} f(y) - \operatorname{grad} f(x)\| \leq \beta_1 \|y - x\|. \tag{16}$$

A key to obtaining a Riemannian counterpart of this global convergence result is to adapt the notion of Lipschitz continuous differentiability to the Riemannian manifold (M, g). The expression $\|x - y\|$ in the right-hand side of (16) naturally becomes the Riemannian distance $\operatorname{dist}(x, y)$. For the left-hand side of (16), observe that the operation $\operatorname{grad} f(x) - \operatorname{grad} f(y)$ is not well defined in general on a Riemannian manifold since $\operatorname{grad} f(x)$ and $\operatorname{grad} f(y)$ belong to two different tangent spaces, namely, T_xM and T_yM. However, if y belongs to a normal neighborhood of x, then there is a unique geodesic $\alpha(t) = \operatorname{Exp}_x(t \operatorname{Exp}_x^{-1} y)$ such that $\alpha(0) = x$ and $\alpha(1) = y$, and we can parallel transport $\operatorname{grad} f(y)$ along α to obtain the vector $P_\alpha^{0\leftarrow 1} \operatorname{grad} f(y)$ in T_xM, to yield the following definition.

Definition 4.3 (Lipschitz continuous differentiability). Assume that (M, g) has an injectivity radius $i(M) > 0$. A real function f on M is *Lipschitz continuous differentiable* if it is differentiable and if, for all x, y in M such that $\mathrm{dist}(x, y) < i(M)$, it holds that

$$\|P_\alpha^{0\leftarrow 1} \operatorname{grad} f(y) - \operatorname{grad} f(x)\| \leq \beta_1 \operatorname{dist}(y, x), \tag{17}$$

where α is the unique geodesic with $\alpha(0) = x$ and $\alpha(1) = y$.

Note that (17) is symmetric in x and y; indeed, since the parallel transport is an isometry, it follows that

$$\|P_\alpha^{0\leftarrow 1} \operatorname{grad} f(y) - \operatorname{grad} f(x)\| = \|\operatorname{grad} f(y) - P_\alpha^{1\leftarrow 0} \operatorname{grad} f(x)\|.$$

Moreover, we place one additional requirement on the retraction R, that there exists some $\mu > 0$ and $\delta_\mu > 0$ such that

$$\|\xi\| \geq \mu\, d(x, R_x(\xi)) \quad \text{for all } x \in M, \ \text{for all } \xi \in T_x M, \quad \|\xi\| \leq \delta_\mu. \tag{18}$$

Note that for the exponential retraction discussed in this paper, (18) is satisfied as an equality, with $\mu = 1$. The bound is also satisfied when R is smooth and M is compact (Corollary 4.6).

We are now ready to show that under some additional assumptions, the gradient of the cost function converges to zero on the whole sequence of iterates. Here again we refer to Proposition 4.5 for a simpler (but slightly stronger) set of assumptions that yield the same result.

Theorem 4.4. *Let $\{x_k\}$ be a sequence of iterates generated by Algorithm* 1. *Suppose that all the assumptions of Theorem* 4.2 *are satisfied. Further suppose that $\rho' \in \left(0, \frac{1}{4}\right)$, that f is Lipschitz continuously differentiable* (*Definition* 4.3), *and that* (18) *is satisfied for some $\mu > 0$, $\delta_\mu > 0$. It then follows that*

$$\lim_{k\to\infty} \operatorname{grad} f(x_k) = 0.$$

Proof. We refer to [**ABG06a**] for a full proof. The general idea is similar to the Euclidean case. Given an index m such that $\operatorname{grad} f(x_m) \neq 0$, it is shown, using the Lipschitz continuous differentiability property, that there is a ball around x_m, whose radius depends on $\|\operatorname{grad} f(x_m)\|$, in which $\|\operatorname{grad} f(x)\| \geq \frac{1}{2}\|\operatorname{grad} f(x_m)\|$. A particularity of the Riemannian case is that the radius of the ball is required to be smaller than the injectivity radius $i(M)$, so that the Lipschitz continuous differentiability bound (17) applies. Because of Theorem 4.2, the iterates must eventually leave the ball at an iterate $l + 1$. It is then shown that the decrease of the cost function f between x_m and x_{l+1} is bounded below by an expression that vanishes only if $\|\operatorname{grad} f(x_m)\|$ vanishes. In the proof of this intermediate result, we have to be mindful that, in contrast to the Euclidean case, $\|\eta_k\|$ is in general different from $\operatorname{dist}(x_k, R_{x_k}(\eta_k))$; we use (18) to fall back to a suitable bound. As in the Euclidean case, the proof is concluded by observing that since f is bounded below on the level set, the decreases of f must go to zero, hence $\|\operatorname{grad} f(x_m)\|$ converges to zero. □

Note that this theorem reduces gracefully to the classical $\mathbb{R}^n$ case, taking $M = \mathbb{R}^n$ endowed with the classical inner product and $R_x(\xi) := x + \xi$. Then $i(M) = +\infty > 0$, R satisfies (18), and the Lipschitz condition (17) reduces to the classical expression, which subsumes the radially L-C^1 condition.

The following proposition (proved in [**ABG06a**]) shows that the regularity conditions on f and $\hat{f}$ required in the previous theorems are satisfied under stronger but possibly easier to check conditions. These conditions impose a bound on the Hessian of f and on the "acceleration" along curves $t \mapsto R(t\xi)$. Note also that all these conditions need only be checked on the level set $\{x \in M : f(x) \leq f(x_0)\}$.

Proposition 4.5. *Suppose that $\|\operatorname{grad} f(x)\| \leq \beta_g$ and $\|\operatorname{Hess} f(x)\| \leq \beta_H$ for some constants β_g, β_H, and all $x \in M$. Moreover, suppose that*

$$\left\| \frac{D}{dt}\frac{d}{dt} R(t\xi) \right\| \leq \beta_D \tag{19}$$

for some constant β_D, for all $\xi \in TM$ with $\|\xi\| = 1$ and all $t < \delta_D$, where D/dt denotes the covariant derivative along the curve $t \mapsto R(t\xi)$ (see [**dC92**, Chap. 2, Prop. 2.2]).

Then the Lipschitz-C^1 condition on f (Definition 4.3) *is satisfied with $\beta_L = \beta_H$; the radially Lipschitz-C^1 condition on $\hat{f}$ (Definition* 4.1) *is satisfied for $\delta_{RL} < \delta_D$ and $\beta_{RL} = \beta_H(1 + \beta_D\delta_D) + \beta_g\beta_D$; and the condition* (18) *on R is satisfied for values of μ and δ_μ satisfying $\delta_\mu < \delta_D$ and $\frac{1}{2}\beta_D\delta_\mu < 1/\mu - 1$.*

In many practical cases, the cost function and the retraction are smooth and the Riemannian manifold is compact. This is a comfortable situation, as the next result shows. (A proof is given in [**ABG06a**].)

Corollary 4.6 (Smoothness and compactness). *If the cost function f and the retraction R are smooth and the Riemannian manifold M is compact, then all the conditions in Proposition* 4.5 *are satisfied.*

4.2. *Local Convergence*

We now state the local convergence properties of Algorithm 1-2 (i.e., Algorithm 1 where the trust-region subproblem (7) is solved approximately with Algorithm 2). We first state a few preparation lemmas. We refer to [**ABG06a**] for proofs.

As before, (M, g) is a complete Riemannian manifold of dimension d, and R is a retraction on M (Definition 2.1). The first lemma is a first-order Taylor formula for tangent vector fields. (Similar Taylor developments on manifolds can be found in [**Smi94**].)

Lemma 4.7 (Taylor). *Let $x \in M$, let V be a normal neighborhood of x, and let ζ be a C^1 tangent vector field on M. Then, for all $y \in V$,*

$$P_\gamma^{0\leftarrow 1}\zeta_y = \zeta_x + \nabla_\xi \zeta + \int_0^1 (P_\gamma^{0\leftarrow\tau}\nabla_{\gamma'(\tau)}\zeta - \nabla_\xi\zeta)\, d\tau, \tag{20}$$

where γ is the unique minimizing geodesic satisfying $\gamma(0) = x$ and $\gamma(1) = y$, and $\xi = \mathrm{Exp}_x^{-1} y = \gamma'(0)$.

We use this lemma to show that in some neighborhood of a nondegenerate local minimizer v of f, the norm of the gradient of f can be taken as a measure of the Riemannian distance to v.

Lemma 4.8. *Let $v \in M$ and let f be a C^2 cost function such that* $\operatorname{grad} f(v) = 0$ *and* $\operatorname{Hess} f(v)$ *is positive definite with maximal and minimal eigenvalues $\lambda_{\max}$ and $\lambda_{\min}$. Then, given $c_0 < \lambda_{\min}$ and $c_1 > \lambda_{\max}$, there exists a neighborhood V of v such that, for all $x \in V$, it holds that*

$$c_0 \operatorname{dist}(v, x) \le \|\operatorname{grad} f(x)\| \le c_1 \operatorname{dist}(v, x). \tag{21}$$

We need a relation between the gradient of f at $R_x(\xi)$ and the gradient of $\hat{f}_x$ at ξ.

Lemma 4.9. *Let R be a retraction on M and let f be a C^1 cost function on M. Then, given $v \in M$ and $c_5 > 1$, there exists a neighborhood V of v and $\delta > 0$ such that*

$$\|\operatorname{grad} f(R(\xi))\| \le c_5 \|\operatorname{grad} \hat{f}(\xi)\|$$

for all $x \in V$ and all $\xi \in T_x M$ with $\|\xi\| \le \delta$, where $\hat{f}$ is as in (11).

Finally, we need the following result concerning the Hessian at stationary points.

Lemma 4.10. *Let R be a C^2 retraction, let f be a C^2 cost function, and let v be a stationary point of f (i.e.,* $\operatorname{grad} f(v) = 0$). *Then* $\operatorname{Hess} \hat{f}_v(0_v) = \operatorname{Hess} f(v)$.

Away from the stationary points, the Hessians $\operatorname{Hess} f(x)$ and $\operatorname{Hess} \hat{f}_x(0_x)$ do not coincide. They do coincide if a "zero acceleration" condition (22) is imposed on the retraction. This result will not be used in the convergence analysis but it can be useful in applications, as explained after (9).

Lemma 4.11. *Suppose that*

$$\frac{D}{dt}\left(\frac{d}{dt}R(t\xi)\right)\bigg|_{t=0} = 0 \qquad \textit{for all} \quad \xi \in TM, \tag{22}$$

where D/dt denotes the covariant derivative along the curve $t \mapsto R(t\xi)$ (see [**dC92**, Chap. 2, Prop. 2.2]). *Then* $\operatorname{Hess} f(x) = \operatorname{Hess} \hat{f}(0_x)$.

We now state and prove [**ABG06a**] the local convergence results. The first result states that the nondegenerate local minima are attractors of Algorithm 1-2. The principle of the argument is closely related to the Capture Theorem, see [**Ber95**, Theorem 1.2.5].

Theorem 4.12 (Local convergence to local minima). *Consider Algorithm* 1-2—*i.e., the Riemannian trust-region algorithm where the trust-region subproblems* (7) *are solved using the truncated CG algorithm with stopping criterion* (10)—*with all the assumptions of Theorem* 4.2. *Let v be a nondegenerate local minimizer of f, i.e.,* $\operatorname{grad} f(v) = 0$ *and* $\operatorname{Hess} f(v)$ *is positive definite. Assume that $x \mapsto \|\mathcal{H}_x^{-1}\|$ is bounded on a neighborhood of v and that* (18) *holds for some $\mu > 0$ and $\delta_\mu > 0$. Then there exists a neighborhood V of v such that, for all $x_0 \in V$, the sequence $\{x_k\}$ generated by Algorithm* 1-2 *converges to v.*

Now we study the order of convergence of the sequences that converge to a nondegenerate local minimizer.

Theorem 4.13 (Order of convergence). *Consider Algorithm* 1-2 *with stopping criterion* (10). *Suppose that R is a C^2 retraction, that f is a C^2 cost function on M, and that*

$$\|\mathcal{H}_{x_k} - \operatorname{Hess} \hat{f}_{x_k}(0_k)\| \leq \beta_{\mathcal{H}} \|\operatorname{grad} f(x_k)\|, \tag{23}$$

that is, $\mathcal{H}_{x_k}$ is a sufficiently good approximation of $\operatorname{Hess} \hat{f}_{x_k}(0_{x_k})$. *Let $v \in M$ be a nondegenerate local minimizer of f (i.e.,* $\operatorname{grad} f(v) = 0$ *and* $\operatorname{Hess} f(v)$ *is positive definite). Further assume that* $\operatorname{Hess} \hat{f}_x$ *is Lipschitz-continuous at 0_x uniformly in x in a neighborhood of v, i.e., there exist $\beta_{L2} > 0$, $\delta_1 > 0$, and $\delta_2 > 0$ such that, for all $x \in B_{\delta_1}(v)$ and all $\xi \in B_{\delta_2}(0_x)$, there holds*

$$\|\operatorname{Hess} \hat{f}_x(\xi) - \operatorname{Hess} \hat{f}_x(0_x)\| \leq \beta_{L2}\|\xi\|, \tag{24}$$

where $\|\cdot\|$ in the left-hand side denotes the operator norm in $T_x M$ defined as in (15).

Then there exists $c > 0$ such that, for all sequences $\{x_k\}$ generated by the algorithm converging to v, there exists $K > 0$ such that, for all $k > K$,

$$\operatorname{dist}(x_{k+1}, v) \leq c\,(\operatorname{dist}(x_k, v))^{\min\{\theta+1,2\}} \tag{25}$$

with $\theta > 0$ as in (10).

Proof. In the detailed proof [**ABG06a**], we show that there exist $\tilde{\Delta}$, c_0, c_1, c_2, c_3, c_3', c_4, c_5 such that, for all sequences $\{x_k\}$ satisfying the conditions asserted, all

$x \in M$, all ξ with $\|\xi\| < \tilde{\Delta}$, and all k greater than some K, there holds

$$\begin{aligned} c_0 \operatorname{dist}(v, x_k) &\leq \|\operatorname{grad} f(x_k)\| \leq c_1 \operatorname{dist}(v, x_k), && (26)\\ \|\eta_k\| &\leq c_4 \|\operatorname{grad} m_{x_k}(0)\| \leq \tilde{\Delta}, && (27)\\ \rho_k &> \rho', && (28)\\ \|\operatorname{grad} f(R_{x_k}(\xi))\| &\leq c_5 \|\operatorname{grad} \hat{f}_{x_k}(\xi)\|, && (29)\\ \|\operatorname{grad} m_{x_k}(\xi) - \operatorname{grad} \hat{f}_{x_k}(\xi)\| &\leq c_3 \|\xi\|^2 + c_3' \|\operatorname{grad} f(x_k)\| \, \|\xi\|, && (30)\\ \|\operatorname{grad} m_{x_k}(\eta_k)\| &\leq c_2 \|\operatorname{grad} m_{x_k}(0)\|^{\theta+1}, && (31) \end{aligned}$$

where $\{\eta_k\}$ is the sequence of update vectors corresponding to $\{x_k\}$. In particular, (26) comes from Lemma 4.8 and (29) follows from Lemma 4.9. With these results at hand the proof is concluded as follows. For all $k > K$, it follows from (26) and (28) that

$$c_0 \operatorname{dist}(v, x_{k+1}) \leq \|\operatorname{grad} f(x_{k+1})\| = \|\operatorname{grad} f(R_{x_k}(\eta_k))\|,$$

from (29) and (27) that

$$\|\operatorname{grad} f(R_{x_k}(\eta_k))\| \leq c_5 \|\operatorname{grad} \hat{f}_{x_k}(\eta_k)\|,$$

from (27) and (30) and (31) that

$$\begin{aligned} \|\operatorname{grad} \hat{f}_{x_k}(\eta_k)\| &\leq \|\operatorname{grad} m_{x_k}(\eta_k) - \operatorname{grad} \hat{f}_{x_k}(\eta_k)\| + \|\operatorname{grad} m_{x_k}(\eta_k)\|\\ &\leq (c_3 c_4^2 + c_3' c_4) \|\operatorname{grad} m_{x_k}(0)\|^2 + c_2 \|\operatorname{grad} m_{x_k}(0)\|^{1+\theta}, \end{aligned}$$

and from (26) that

$$\|\operatorname{grad} m_{x_k}(0)\| = \|\operatorname{grad} f(x_k)\| \leq c_1 \operatorname{dist}(v, x_k).$$

Consequently, taking K larger if necessary so that $\operatorname{dist}(v, x_k) < 1$ for all $k > K$, it follows that

$$\begin{aligned} c_0 \operatorname{dist}(v, x_{k+1}) &\leq \|\operatorname{grad} f(x_{k+1})\| && (32)\\ &\leq c_5 (c_3 c_4^2 + c_3' c_4) \|\operatorname{grad} f(x_k)\|^2 + c_5 c_2 \|\operatorname{grad} f(x_k)\|^{\theta+1} && (33)\\ &\leq c_5 ((c_3 c_4^2 + c_3' c_4) c_1^2 (\operatorname{dist}(v, x_k))^2 + c_2 c_1^{1+\theta} (\operatorname{dist}(v, x_k))^{1+\theta})\\ &\leq c_5 ((c_3 c_4^2 + c_3' c_4) c_1^2 + c_2 c_1^{1+\theta}) (\operatorname{dist}(v, x_k))^{\min\{2, 1+\theta\}} \end{aligned}$$

for all $k > K$, which is the desired result. □

The constants in the proof of Theorem 4.13 can be chosen as $c_0 < \lambda_{\min}$, $c_1 > \lambda_{\max}$, $c_4 > 1/\lambda_{\min}$, $c_5 > 1$, $c_3 \geq \beta_{L2}$, $c_3' \geq \beta_{\mathcal{H}}$, $c_2 \geq 1$, where $\lambda_{\min}$ and

$\lambda_{\max}$ are the smallest and largest eigenvalue of Hess $f(v)$ respectively [**ABG06a**]. Consequently, the constant c in the convergence bound (25) can be chosen as

$$c > \frac{1}{\lambda_{\min}}((\beta_{L2}/\lambda_{\min}^2 + \beta_{\mathcal{H}}/\lambda_{\min})\lambda_{\max}^2 + \lambda_{\max}^{1+\theta}). \tag{34}$$

A nicer-looking bound holds when convergence is evaluated in terms of the norm of the gradient, as expressed in the theorem below which is a direct consequence of (32)–(33).

Theorem 4.14. *Under the assumptions of Theorem* 4.13, *if* $\theta + 1 < 2$, *then given* $c_g > 1$ *and* $\{x_k\}$ *generated by the algorithm, there exists* $K > 0$ *such that*

$$\|\mathrm{grad}\, f(x_{k+1})\| \leq c_g \|\mathrm{grad}\, f(x_k)\|^{\theta+1}$$

for all $k > K$.

Nevertheless, (32)–(33) suggest that the algorithm may not perform well when the relative gap $\lambda_{\max}/\lambda_{\min}$ is large. In spite of this, numerical experiments on eigenvalue problems have shown that the method tends to behave as well, or even better than other methods in the presence of a small relative gap [**ABG06b**].

5. Applications

In this section we briefly review the essential "ingredients" necessary for applying the RTR-tCG method (Algorithm 1-2) and we present two examples in detail. These examples are presented as illustrations: comparing the resulting algorithms with existing methods and conducting numerical experiments is beyond the scope of this paper. For the problem of computing extreme eigenspaces of matrices, numerical experiments show that the RTR-tCG algorithm can match and sometimes dramatically outperform existing algorithms; experiments, comparisons, and further developments are presented in [**ABG06b**] and [**ABGS05**]. Other applications that would lend themselves nicely to an RTR approach include reduced-rank approximation to matrices, the Procrustes problem, nearest-Jordan structure, trace minimization with a nonlinear term, simultaneous Schur decomposition, and simultaneous diagonalization; see, e.g., [**HM94**], [**LE00**].

5.1. *Checklist*

The following elements are required for applying the RTR method to optimizing a cost function f on a Riemannian manifold (M, g):

(i) a tractable numerical representation for points x on M, for tangent spaces $T_x M$, and for the inner products $g_x(\cdot,\cdot)$ on $T_x M$;

(ii) choice of a retraction $R_x : T_xM \to M$ (Definition 2.1);
(iii) formulas for $f(x)$, grad $f(x)$, and the approximate Hessian $\mathcal{H}_x$ that satisfies the properties required for the convergence results in Section 4.

Choosing a good retraction amounts to finding an approximation of the exponential mapping that can be computed with low computational cost. Guidelines can be found in [**CI01**], [**DN04**]. This is an important open research topic.

Formulas for grad $f(x)$ and Hess $\hat{f}_x(0_x)$ can be obtained by identification in a Taylor expansion of the lifted cost function $\hat{f}_x$, namely,

$$\hat{f}_x(\eta) = f(x) + g_x(\operatorname{grad} f(x), \eta) + \tfrac{1}{2} g_x(\operatorname{Hess} \hat{f}_x(0_x)[\eta], \eta) + O(\|\eta\|^3),$$

where grad $f(x) \in T_xM$ and Hess $\hat{f}_x(0_x)$ is a linear transformation of T_xM. In order to obtain an "approximate Hessian" $\mathcal{H}_x$ that satisfies the approximation condition (23), one can pick $\mathcal{H}_x := \operatorname{Hess}(f \circ \tilde{R}_x)(0_x)$ where $\tilde{R}_x$ is any retraction. Then, assuming sufficient smoothness of f, R and $\tilde{R}$, the bound (23) follows from Lemmas 4.8 and 4.10. In particular, the choice $\tilde{R}_x = \operatorname{Exp}_x$ yields $\mathcal{H}_x = \nabla \operatorname{grad} f(x)$. If M is an embedded submanifold of a Euclidean space, then $\nabla_\eta \operatorname{grad} f(x) = \pi \mathrm{D} \operatorname{grad} f(x)[\eta]$ where π denotes the orthogonal projector onto T_xM.

5.2. *Symmetric Eigenvalue Decomposition*

Let M be the orthogonal group,

$$M = O_n = \{Q \in \mathbb{R}^{n\times n} : Q^TQ = I_n\}.$$

This manifold is an embedded submanifold of $\mathbb{R}^{n\times n}$. It can be shown that $T_QO_n = \{Q\Omega : \Omega = -\Omega^T\}$; see, e.g., [**HM94**]. The canonical Euclidean metric $g(A, B) = \operatorname{trace}(A^TB)$ on $\mathbb{R}^{n\times n}$ induces on O_n the metric

$$g_Q(Q\Omega_1, Q\Omega_2) = \operatorname{trace}(\Omega_1^T\Omega_2). \tag{35}$$

A retraction $R_Q : T_QO_n \to O_n$ must be chosen that satisfies the properties stated in Section 2. The Riemannian geodesic-based choice is

$$R_Q(Q\Omega) = \operatorname{Exp}_Q(Q\Omega) = Q\exp(Q^T(Q\Omega)) = Q\exp(\Omega)$$

where exp denotes the matrix exponential. However, the matrix exponential is numerically expensive to compute (the computational cost is comparable to solving an $n \times n$ eigenvalue problem!), which makes it essential to use computationally cheaper retractions. Given a Lie group G (here the orthogonal group) and its Lie algebra $\mathfrak{g}$ (here the set of skew-symmetric matrices), there exist several ways of approximating $\exp(\Omega)$, $\Omega \in \mathfrak{g}$, by an $R(\Omega)$ such that $R(\Omega) \in G$ if $B \in \mathfrak{g}$; these techniques are well known in geometric integration (see, e.g., [**CI01**] and references therein) and can be applied to our case where G is the orthogonal group O_n. For example, $\exp(\Omega)$ can be approximated by a product of plane (or Givens) rotations [**GV96**] in such a way that R is a second-order approximation of the exponential; see [**CI01**]. This approach has the advantage of being very efficient computationally.

Consider the cost function

$$f(Q) = \operatorname{trace}(Q^T A Q N)$$

where A and N are given $n \times n$ symmetric matrices. For $N = \operatorname{diag}(\mu_1, \ldots, \mu_n)$, $\mu_1 < \cdots < \mu_n$, the minimum of f is realized by the orthonormal matrices of eigenvectors of A sorted in increasing order of corresponding eigenvalue; see, e.g., [**HM94**, Sec. 2.1]. Assume that the retraction R approximates the exponential at least to order 2. With the metric g defined as in (35), we obtain

$$\begin{aligned}
\hat{f}_Q(Q\Omega) &:= f(R_Q(Q\Omega)) \\
&= \operatorname{trace}((I+\Omega+\tfrac{1}{2}\Omega^2+O(\Omega^3))^T Q^T A Q (I+\Omega+\tfrac{1}{2}\Omega^2+O(\Omega^3))N) \\
&= f(Q) + 2\operatorname{trace}(\Omega^T Q^T A Q N) \\
&\quad + \operatorname{trace}(\Omega^T Q^T A Q \Omega N - \Omega^T \Omega Q^T A Q N) + O(\Omega^3)
\end{aligned}$$

from which it follows

$$D\hat{f}_Q(0)[Q\Omega] = 2\operatorname{trace}(Q^T A Q \Omega N)$$

$$\tfrac{1}{2} D^2 \hat{f}_Q(0)[Q\Omega_1, Q\Omega_2] = \operatorname{trace}(\Omega_1^T Q^T A Q \Omega_2 N - \tfrac{1}{2}(\Omega_1^T \Omega_2 + \Omega_2^T \Omega_1) Q^T A Q N)$$

$$\operatorname{grad} \hat{f}_Q(0) = \operatorname{grad} f(Q) = Q[Q^T A Q, N]$$

$$\begin{aligned}
\operatorname{Hess} \hat{f}_Q(0)[Q\Omega] &= \operatorname{Hess} f(Q)[Q\Omega] \\
&= \tfrac{1}{2} Q[[Q^T A Q, \Omega], N] + \tfrac{1}{2} Q[[N, \Omega], Q^T A Q],
\end{aligned}$$

where $[A, B] := AB - BA$. It is now straightforward to replace these expressions in the general formulation of Algorithm 1-2 and obtain a practical matrix algorithm.

An alternative way to obtain $\operatorname{Hess} \hat{f}_Q(0)$ is to exploit Lemma 4.11 which yields $\operatorname{Hess} \hat{f}_Q(0) = \nabla \operatorname{grad} f(Q)$. Since the manifold M is an embedded Riemannian submanifold of $\mathbb{R}^{n\times p}$, the covariant derivative ∇ is obtained by projecting the derivative in $\mathbb{R}^{n\times p}$ onto the tangent space to M; see [**dC92**, Chap. 2, Sec. 1] or [**Boo75**, VII.2]. We obtain $\operatorname{Hess} f(Q)[Q\Omega] = Q \operatorname{skew}(\Omega[Q^T Q Q, N] + [\Omega^T Q^T A Q + Q^T A Q \Omega, N]$, which yields the same result as above.

5.3. *Computing an Extreme Eigenspace of a Symmetric Definite Matrix Pencil*

We assume that A and B are $n \times n$ symmetric matrices and that B is positive definite. An eigenspace $\mathcal{Y}$ of (A, B) satisfies $B^{-1}Ay \in \mathcal{Y}$ for all $y \in \mathcal{Y}$, which can also be written $B^{-1}A\mathcal{Y} \subseteq \mathcal{Y}$ or $A\mathcal{Y} \subseteq B\mathcal{Y}$. The simplest example is when $\mathcal{Y}$ is spanned by a single eigenvector of (A, B), i.e., a nonvanishing vector y such that $Ay = \lambda By$ for some eigenvalue λ. More generally, an eigenspace can be spanned by a subset of eigenvectors of (A, B). For more details we refer to the review of the generalized eigenvalue problem in [**Ste01**].

Let $\lambda_1 \leq \cdots \leq \lambda_p < \lambda_{p+1} \leq \cdots \leq \lambda_n$ be the eigenvalues of the pencil (A, B). We consider the problem of computing the (unique) eigenspace $\mathcal{V}$ of (A, B) associated to the p leftmost eigenvalues (in other words, $\mathcal{V}$ is characterized by $\mathcal{V} = \text{colsp}(V)$ where $AV = BV\,\text{diag}(\lambda_1, \ldots, \lambda_p)$ and $V^TV = I$). We will call $\mathcal{V}$ the *leftmost* p-dimensional eigenspace of the pencil (A, B). Note that the algorithms we are about to present work equally well for computing the *rightmost* eigenspace: replace A by $-A$ throughout and notice that the leftmost eigenspace of $-A$ is the rightmost eigenspace of A.

It is well known (see, e.g., [**SW82**], [**ST00**]) that the leftmost eigenspace $\mathcal{V}$ of (A, B) is the minimizer of the Rayleigh cost function

$$f(\text{colsp}(Y)) = \text{trace}((Y^TAY)(Y^TBY)^{-1}) \tag{36}$$

where Y is full-rank $n \times p$ and colsp(Y) denotes the column space of Y. It is readily checked that the right-hand side only depends on colsp(Y).

The domain M of the cost function f is the set of p-dimensional subspaces of $\mathbb{R}^n$, called the *Grassmann manifold* and denoted by Grass(p, n). A difficulty with the Grassmann manifold is that it is not directly defined as a submanifold of a Euclidean space (in contrast to the orthogonal group considered in Sec. 5.2). The first action to take is thus to devise a matrix representation of the elements of Grass(p, n) and its tangent vectors. This can be done in several ways.

A possibility is to rely on the one-to-one correspondence between subspaces and projectors; this idea is detailed in [**MS85**]. Another possibility is to rely on the definition of Grass(p, n) as a quotient of Lie groups; see [**EAS98**] and references therein. Yet another possibility is to rely on coordinate charts on the Grassmannian (see, e.g., [**HM94**, Sec. C4]); this approach is appealing because it uses a minimal set of variables, but it has the drawback of relying on arbitarily fixed reference points.

A fourth way, which we will follow here, is to consider Grass(p, n) as the quotient $\mathbb{R}_*^{n\times p}/\text{GL}_p$ of the locally Euclidean space $\mathbb{R}_*^{n\times p}$ (the set of full-rank $n \times p$ matrices) by the set of transformations that preserve the column space. This approach was developed in [**AMS04**]. The principle is to allow a subspace to be represented by any $n \times p$ matrix whose columns span the subspace; that is, the subspaces are represented by bases (which are allowed to be nonorthonormal, although in practical computations it is often desirable to require some form of orthonormalization). This representation is particularly appropriate in the scope of numerical computations. The set of matrices that represent the same subspace as a matrix $Y \in \mathbb{R}_*^{n \times p}$ is the *fiber* $Y\mathrm{GL}_p = \{YM : \det(M) \neq 0\}$. The *vertical space* at Y is $V_Y = \{YM : M \in \mathbb{R}^{p \times p}\}$. A real function h on $\mathrm{Grass}(p, n)$ is represented by its lift $h_\uparrow(Y) = h\,(\mathrm{colsp}(Y))$. To represent a tangent vector ξ to $\mathrm{Grass}(p, n)$ at a point $\mathcal{Y} = \mathrm{colsp}(Y)$, first define a *horizontal space* H_Y whose direct sum with V_Y is the whole $\mathbb{R}^{n \times p}$; then ξ is uniquely represented by its *horizontal lift* $\xi_{\uparrow Y}$ defined by the following two conditions: (i) $\xi_{\uparrow Y} \in H_Y$; and (ii) $\mathrm{D}h(\mathcal{Y})[\xi] = \mathrm{D}h_\uparrow(Y)[\xi_{\uparrow Y}]$ for all real functions h on $\mathrm{Grass}(p, n)$. Therefore, the horizontal space H_Y represents the tangent space $T_{\mathcal{Y}}\,\mathrm{Grass}(p, n)$.

In this section, with a view to simplifying the derivation of the gradient and Hessian of the Rayleigh cost function (36), we define the horizontal space as

$$H_Y = \{Z \in \mathbb{R}^{n \times p} : Y^T B Z = 0\},$$

which reduces to the definition in [**AMS04**] when B is the identity. We then define a noncanonical metric on $\mathrm{Grass}(p, n)$ as

$$g_{\mathcal{Y}}(\xi, \zeta) = \mathrm{trace}((Y^T BY)^{-1} \xi_{\uparrow Y}^T \zeta_{\uparrow Y}). \tag{37}$$

From now on, the definitions of the gradient, Hessian, and Riemannian connection will be with respect to the metric (37). We will use the retraction

$$R_{\mathcal{Y}}(\xi) = \mathrm{colsp}(Y + \xi_{\uparrow Y}) \tag{38}$$

where $\mathcal{Y} = \mathrm{colsp}(Y)$.

For the Rayleigh cost function (36), using the notation

$$P_{U,V} = I - U(V^T U)^{-1} V^T \tag{39}$$

for the projector parallel to the span of U onto the orthogonal complement of the span of V, we obtain

$$\begin{aligned}
\hat{f}_{\mathcal{Y}}(\xi) &= f(R_{\mathcal{Y}}(\xi)) \\
&= \mathrm{trace}(((Y + \xi_{\uparrow Y})^T B(Y + \xi_{\uparrow Y}))^{-1}((Y + \xi_{\uparrow Y})^T A(Y + \xi_{\uparrow Y}))) \\
&= \mathrm{trace}((Y^T BY)^{-1} Y^T AY) + 2\,\mathrm{trace}((Y^T BY)^{-1} \xi_{\uparrow Y}^T AY) \\
&\quad + \mathrm{trace}((Y^T BY)^{-1} \xi_{\uparrow Y}^T (A\xi_{\uparrow Y} - B\xi_{\uparrow Y}(Y^T BY)^{-1}(Y^T AY))) + HOT \\
&= \mathrm{trace}((Y^T BY)^{-1} Y^T AY) + 2\,\mathrm{trace}((Y^T BY)^{-1} \xi_{\uparrow Y}^T P_{BY,BY} AY) \\
&\quad + \mathrm{trace}((Y^T BY)^{-1} \xi_{\uparrow Y}^T P_{BY,BY}(A\xi_{\uparrow Y} - B\xi_{\uparrow Y}(Y^T BY)^{-1}(Y^T AY))) \\
&\quad + HOT,
\end{aligned} \tag{40}$$

where the introduction of the projectors do not modify the expression since $P_{BY,BY}\xi_{\uparrow Y} = \xi_{\uparrow Y}$. By identification, using the noncanonical metric (37), we obtain

$$(\mathrm{grad}\, f(\mathcal{Y}))_{\uparrow Y} = (\mathrm{grad}\, \hat{f}_{\mathcal{Y}}(0))_{\uparrow Y} = 2P_{BY,BY} AY \tag{41}$$

and

$$(\text{Hess}\,\hat{f}_{\mathcal{Y}}(0\mathcal{Y})[\xi])_{\uparrow Y} = 2P_{BY,BY}(A\xi_{\uparrow Y} - B\xi_{\uparrow Y}(Y^T BY)^{-1}(Y^T AY)). \qquad (42)$$

Notice that Hess $\hat{f}_{\mathcal{Y}}(0_{\mathcal{Y}})$ is symmetric with respect to the metric, as required.

We choose to take

$$\mathcal{H}_{\mathcal{Y}} = \text{Hess}\,\hat{f}_{\mathcal{Y}}(0_{\mathcal{Y}}). \qquad (43)$$

Therefore, the approximation condition (23) is trivially satisfied. The model (7) is thus

$$\begin{aligned} m_{\mathcal{Y}}(\xi) &= f(\mathcal{Y}) + g_{\mathcal{Y}}(\text{grad}\, f(\mathcal{Y}), \xi) + \tfrac{1}{2} g_{\mathcal{Y}}(\mathcal{H}_{\mathcal{Y}}\xi, \xi) \\ &= \text{trace}((Y^T BY)^{-1} Y^T AY) + 2\,\text{trace}((Y^T BY)^{-1} \xi_{\uparrow Y}^T AY) \\ &\quad + \text{trace}((Y^T BY)^{-1} \xi_{\uparrow Y}^T (A\xi_{\uparrow Y} - B\xi_{\uparrow Y}(Y^T BY)^{-1} Y^T AY)). \quad (44) \end{aligned}$$

Since the Rayleigh cost function (36) is smooth on Grass(p, n)—recall that B is positive definite—and since Grass(p, n) is compact, it follows that all the assumptions involved in the convergence analysis of the general RTR-tCG algorithm (Section 4) are satisfied. The only complication is that we do not have a closed-form expression for the distance involved in the superlinear convergence result (25). (Since the metric (37) is different from the canonical metric, the formulas given in [**AMS04**] do not apply.) But since B is fixed and positive definite, the distances induced by the noncanonical metric (37) and by the canonical metric—(37) with $B := I$—are locally equivalent, and therefore for a given sequence both distances yield the same rate of convergence.

We now have all the required information to use the RTR-tCG method (Algorithm 1-2) for minimizing the Rayleigh cost function (36) on the Grassmann manifold Grass(p, n) endowed with the noncanonical metric (37). This yields the following matrix version of the inner iteration. (We omit the horizontal lift notation for conciseness.) We use the notation

$$\overline{\mathcal{H}}_Y[Z] = P_{BY,BY}(AZ - BZ(Y^T BY)^{-1} Y^T AY). \qquad (45)$$

Note that the omission of the factor 2 in both the gradient and the Hessian does not affect the sequence $\{\eta\}$ generated by the tCG algorithm.

Algorithm 3 (tCG for (A, B)). *Given two symmetric $n \times n$ matrices A and B with B positive definite, and a B-orthonormal full-rank $n \times p$ matrix Y (i.e., $Y^T BY = I$).*

```
Set $\eta^0 = 0 \in \mathbb{R}^{n\times p}$, $r_0 = P_{BY,BY} AY$, $\delta_0 = -r_0$;
for $j = 0, 1, 2, \ldots$ until a stopping criterion is satisfied, perform the iteration:
    if trace $\left(\delta_j^T \overline{\mathcal{H}}_Y[\delta_j]\right) \leq 0$
        Compute $\tau > 0$ such that $\eta = \eta^j + \tau\delta_j$ satisfies trace $\left(\eta^T \eta\right) = \Delta$;
        return $\eta$;
    Set $\alpha_j = \text{trace}\left(r_j^T r_j\right) / \text{trace}\left(\delta_j^T \overline{\mathcal{H}}_Y[\delta_j]\right)$;
    Set $\eta^{j+1} = \eta^j + \alpha_j \delta_j$;
    if trace $\left(\left(\eta^{j+1}\right)^T \eta^{j+1}\right) \geq \Delta$
        Compute $\tau \geq 0$ such that $\eta = \eta^j + \tau\delta_j$ satisfies trace $\left(\eta^T \eta\right) = \Delta$;
        return $\eta$;
    Set $r_{j+1} = r_j + \alpha \overline{\mathcal{H}}_Y[\delta_j]$;
    Set $\beta_{j+1} = trace\left(r_{j+1}^T r_{j+1}\right) / \text{trace}\left(r_j^T r_j\right)$;
    Set $\delta_{j+1} = -r_{j+1} + \beta_{j+1}\delta_j$;
end (for).
```

According to the retraction formula (38), the returned η yields a candidate new iterate

$$Y_+ = (Y + \eta)M$$

where M is chosen such that $Y_+^T B Y_+ = I$. The candidate is accepted or rejected and the trust-region radius is updated as prescribed in the outer RTR method (Algorithm 1), where ρ is computed using m as in (44) and $\hat{f}$ as in (40).

The resulting algorithm converges to eigenspaces of (A, B)—which are the stationary points of the cost function (36)—and convergence to the leftmost eigenspace $\mathcal{V}$ is expected to occur in practice since the other eigenspaces are numerically unstable. Moreover, since $\mathcal{V}$ is a nondegenerate local minimizer (under our assumption that $\lambda_p < \lambda_{p+1}$), it follows that the rate of convergence is $\min\{\theta + 1, 2\}$, where θ is the parameter appearing in the stopping criterion (10) of the inner (tCG) iteration.

This algorithm is further developed in [**ABGS05**]. Relations with other methods are investigated in [**ABG06b**].

6. Conclusion

We have proposed a trust-region approach for optimizing a smooth function on a Riemannian manifold. The method improves on the well-known Riemannian Newton method of Smith and Udrişte in three ways. First, the exponential mapping is relaxed to general retractions with a view to reducing computational complexity. Second, a trust-region safeguard is applied for global convergence. Third, early stopping of the inner iteration (yielding inexact solutions of the trust-region subproblems) is allowed under criteria that preserve the convergence properties of the overall algorithm. Taken independently, none of these concepts is new; the novelty is their combination in a general algorithm for optimization on manifolds, aimed at numerical efficiency with reliable global behavior, and supported by a detailed convergence analysis.

Acknowledgments

The authors wish to thank A. Edelman, U. Helmke, R. Mahony, A. Sameh, R. Sepulchre, S. T. Smith, and P. Van Dooren for useful discussions. They also wish to thank the anonymous editor and referees for their careful reading of the manuscript and for many helpful comments. This work was supported by the US National Science Foundation under Grants ACI0324944 and CCR9912415, by the School of Computational Science of Florida State University through a postdoctoral fellowship, and by Microsoft Research through a Research Fellowship at Peterhouse, Cambridge. This research was initiated while the first author was a Research Fellow with the Belgian National Fund for Scientific Research (FNRS) at the University of Liège.

References

[ABG04] P.-A. Absil, C. G. Baker, and K. A. Gallivan, Trust-region methods on Riemannian manifolds with applications in numerical linear algebra, *Proceedings of the* 16*th International Symposium on Mathematical Theory of Networks and Systems* (*MTNS*2004), Leuven, Belgium, 5–9 July 2004.

[ABG06a] P.-A. Absil, C. G. Baker, and K. A. Gallivan, *Convergence Analysis of Riemannian Trust-Region Methods*, Technical Report, http://www.optimization-online.org/ DB_HTML/ 2006/06/1416.html, 2006.

[ABG06b] P.-A. Absil, C. G. Baker, and K. A. Gallivan, A truncated-CG style method for symmetric generalized eigenvalue problems, *J. Comput. Appl. Math.* **189** (2006), nos. 1–2, 274–285.

[ABGS05] P.-A. Absil, C. G. Baker, K. A. Gallivan, and A. Sameh, Adaptive model trust region methods for generalized eigenvalue problems, in *International Conference on Computational Science* (V. S. Sunderam, G. Dick van Albada, and P. M. A. Slot, et al., eds.), Lecture Notes in Computer Science, Vol. 3514, Springer-Verlag, New York, 2005, pp. 33–41.

[ADM$^+$02] R. L. Adler, J.-P. Dedieu, J. Y. Margulies, M. Martens, and M. Shub, Newton's method on Riemannian manifolds and a geometric model for the human spine, *IMA J. Numer. Anal.* **22** (2002), no. 3, 359–390.

[AMS04] P.-A. Absil, R. Mahony, and R. Sepulchre, Riemannian geometry of Grassmann manifolds with a view on algorithmic computation, *Acta Appl. Math.* **80** (2004), no. 2, 199–220.

[Ber95] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[Boo75] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, New York, 1975.

[CGT00] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, MPS/SIAM Series on Optimization, SIAM, Philadelphia, PA, and MPS, Philadelphia, PA, 2000.

[CI01] E. Celledoni and A. Iserles, Methods for the approximation of the matrix exponential in a Lie-algebraic setting, *IMA J. Numer. Anal.* **21** (2001), no. 2, 463–488.

[dC92] M. P. do Carmo, *Riemannian Geometry*, Mathematics: Theory & Applications, Birkhäuser Boston, Boston, MA, 1992; translated from the second Portuguese edition by Francis Flaherty.

[DN04] J.-P. Dedieu and D. Novitsky, Symplectic methods for the approximation of the exponential and the Newton sequence on Riemannian submanifolds, *J. Complexity*, 2005, pp. 487–501.

[DPM03] J.-P. Dedieu, P. Priouret, and G. Malajovich, Newton's method on Riemannian manifolds: Convariant alpha theory, *IMA J. Numer. Anal.* **23** (2003), no. 3, 395–419.

[EAS98] A. Edelman, T. A. Arias, and S. T. Smith, The geometry of algorithms with orthogonality constraints, *SIAM J. Matrix Anal. Appl.* **20** (1998), no. 2, 303–353.

[Gab82] D. Gabay, Minimizing a differentiable function over a differential manifold, *J. Optim. Theory Appl.* **37** (1982), no. 2, 177–219.

[GLRT99] N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint, Solving the trust-region subproblem using the Lanczos method, *SIAM J. Optim.* **9** (1999), no. 2, 504–525.

[GV96] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 1996.

[Hag01] W. W. Hager, Minimizing a quadratic over a sphere, *SIAM J. Optim.* **12** (2001), no. 1, 188–208.

[HHM02] U. Helmke, K. Hüper, and J. B. Moore, Quadratically convergent algorithms for optimal dexterous hand grasping, *IEEE Trans. Robotics Automat.* **18** (2002), no. 2, 138–146.

[HM94] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*, Springer-Verlag, New York, 1994.

[HP05] W. W. Hager and S. Park, Global convergence of SSM for minimizing a quadratic over a sphere, *Math. Comp.* **74** (2005), no. 251, 1413–1423.

[HT04] K. Hüper and J. Trumpf, Newton-like methods for numerical optimization on manifolds, *Proc.* 38*th IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, November 7–10, 2004.

[HV01] M. Heinkenschloss and L. N. Vicente, Analysis of inexact trust-region SQP algorithms, *SIAM J. Optim.* **12** (2001), 283–302.

[LE00] R. Lippert and A. Edelman, Nonlinear eigenvalue problems with orthogonality constraints (Section 9.4), in *Templates for the Solution of Algebraic Eigenvalue Problems* (Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds.), SIAM, Philadelphia, PA, 2000, pp. 290–314.

[LT01] C. T. Lawrence and A. L. Tits, A computationally efficient feasible sequential quadratic programming algorithm, *SIAM J. Optim.* **11** (2001), no. 4, 1092–1118.

[Lue72] D. G. Luenberger, The gradient projection method along geodesics, *Management Sci.* **18** (1972), 620–631.

[Man02] J. H. Manton, Optimization algorithms exploiting unitary constraints, *IEEE Trans. Signal Process.* **50** (2002), no. 3, 635–650.

[MKS01] Y. Ma, J. Kosecka, and S. S. Sastry, Optimization criteria and geometric algorithms for motion and structure estimation, *Internat. J. Comput. Vision* **44** (2001), no. 3, 219–249.

[MM02] R. Mahony and J. H. Manton, The geometry of the Newton method on non-compact Lie groups, *J. Global Optim.* **23** (2002), no. 3, 309–327.

[MS84] J. J. Moré and D. C. Sorensen, *Newton's Method*, in *Studies in Numerical Analysis*, MAA Stud. Math., Vol. 24, Math. Assoc. America, Washington, DC, 1984, pp. 29–82.

[MS85] A. Machado and I. Salavessa, Grassmannian manifolds as subsets of Euclidean spaces, *Res. Notes Math.* **131** (1985), 85–102.

[NA05] Y. Nishimori and S. Akaho, Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold, *Neurocomputing* **67** (2005), 106–135.

[Nas56] J. Nash, The imbedding problem for Riemannian manifolds, *Ann. of Math.* (2) **63** (1956), 20–63.

[NW99] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999.

[O'N83] B. O'Neill, *Semi-Riemannian Geometry*, Pure and Applied Mathematics, Vol. 103, Academic Press, [Harcourt Brace Jovanovich Publishers], New York, 1983.

[OW00] B. Owren and B. Welfert, The Newton iteration on Lie groups, *BIT* **40** (2000), no. 1, 121–145.

[Pow74] M. J. D. Powell, Convergence properties of a class of minimization algorithms, in *Nonlinear Programming*, Vol. 2 (*Proc. Sympos. Special Interest Group on Math. Programming*, University of Wisconsin, Madison, WI, 1974), Academic Press, New York, 1974, pp. 1–27.

[Rid06] D. Ridzal, *Trust-Region SQP Methods with Inexact Linear System Solves for Large-Scale Optimization*, PhD thesis, Rice University, Houston, TX, April 2006.

[Shu86] M. Shub, Some remarks on dynamical systems and numerical analysis, *Proc. VII ELAM* (L. Lara-Carrero and J. Lewowicz, eds.), Equinoccio, U. Simón Bolívar, Caracas, 1986, pp. 69–92.

[Smi93] S. T. Smith, *Geometric Optimization Methods for Adaptive Filtering*, PhD thesis, Division of Applied Sciences, Harvard University, Cambridge, MA, 1993.

[Smi94] S. T. Smith, Optimization techniques on Riemannian manifolds, in *Hamiltonian and Gradient Flows*, *Algorithms and Control*, Fields Inst. Commun., Vol. 3, Amer. Math. Soc., Providence, RI, 1994, pp. 113–136.

[Smi05] S. T. Smith, Covariance, subspace, and intrinsic Cramér–Rao bounds, *IEEE Trans*. *Signal Process*. **53** (2005), no. 5, 1610–1630.

[Sor82] D. C. Sorensen, Newton's method with a model trust region modification, *SIAM J. Numer. Anal*. **19** (1982), no. 2, 409–426.

[ST00] A. Sameh and Z. Tong, The trace minimization method for the symmetric generalized eigenvalue problem, *J. Comput*. *Appl*. *Math*. **123** (2000), 155–175.

[Ste83] T. Steihaug, The conjugate gradient method and trust regions in large scale optimization, *SIAM J. Numer. Anal*. **20** (1983), 626–637.

[Ste01] G. W. Stewart, *Matrix Algorithms*, Vol. II: *Eigensystems*, SIAM, Philadelphia, PA, 2001.

[SW82] A. H. Sameh and J. A. Wisniewski, A trace minimization algorithm for the generalized eigenvalue problem, *SIAM J. Numer. Anal*. **19** (1982), no. 6, 1243–1259.

[Toi81] Ph. L. Toint, Towards an efficient sparsity exploiting Newton method for minimization, in *Sparse Matrices and Their Uses* (I. S. Duff, ed.), Academic Press, London, 1981, pp. 57–88.

[Udr94] C. Udrişte, *Convex Functions and Optimization Methods on Riemannian Manifolds*, Kluwer Academic, Dordrecht, 1994.

[WD05] J. M. B. Walmag and É. J. M. Delhez, A note on trust-region radius update, *SIAM J. Optim*. **16** (2005), no. 2, 548–562.

[XB05] J. Xavier and V. Barroso, Intrinsic variance lower bound (IVLB): An extension of the Cramer–Rao bound to Riemannian manifolds, *Proc*. 2005 *IEEE International Conference on Acoustics*, *Speech*, *and Signal Processing* (ICASSP 2005), Philadelphia, PA, 2005.

[Yan06] Y. Yang, Globally convergent optimization algorithms on Riemannian manifolds: Uniform framework for unconstrained and constrained optimization, *J. Optim*. *Theory Appl*. 2006 (to appear).

[YL99] W.-Y. Yan and J. Lam, An approximate approach to $H2$ optimal model reduction, *IEEE Trans*. *Automat*. *Control* **44** (1999), no. 7, 1341–1358.