Trust-region methods on Riemannian manifolds with applications in numerical linear algebra

P.-A. Absil*† C. G. Baker* K. A. Gallivan*

Abstract

A general scheme for trust-region methods on Riemannian manifolds is proposed. A truncated conjugate-gradient method is utilized to solve the trust-region subproblems. The method is illustrated several problems of numerical linear algebra.

Key words. Numerical optimization on manifolds, trust-region, truncated conjugate-gradient, Steihaug-Toint, global convergence, local convergence, symmetric eigenvalue problem, singular value decomposition.

1 Introduction

Several problems related to numerical linear algebra can be expressed as optimizing a smooth function on a differentiable manifold. Domains of application include model reduction, principal component analysis, electronic structure computation and signal processing; see e.g. [LE00] and [HM94] for details. Early algorithms for solving optimization problems on manifolds were based on steepest descent; see e.g. [HM94] and references therein. These algorithms have good global convergence properties but slow (linear) local convergence.

In $\mathbb{R}^n$, it is well known that higher rates of convergence can be achieved by using second-order information on the cost function. The classical choice is Newton's method; it plays a central role in the development of numerical techniques for optimization, because of its simple formulation and its quadratic convergence properties. The history of Newton's method on manifolds can be traced back to Gabay [Gab82] who proposed a formulation for the method on embedded submanifolds of $\mathbb{R}^n$. Smith [Smi93, Smi94] proposed a formulation of Newton's method on Riemannian manifolds; see also the related work by Udrişte [Udr94], Owren and Welfert [OW00], Mahony [Mah96], and Mahony and Manton [MM02]. However, the pure Newton method converges only locally, and it cannot distinguish between local minima, local maxima and saddle points.

In classical optimization, several techniques exist to improve the global convergence properties of Newton's method. Most of these techniques fall into two categories: line-search methods and trust-region methods; see e.g. [MS84, NW99]. Line-search techniques have been considered on Riemannian manifolds by Udrişte [Udr94] and Yang [Yan99]. However, to our knowledge, there is no mention of Riemannian trust-region methods in the literature. An objective of this paper

*School of Computational Science and Information Technology, Florida State University, Tallahassee FL 32306-4120, USA (http://www.csit.fsu.edu/{∼absil,∼cbaker,∼gallivan}).

†This author's work was supported by the National Science Foundation of the USA under Grant ACI0324944 and by the School of Computational Science of Florida State University through a postdoctoral fellowship. This research was initiated while the author was a Research Fellow with the Belgian National Fund for Scientific Research at the University of Liège.

is to fill this gap and to provide a theoretical and algorithmic framework applicable to multiple problems.

The trust-region approach we propose works along the following lines. First, a *retraction* R 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. 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 an 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 $\hat{f}_x$ 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.

The advantages of considering a trust-region method instead of 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, whereas the pure Newton method may cycle without approaching a set of stationary points. Moreover, the cost function is nonincreasing at each iterate which favors convergence to a local minimum, while the pure Newton method does not discriminate between local minima, local maxima and saddle points. 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 good convergence properties of the exact scheme.

Another interesting feature of our trust-region scheme is the use of retractions. As in most other optimization algorithms on Riemannian manifolds, our trust-region scheme first computes an update vector in the form of a tangent vector to the manifold at the current iterate. The classical technique (see [Smi94, Udr94, EAS98, Yan99]) then uses the Riemannian exponential mapping to select the next iterate from the update vector. However, as pointed out by Manton [Man02, Section IX], the exponential may not be the most appropriate or computationally efficient way of performing the update. Therefore, we allow the exponential to be replaced by any retraction. Our convergence analysis shows that, under reasonable conditions, the good properties of the algorithms are preserved.

The goal of this paper is to succinctly present a general trust-region scheme on Riemannian manifolds and to state current convergence results. The theory and algorithms can be adapted to exploit the properties of specific manifolds and problems in several disciplines. More details are in an expanded version of this paper [ABG04b].

Numerical linear algebra considers several problems that can be analyzed and solved using this approach. A particularly interesting application concerns the computation of the rightmost eigenvalue of a symmetric matrix and its associated eigenvector, in which case the manifold under consideration is the projective space and the cost function can be chosen as a Rayleigh quotient. The resulting trust-region algorithm can be interpreted as an inexact Rayleigh quotient iteration and is related to the restarted Lanczos method; we refer to the recent paper [ABG04a] for details.

This paper makes use of basic notions of Riemannian geometry and numerical optimization; all the necessary background can be found at an introductory level in [dC92] and [NW99]. The general theory of trust-region methods on Riemannian manifolds is presented in Section 2. Methods for (approximately) solving the TR subproblems are considered in Section 3. Convergence properties are investigated in Section 4. The theory is illustrated on practical examples in Section 5. Numerical experiments are reported on in Section 6. Conclusions are presented in Section 7.

2 General theory

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 $\phi_\alpha : \Omega_\alpha \subset M \to \mathbb{R}^n$; see e.g. [dC92] for details. Let f be a cost function on M and consider the problem of defining a trust-region method for 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 $m \circ \phi_\alpha$ and a different the trust region $\{y \in M : \|\phi_\alpha(y)\| \leq \Delta\}$, hence a different next iterate x_+.

A way to overcome this difficulty is to associate to each $x \in M$ a single coordinate chart. In fact, it is sufficient to define around each $x \in M$ a diffeomorphism with a Euclidean space; a coordinate chart can then be obtained by choosing an orthonormal basis of the Euclidean space. In what follows, M will be a (C^∞) Riemannian manifold, i.e., M is endowed with a correspondence, called 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 (see [dC92, Chap. 1] for details). 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 $\dot\gamma(0) = \xi$. We will also assume that M is complete, which guarantees that $\mathrm{Exp}_x\xi$ exists for all $x \in M$ and all $\xi \in T_xM$. We refer to [dC92] or [Boo75] for details.

The inverse of the exponential map Exp_x is a natural candidate for the above-mentioned diffeomorphism since Exp_x is a diffeomorphism between a neighborhood of the zero element 0_x in the Euclidean space T_xM and a neighborhood of x in M (see [dC92, Chap. 3, Proposition 2.9]). From a numerical point of view, however, the exponential may not be the best choice as it may be expensive to compute. Therefore, it is interesting to consider approximations of the exponential. Such approximations are required to satisfy at least the properties of a retraction, a concept that we borrow from [ADM$^+$02] with some modifications.

Definition 2.1 (retraction) *A* retraction *on a manifold M is a mapping $R : TM \to M$ with the following properties. Let R_x denote the restriction of R to T_xM.*

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

If moreover $\mathrm{D}^2(\mathrm{Exp}_x^{-1} \circ R_x)(0_x)$ vanishes, then R is a second-order retraction.

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). Several practical examples of retractions on Riemannian manifolds, that may be more tractable computationally than the exponential, are given in Section 5.

Our definition of a trust-region algorithm on the Riemannian manifold (M, g) with retraction R, is based on the following principles. Given a cost function $f : M \to \mathbb{R}$ and a current iterate

$x_k \in M$, we use $R_{x_k}^{-1}$ to map the local minimization problem for f on M into a minimization problem for

$$\hat{f}_{x_k} : T_{x_k}M \to \mathbb{R} : \xi \mapsto f(R_{x_k}\xi) \tag{1}$$

on the tangent space $T_{x_k}M$. The tangent space is a Euclidean space endowed with the inner product $g_{x_k}(\cdot,\cdot)$, which makes it possible to adapt classical techniques in order to solve (approximately) the trust-region subproblem for the function $\hat{f}$, namely

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

Note that, since $\mathrm{D}R_x(0_x) = \mathrm{id}_{T_xM}$, it follows that $\operatorname{grad}\hat{f}_{x_k}(0_x) = \operatorname{grad} f(x)$, 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]). Moreover, if R is a second-order retraction, then $\operatorname{Hess}\hat{f}_{x_k}(0_x) = \operatorname{Hess} f(x)$, where $\operatorname{Hess} f(x) : T_xM \to T_xM$, the Hessian (linear) operator, is defined by $\operatorname{Hess} f(x)\xi = \nabla_\xi \operatorname{grad} f(x)$, $\xi \in T_xM$ (see [dC92, Chap. 6, Ex. 11]). The Hessian operator is related to the second tensorial derivative $\mathrm{D}^2 f(x)$ by $\mathrm{D}^2 f(\xi,\chi) = \nabla_\chi\nabla_\xi f - \nabla_{\nabla_\chi\xi} f = g_x(\operatorname{Hess} f(x)\xi,\chi)$; see [Lan99, Chap. XIII, Theorem 1.1].

For the global convergence results it is only required that R be a (first-order) retraction and that the second-order term in the model be some symmetric form. Therefore, instead of (2), we consider the following more general formulation

$$\min_{\eta\in T_{x_k}M} m_{x_k}(\eta) = f(x_k) + g_{x_k}(\operatorname{grad} f(x_k),\eta) + \frac{1}{2}g_{x_k}(\mathcal{H}_{x_k}\eta,\eta) \quad \text{s.t. } g_{x_k}(\eta,\eta) \le \Delta_k^2, \tag{3}$$

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

Next, an (approximate) solution η_k of the trust-region subproblem (3) is computed, for example using a truncated conjugate-gradient method; several other possibilities are mentionned in [CGT00, Chap. 7]. The method used for computing η_k is called the *inner iteration*. The candidate for the new iterate is then given by

$$x_+ = R_{x_k}(\eta_k). \tag{4}$$

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

$$\rho_k = \frac{f(x_k) - f(R_{x_k}(\eta_k))}{m_{x_k}(0_{x_k}) - m_{x_k}(\eta_k)} = \frac{\hat{f}_{x_k}(0_{x_k}) - \hat{f}_{x_k}(\eta_k)}{m_{x_k}(0_{x_k}) - m_{x_k}(\eta_k)}. \tag{5}$$

If ρ_k is exceedingly small, then the model is very bad: 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 a 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 (see e.g. [NW99] for the classical case where M is $\mathbb{R}^n$ with its canonical metric).

Algorithm 1 (Riemann-TR – RTR) *Data: Complete Riemannian manifold (M,g); smooth scalar field 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 $k = 0, 1, 2, \dots$
    Obtain $\eta_k$ by (approximately) solving (3);
    Evaluate $\rho_k$ from (5);
    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).
```

This algorithm admits several variations and extensions; see e.g. [CGT00, Chap. 10].

3 Computing a trust-region step

The use of a retraction has made it possible to express the trust-region subproblem in the Euclidean space T_xM. Therefore, all the classical methods for solving the trust-region subproblem can be applied. The following truncated conjugate-gradient algorithm is particularly appropriate for solving the trust-region subproblem (3) when the dimension d of the manifold M is very large. Its original version in $\mathbb{R}^d$ was proposed independently by Steihaug [Ste83] and Toint [Toi81] and is therefore sometimes referred to as the Steihaug-Toint algorithm; see e.g. [CGT00, Algorithm 7.5.1]. The algorithm can be adapted as follows to the trust-region subproblem (3). 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 (truncated CG for the TR subproblem – tCG) *Set $\eta_0 = 0$, $r_0 = \operatorname{grad} f(x_k)$, $\delta_0 = -r_0$;*

```
for $j = 0, 1, 2, \dots$ 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 $m(\eta)$ in (3)
            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 $r_{j+1} = r_j + \alpha\mathcal{H}_{x_k}\delta_j$;
    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 is to truncate after a fixed number of iteration. In order to improve the convergence rate, a possibility is to stop as soon as an iteration j is reached for which

$$\|r_j\| \leq \max(\|r_0\| \min(\|r_0\|^\theta, \kappa)). \tag{6}$$

4 Convergence analysis

We consider next the global and local convergence properties of the Riemannian trust-region method (Algorithm 1). Concerning global convergence, we consider Algorithm 1 without any assumption on the way the trust-region subproblems (3) are solved, except that the approximate solution η_k must produce a decrease of the model that is at least a fixed fraction of the so-called Cauchy decrease, and we prove under some additional mild assumptions that the sequences $\{x_k\}$ converge to the set of stationary points of the cost function. For local convergence, we assume that the trust-region subproblems are solved using Algorithm 2 with stopping criterion (6) and show that the iterates converge to nondegenerate stationary points with a rate of convergence at least $\min\{\theta, 2\}$.

In this conference paper, we only give a succinct overview of the results. They are presented in detail in [ABG04b].

4.1 Global convergence

The global convergence result in $\mathbb{R}^n$, as stated in [NW99, Theorem 4.8] requires that the cost function f be Lipschitz continuously differentiable. That is to say, for any $x, y \in \mathbb{R}^n$,

$$\|\mathrm{grad} f(y) - \mathrm{grad} f(x)\| \leq \beta_1 \|y - x\|. \tag{7}$$

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 (7) naturally becomes the Riemannian distance $\mathrm{dist}(x, y)$. For the left-hand side of (7), observe that the substraction $\mathrm{grad} f(x) - \mathrm{grad} f(y)$ is not well-defined in general on a Riemannian manifold since $\mathrm{grad}\, f(x)$ and $\mathrm{grad}\, f(y)$ 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 $\alpha(t) = \mathrm{Exp}_x(t\mathrm{Exp}_x^{-1} y)$ such that $\alpha(0) = x$ and $\alpha(1) = y$, and we can parallel transport $\mathrm{grad}\, f(y)$ along α to obtain the vector $P_\alpha^{0\leftarrow 1}\mathrm{grad}\, f(y)$ in T_xM, to yield the following definition.

Definition 4.1 (Lipschitz continuous differentiability) *Assume that (M, g) has an injectivity radius $i(M) > 0$. Then a real function f on M is* Lipschitz continuous differentiable *if it is differentiable and for all x, y in M such that* $\mathrm{dist}(x, y) < i(M)$,

$$\|P_\alpha^{0\leftarrow 1}\mathrm{grad}\, f(y) - \mathrm{grad}\, f(x)\| \leq \beta_1 \mathrm{dist}(y, x), \tag{8}$$

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

Note that (8) is symmetric in x and y; indeed, since the paralel transport is an isometry, it follows that

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

For the purpose of Algorithm 1, which is a descent algorithm, condition (8) need only to be imposed for all x, y in the level set

$$\{x \in M : f(x) \leq f(x_0)\}. \tag{9}$$

Another important assumption in the global convergence result in $\mathbb{R}^n$ is that the approximate solution η_k of the trust-region subproblem (3) produces at least as much decrease in the model function as a fixed fraction of the Cauchy decrease; see [NW99, Section 4.3]. Since the trust-region subproblem (3) is expressed on a Euclidean space, the definition of the Cauchy point is adapted from $\mathbb{R}^n$ without difficulty, and the bound

$$m_k(0) - m_k(\eta_k) \geq c_1 \|\mathrm{grad} f(x_k)\| \min\left(\Delta_k, \frac{\|\mathrm{grad} f(x_k)\|}{\|\mathcal{H}_k\|}\right), \tag{10}$$

for some constant $c_1 > 0$, is readily obtained from the $\mathbb{R}^n$ case, where $\|\mathcal{H}_k\| := \sup\{\|\mathcal{H}_k\xi\| : \xi \in T_{x_k}M,\ \|\xi\| = 1\}$. Moreover, we allow the approximate solution of (3) to exceed the trust-region radius by some constant multiple,

$$\|\eta_k\| \leq \gamma\Delta_k, \quad \text{for some constant } \gamma \geq 1. \tag{11}$$

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

$$\|\xi\| \geq \mu\, d(x, R_x\xi), \quad \forall x \in M, \forall \xi \in T_xM. \tag{12}$$

Note that for the exponential retraction discussed in this paper, (12) is satisfied as an equality, with $\mu = 1$.

With these things in place, we can state and prove the global convergence results. The first result shows that $\mathrm{grad}\, f$ converges to zero on a subsequence of iterates.

Theorem 4.2 *Let $\{x_k\}$ be a sequence of iterates generated by Algorithm 1 with $\rho' \in [0, \frac{1}{4})$. Suppose that f is bounded below on the level set* (9) *and that there exist constants $\beta > 0$ and $\delta > 0$ such that, for all $x \in M$, all $\xi \in T_xM$, $\|\xi\| = 1$, all $t < \delta$, and all k, it holds $|\frac{d^2}{dt^2}\hat{f}_{x_k}(t\xi)| \leq \beta$ and $\|\mathcal{H}_k\| \leq \beta$. Further suppose that all approximate solutions η_k of* (3) *satisfy the inequalities* (10) *and* (11)*, for some positive constants c_1 and γ. We then have*

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

The next result shows that $\mathrm{grad}\, f$ goes to zero on the whole sequence of iterates if some additional assumptions are satisfied.

Theorem 4.3 *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 $\rho' \in (0, \frac{1}{4})$ and that f is Lipschitz continuously differentiable (Definition 4.1). It then follows that*

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

4.2 Local convergence

We first show that the nondegenerate local minima are attractors of Algorithm 1-2 (i.e., Algorithm 1 where the trust-region subproblem (3) is solved with Algorithm 2). The principle of the argument is closely related to the Capture Theorem, see [Ber95, Theorem 1.2.5].

Theorem 4.4 (convergence to local minima) *Consider Algorithm 1-2 with all the assumptions of Theorem 4.2. Let v be a nondegenerate local minimum of f. Then there exist $\delta > 0$ such that, for all $x_0 \in B_\delta(v)$ and all $\Delta_0 \in (0, \bar{\Delta})$, the sequence $\{x_k\}$ generated by Algorithm 1-2 converges to v.*

Now we study the rate of convergence of the sequences that converge to a nondegenrate local minimum.

Theorem 4.5 (rate of convergence) *Consider Algorithm 1-2. Suppose that R is a second-order C^2 retraction (Definition 2.1); that f is a C^2 cost function on M; that $\mathcal{H}_k = \operatorname{Hess} f(x_k)$. Let $v \in M$ be a nondegenerate local minimum of f, (i.e., $\operatorname{grad} f(v) = 0$ and $\operatorname{Hess} f(v)$ is positive definite). Further assume that $\operatorname{Hess} \hat{f}_{x_k}$ is Lipschitz-continuous at 0_x uniformly in x in a neighborhood of v; that is, there exist $\beta_1 > 0$, $\delta_1 > 0$ and $\delta_2 > 0$ such that, for all $x \in B_{\delta_1}(v)$ and all $\xi \in B_{\delta_2}(0_x)$, it holds*

$$\|\operatorname{Hess} \hat{f}_{x_k}(\xi) - \operatorname{Hess} \hat{f}_{x_k}(0_x)\| \leq \beta_1 \|\xi\|. \tag{13}$$

Then there exists $c > 0$ such that, for all sequences $\{x_k\}$ generated by Algorithm 1-2 converging to v, there exists $K > 0$ such that for all $k > K$,

$$\operatorname{dist}(x_{k+1}, v) \leq c\,(\operatorname{dist}(x_k, v))^{\min\{\theta+1,2\}} \tag{14}$$

with $\theta > 0$ as in (6).

5 Practical examples

In this section we briefly illustrate how Algorithm 1-2 applies to various practical cases.

5.1 Symmetric eigenvalue decomposition

Let M be the orthogonal group,

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

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

$$g_Q(Q\Omega_1, Q\Omega_2) = \operatorname{trace}(\Omega_1^T \Omega_2). \tag{15}$$

We must choose a retraction $R_Q : T_Q O_n \to O_n$ satisfying the properties stated in Section 2. The Riemannian geodesic-based choice is

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

where exp denotes the matrix exponential. However, the matrix exponential is numerically very 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 exists 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.

For the sake of illustration, 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 $N = \operatorname{diag}(\mu_1, \ldots, \mu_n)$, $\mu_1 < \ldots < \mu_n$, 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, Section 2.1].

Assume that a retraction R is chosen that approximates the exponential at least to order 2. With the metric g defined as in (15), we obtain

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

from which it follows

$$\begin{aligned}\mathrm{D}\hat{f}_Q(0)[Q\Omega] &= 2\mathrm{trace}(Q^T A Q \Omega N) \\ \frac{1}{2}\mathrm{D}^2\hat{f}_Q(0)[Q\Omega_1, Q\Omega_2] &= \mathrm{trace}(\Omega_1^T Q^T A Q \Omega_2 N - \frac{1}{2}(\Omega_1^T\Omega_2 + \Omega_2^T\Omega_1)Q^T A Q N) \\ \mathrm{grad}\,\hat{f}_Q(0) &= \mathrm{grad}\, f(Q) = Q[Q^T A Q, N] \\ \mathrm{Hess}\,\hat{f}_Q(0)[Q\Omega] &= \mathrm{Hess}\, f(Q)[Q\Omega] = \frac{1}{2}Q[[Q^T A Q, \Omega], N] + \frac{1}{2}Q[[N,\Omega], Q^T A Q]\end{aligned}$$

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. Numerical results are presented in Section 6.

5.2 Singular value decomposition

Let $A \in \mathbb{R}^{n\times p}$, $n > p$. Let

$$M = O_n \times O_p = \{(U,V) : U \in O_n, V \in O_p\}$$

endowed with the canonical product metric

$$g_{(U,V)}((U\Omega_{U1}, V\Omega_{V1}), (U\Omega_{U2}, V\Omega_{V2})) = \mathrm{trace}(\Omega_{U1}^T\Omega_{U2} + \Omega_{V1}^T\Omega_{V2}).$$

Consider the cost function

$$f(U,V) = \mathrm{trace}(U^T A V N)$$

on $O_n \times O_p$, where $N = [\mathrm{diag}(\mu_1,\ldots,\mu_p)|0_{p\times(n-p)}]$, $\mu_1 < \ldots < \mu_p < 0$. The minima of f correspond to ordered left and right singular vectors of A; see [HM94, Section 3.2] for details. Assume that a retraction R is chosen such that $R_{(U,V)}(U\Omega_U, V\Omega_V) = (U\exp\Omega_U, V\exp\Omega_V) + O((U,V)^3)$. Then we obtain, using the notation $\mathrm{skew}(B) = (B - B^T)/2$, we obtain

$$\mathrm{grad}\,\hat{f}_{(U,V)}(0,0) = \mathrm{grad}\, f(U,V) = (U\mathrm{skew}(U^T A V N), -V\mathrm{skew}(N U^T A V)),$$

and

$$\begin{aligned}&\mathrm{Hess}\hat{f}_{(U,V)}(0,0)[(U\Omega_U, V\Omega_V)] = \mathrm{Hess}\, f(U,V)[(U\Omega_U, V\Omega_V)] \\ &\quad = \big(\ U(\mathrm{skew}(\Omega_U\mathrm{skew}(U^T A V N)) + \mathrm{skew}(\Omega_U^T U^T A V N) + \mathrm{skew}(U^T A V \Omega_V N))\ , \\ &\qquad\qquad -V(\mathrm{skew}(\Omega_V\mathrm{skew}(N U^T A V) + \mathrm{skew}(N\Omega_U^T U^T A V) + \mathrm{skew}(N U^T A V \Omega_V))\ \big)\,.\end{aligned}$$

5.3 Computing the dominant eigenpairs of a symmetric matrix

Let A be a symmetric (not necessarily positive definite) $n \times n$ matrix with eigenvalues $\lambda_1 \leq \ldots \leq \lambda_p < \lambda_{p+1} \leq \ldots \leq \lambda_n$. Consider the problem of computing the invariant subspace $\mathcal{V}$ of A associated to the p leftmost eigenvalues (in other words, $\mathcal{V} = \text{span}(V)$, where $AV = V\text{diag}(\lambda_1, \ldots, \lambda_p)$ and $V^TV = I$). It is well known that $\mathcal{V}$ is the minimizer of the Rayleigh cost function

$$f(\text{span}(Y)) = \text{trace}((Y^TAY)(Y^TY)^{-1}) \tag{16}$$

where Y is full-rank $n \times p$. Alternatively, it is possible to compute the dominant eigenspace by maximizing f (i.e., minimizing $-f$). Here, the manifold M is the set of p-dimensional subspaces of $\mathbb{R}^n$, called Grassmann manifold. We refer to [AMS04] for details about the Riemannian structure of the Grassmann manifold, including formulas for gradients and Hessians. The Riemannian trust-region approach yields an algorithm that is closely connected to the (inexact) Rayleigh quotient iteration and to restarted Lanczos methods. Moreover, it follows from the convergence analysis of the Riemannian trust-region scheme that the method has excellent global and local convergence properties. Preliminary numerical tests suggest that the trust-region algorithm can match and even outperform restarted Lanczos. The simpler case where $p = 1$ is presented in [ABG04a].

5.4 Other examples

The Riemannian trust-region algorithm can be applied in general to minimize smooth functions on smooth manifolds where a retraction, the gradient and the Hessian have tractable formulations. Other applications include constrained least squares [HM94, Section 1.6], approximation by lower rank matrices [HM94, Section 5.1], output feedback control [HM94, Section 5.3], sensitivity optimization [HM94, Chapter 9], and also (see Lippert and Edelman [LE00]) the Procrustes problem, nearest-Jordan structure, trace minimization with a nonlinear term, simultaneous Schur decomposition, and simultaneous diagonalization.

6 Numerical Experiments

We performed numerical experiments using a Matlab implementation of the SVD algorithm (Section 5.2). The matrix A used was a 100×40 matrix with elements randomly selected from a uniform distribution. The left and right bases U and V were initialized by generating random matrices of the proper order (100×100 and 40×40, respectively) and orthogonalized using the Matlab QR decomposition. Convergence to a solution was observed on each of the 1000 numerical experiments conducted.

Figure 1 shows the error in the computed singular values at each iteration of the algorithm applied to one of the randomly chosen examples. As Algorithm 1 only produces the left and right singular vectors, the singular values had to be recovered from the matrix A. This was by producing the matrix $\hat{\Sigma} = U^TAV$. The error was measured by computing the Frobenius norm of the difference between $\hat{\Sigma}$ and the diagonal matrix of ordered singular values Σ produced by the Matlab SVD. The numerical results clearly point to a superlinear rate of convergence.

Similar numerical experiments were performed on the EVD algorithm (Section 5.1) and comparable results were obtained.

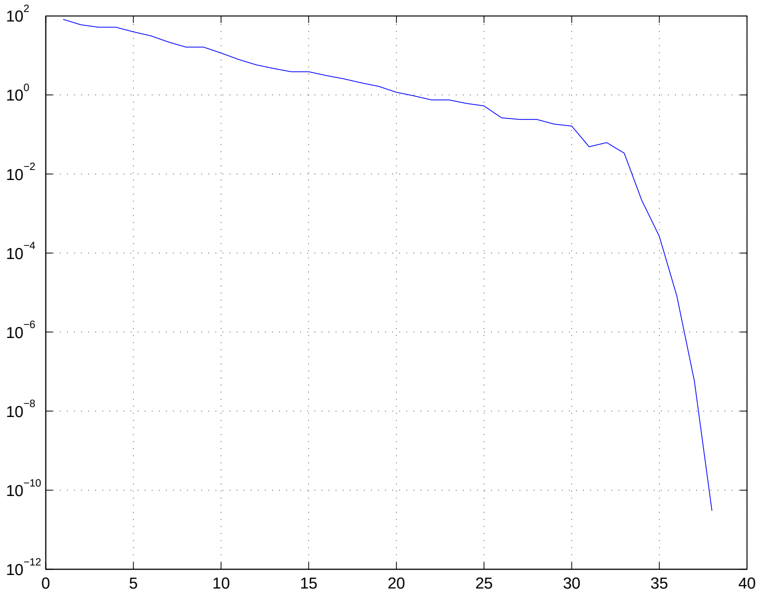

Figure 1: Illustration of the convergence for the SVD-RTR experiment. The vertical axis gives the measure $\|\hat{\Sigma} - \Sigma\|_F$ and the horizontal axis indicates the number of iterations of Algorithm 1.

7 Conclusion

We have proposed a trust-region approach for optimizing a smooth function on a Riemannian manifold. The technique relies on retractions that define particular one-to-one correspondences between the manifold and the tangent space at the current iterate. The Riemannian TR algorithms have, mutatis mutandis, the same convergence properties as the original algorithms in $\mathbb{R}^n$. Since several problems of numerical linear algebra can be expressed as an optimization problem on a Riemannian manifold, it can be anticipated that our general TR algorithm will lead to new computational algorithms and to new convergence results for existing algorithms; applications to full eigenvalue and singular value decomposition have been briefly presented, and an application to the computation of a few dominant or minor eigenvectors has been presented in detail in [ABG04a].

Acknowledgements

The authors wish to thank R. Sepulchre, R. Mahony, P. Van Dooren, U. Helmke, A. Edelman and S. T. Smith for useful discussions.

References

[ABG04a] P.-A. Absil, C. G. Baker, and K. A. Gallivan, *A superlinear method with strong global convergence properties for computing the extreme eigenvectors of a large symmetric matrix*, submitted to the 43rd IEEE Conference on Decision and Control, March 2004.

[ABG04b] ______, *Trust-region methods on Riemannian manifolds*, Tech. report, School of Computational Science and Information Technology, Florida State University, 2004.

[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, Massachusetts, 1995.

[Boo75] W. M. Boothby, *An introduction to differentiable manifolds and Riemannian geometry*, Academic Press, 1975.

[CGT00] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-region methods*, MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, and Mathematical Programming Society (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 Inc., Boston, MA, 1992, Translated from the second Portuguese edition by Francis Flaherty.

[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*, Journal of Optimization Theory and Applications **37** (1982), no. 2, 177–219.

[GV96] G. H. Golub and C. F. Van Loan, *Matrix computations, third edition*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 1996.

[HM94] U. Helmke and J. B. Moore, *Optimization and dynamical systems*, Springer, 1994.

[Lan99] S. Lang, *Fundamentals of differential geometry*, Graduate Texts in Mathematics, vol. 191, Springer-Verlag, New York, 1999.

[LE00] R. Lippert and A. Edelman, *Nonlinear eigenvalue problems with orthogonality constraints*, Templates for the Solution of Algebraic Eigenvalue Problems (Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst, eds.), SIAM, Philadelphia, 2000, pp. 290–314.

[Mah96] R. E. Mahony, *The constrained Newton method on a Lie group and the symmetric eigenvalue problem*, Linear Algebra Appl. **248** (1996), 67–89.

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

[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*, Studies in numerical analysis, MAA Stud. Math., vol. 24, Math. Assoc. America, Washington, DC, 1984, pp. 29–82.

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

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

[Smi93] S. T. Smith, *Geometric optimization methods for adaptive filtering*, Ph.D. thesis, Division of Applied Sciences, Harvard University, Cambridge, Massachusetts, 1993.

[Smi94] S. T. Smith, *Optimization techniques on Riemannian manifolds*, Hamiltonian and gradient flows, algorithms and control, Fields Inst. Commun., vol. 3, Amer. Math. Soc., Providence, RI, 1994, pp. 113–136.

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

[Toi81] Ph. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, 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 Publishers, 1994.

[Yan99] Y. Yang, *Optimization on Riemannian manifold*, Proceedings of the 38th Conference on Decision and Control, 1999.