

Trust-region methods on Riemannian manifolds[‡]P.-A. Absil*[†] C. G. Baker* K. A. Gallivan*[†]Technical Report FSU-CSIT-04-13
July 2, 2004**Abstract**

A general scheme for trust-region methods on Riemannian manifolds is proposed. A truncated conjugate-gradient algorithm is utilized to solve the trust-region subproblems. The method is illustrated on several problems from numerical linear algebra. In particular, for computing an extreme eigenspace of a symmetric/positive-definite matrix pencil, the method yields an efficient inverse-free superlinear algorithm, with good global convergence properties and minimal storage space requirements. This algorithm is related to a Krylov subspace method proposed by Golub and Ye [SIAM J. Sci. Comput. 24 (2002), 312–334] and to the trace minimization method of Sameh and Wisniewski [SIAM J. Numer. Anal. 19 (1982), 1243–1259], among others. Numerical experiments show that our trust-region method outperforms the Krylov subspace method on certain types of problems.

Key words. Numerical optimization on manifolds, trust-region, truncated conjugate-gradient, Steihaug-Toint, global convergence, local convergence, symmetric eigenvalue problem, singular value decomposition, symmetric/positive-definite matrix pencil, principal component analysis.

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.

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

[†]These authors' work was supported by the National Science Foundation under Grant ACI0324944 and by the School of Computational Science of Florida State University through a postdoctoral fellowship. This work was initiated while the first author was a Research Fellow with the Belgian National Fund for Scientific Research (FNRS) at the University of Liège.

[‡]A current electronic version is accessible from the first author's web page.

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

The Riemannian 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 $T_x M$ (see illustration on 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 $T_x M$. Since $T_x M$ 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 $0_x \in T_x M$. This minimizer is then retracted back from $T_x M$ 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 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. We assume throughout that it is computationally impossible or undesirable 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 Cauchy point by only using the Hessian of the model through Hessian-vector computations. 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, since the value of the cost function always decreases from one iterate to the next, it follows that 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. Moreover, we present a simple randomization technique that explicitly guarantees convergence to local minima with probability one.

Numerical linear algebra considers several problems that can be analyzed and solved using this approach. A particularly interesting application is the computation of the rightmost or leftmost eigenvalues and associated eigenvectors of a symmetric/positive-definite matrix pencil (A, B) . In this 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 in the case $B = I$. In Section 5.3, we consider the case of general positive-definite B and we propose a block generalization of the algorithm (the manifold is then a Grassmann manifold). It converges locally superlinearly to the leftmost (or rightmost, if preferred) p -dimensional eigenspace of (A, B) , with excellent global convergence properties, and without requiring inversion or factorization of A or B . Furthermore we investigate how the algorithm relates to the Tracemin algorithm of Sameh and Wisniewski [SW82, ST00] and with various restarted Krylov subspace methods, including an inverse-free Krylov subspace method for (A, B) recently proposed by Golub and Ye [GY02, Alg. 1]. Numerical experiments show that our trust-region algorithm can outperform the algorithm of [GY02].

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.

A preliminary version of the general Riemannian theory appeared in [ABG04b]. Part of the material in Section 5.3 was introduced in [ABG04a].

2 General theory

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

$$\min_{\eta \in \mathbb{R}^n} m(\eta) = f(x) + \partial f(x)\eta + \frac{1}{2}\eta^T \partial^2 f(x)\eta \quad \|\eta\| \leq \Delta \quad (1)$$

where $\partial f = (\partial_1 f, \dots, \partial_n f)$ is the differential of f , $(\partial^2 f)_{ij} = \partial_{ij}^2 f$ 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

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

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; or see e.g. [NW99, CGT00].

We will extend this concept of a trust-region subproblem to Riemannian manifolds. For this, it is useful to first consider the case of an abstract Euclidean space, i.e., a vector space endowed with an inner product (that is, a symmetric, bilinear, positive-definite form). The generalization to an 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, Section 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 $f : E \rightarrow \mathbb{R}$ and a current iterate $x \in E$, one can choose a basis $(e_i)_{i=1, \dots, d}$ of E (not necessarily orthonormal with respect to the inner product) and write a classical G -norm trust-region subproblem (see e.g. [GLRT99, Section 2])

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

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

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

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 $\phi_\alpha : \Omega_\alpha \subset M \rightarrow \mathbb{R}^d$; 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 $T_x M$ 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 $T_x M$. 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 $\text{Exp}_x \xi = \gamma(1)$ 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 $\text{Exp}_x \xi$ exists for all $x \in M$ and all $\xi \in T_x M$. 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 $T_x M$ 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 (see also the illustration on Figure 1).

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

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

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 $T_x M$. 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.

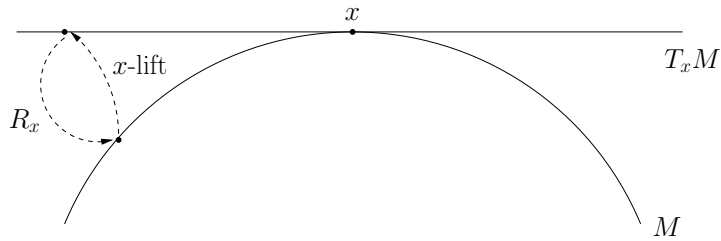


Figure 1: Illustration of retractions.

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 $f : M \rightarrow \mathbb{R}$ 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

$$\hat{f}_{x_k} : T_{x_k} M \rightarrow \mathbb{R} : \xi \mapsto f(R_{x_k} \xi) \quad (5)$$

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

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

Note that, since $DR_x(0_x) = \text{id}_{T_xM}$, it follows that $\text{grad } \hat{f}_{x_k}(0_x) = \text{grad } f(x)$, where $\text{grad } f(x)$, the gradient of f at x , is defined by $g_x(\text{grad } f(x), \xi) = \text{d}f_x(\xi)$, $\xi \in T_xM$ (see [dC92, Chap. 3, Ex. 8]). Moreover, we point out that if R satisfies some second order condition given in Lemma 4.10 page 18, then $\text{Hess } \hat{f}_{x_k}(0_x) = \text{Hess } f(x)$, where $\text{Hess } f(x) : T_xM \rightarrow T_xM$, the Hessian (linear) operator, is defined by

$$\text{Hess } f(x)\xi = \nabla_\xi \text{grad } f(x), \quad \xi \in T_xM, \quad (7)$$

see [dC92, Chap. 6, Ex. 11]. The Hessian operator is related to the second tensorial derivative $D^2f(x)$ by $D^2f(\xi, \chi) = \nabla_\chi \nabla_\xi f - \nabla_{\nabla_\chi \xi} f = g_x(\text{Hess } f(x)\xi, \chi)$; see [Lan99, Chap. XIII, Theorem 1.1].

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

$$\min_{\eta \in T_{x_k}M} m_{x_k}(\eta) = f(x_k) + g_{x_k}(\text{grad } f(x_k), \eta) + \frac{1}{2}g_{x_k}(\mathcal{H}_{x_k}\eta, \eta) \quad \text{s.t. } g_{x_k}(\eta, \eta) \leq \Delta_k^2, \quad (8)$$

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

Next, an (approximate) solution η_k of the trust-region subproblem (8) is computed, for example using a truncated conjugate-gradient method (see forthcoming Algorithm 2); several other possibilities are mentioned 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

$$x_+ = R_{x_k}(\eta_k). \quad (9)$$

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

$$\rho_k = \frac{f(x_k) - f(R_{x_k}(\eta_k))}{m_{x_k}(0_{x_k}) - m_{x_k}(\eta_k)} = \frac{\hat{f}_{x_k}(0_{x_k}) - \hat{f}_{x_k}(\eta_k)}{m_{x_k}(0_{x_k}) - m_{x_k}(\eta_k)}. \quad (10)$$

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 (RTR – basic Riemannian Trust-Region algorithm) *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 η_k by (approximately) solving (8);
 Evaluate ρ_k from (10);
 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].

We conclude this section by pointing out a link between Algorithm 1 and the Riemannian Newton method proposed independently by Smith [Smi94, Smi94] and Udriste [Udr94, Chap. 7, §5]. Assume that \mathcal{H}_{x_k} in (8) is the exact Hessian of f at x_k , and assume that the exact solution η^* of the trust-region subproblem (8) lies in the interior of the trust region. Then

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

which is equivalent to the Riemannian Newton equations of Smith and Udriste. Note that both authors propose to apply the update vector η^* using the Riemannian exponential retraction, namely, the new iterate is defined as $x_+ = \text{Exp}_x \eta^*$. 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 conditions hold [AMS04]. We will see in Section 4 that the superlinear convergence property 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. These properties are well known in the \mathbb{R}^n case.

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 $T_x M$. 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 impossible to detect positive-definiteness of \mathcal{H}_{x_k} . Rather, \mathcal{H}_{x_k} is only available via operations \mathcal{H}_{x_k} times vector.

The truncated conjugate-gradient (CG) algorithm is particularly appropriate for solving the trust-region subproblem (8) in these circumstances. The concept of truncated CG is due to Steihaug [Ste83] and Toint [Toi81]; therefore the method is sometimes referred to as the Steihaug-Toint algorithm, see e.g. [CGT00, Algorithm 7.5.1]. The following algorithm is a straightforward adaptation of the truncated CG algorithm of [Ste83] to the trust-region subproblem (8). 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 TR subproblem) Set $\eta^0 = 0$, $r_0 = \text{grad } f(x_k)$, $\delta_0 = -r_0$;
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 τ such that $\eta = \eta^j + \tau \delta_j$ minimizes $m(\eta)$ in (8)
 and satisfies $\|\eta\|_{g_x} = \Delta$;
 return η ;
 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 η ;
 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 tCG algorithm only requires the following:

- An evaluation of $\text{grad } f(x)$.
- A routine that performs line minimizations for the model m .
- A routine that returns $\mathcal{H}_{x_k} \delta$ given $\delta \in T_x M$.

The algorithm can thus be considered as “inverse-free”. The reader interested in the underlying principles of the Steihaug-Toint truncated CG method should refer to [Ste83], [NW99] or [CGT00].

We conclude this section with several comments about Algorithm 2.

The simplest stopping criterion for Algorithm 2 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

$$\|r_j\| \leq \|r_0\| \min(\|r_0\|^\theta, \kappa). \quad (11)$$

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

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

Algorithm 2 can be optimized in several ways. For example, there may be efficient formulas for computing $g_x(r, r)$ and $g_x(\delta, \mathcal{H}_{x_k} \delta)$; the value of $g_x(r_{j+1}, r_{j+1})$ can be kept in memory since it will be needed at the next iteration. Since the Hessian operator \mathcal{H}_{x_k} is an operator on a vector space of dimension d where d may be huge, much effort should be put in implementing an efficient routine for computing $\mathcal{H}_{x_k} \delta$. In many practical cases the tangent space $T_x M$ to which the quantities η , r and δ belong, will be represented as a linear subspace of a bigger Euclidean space; to prevent numerical errors it may be useful from time to time to re-project the above quantities onto the linear subspace.

In an attempt to make the algorithm more efficient, one could also make the stopping criterion depend on the value of ρ_{k-1} . If ρ_{k-1} is small, this means that the previous model was quite inaccurate and it is likely that the current model is also quite inaccurate. This would make one reluctant to compute the new η with high precision; thus the tCG process should be stopped after just a few steps.

The truncated CG algorithm is not the only available method for solving trust-region subproblems. Alternatives include the method of Moré and Sorensen [MS83], the dogleg method of Powell [Pow70], the double-dogleg method of Dennis and Mei [DM79], the two-dimensional subspace minimization strategy of Byrd *et al.* [BSS88] and the truncated Lanczos approach of Gould *et al.* [GLRT99]; see e.g. [CGT00, Section 7.5.4] for details.

The method proposed by Sorensen in [Sor97] violates our cost restrictions; it is appropriate only for applications where an eigenvalue problem can be solved as a computational primitive on each step and therefore will be considered in later work.

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 (8) 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 (11) and show that the iterates converge to nondegenerate stationary points with a rate of convergence of at least $\min\{\theta + 1, 2\}$.

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 $\lim_{k \rightarrow \infty} \|\text{grad } f(x_k)\| = 0$; this generalizes a classical convergence property of trust-region methods in \mathbb{R}^n , see [NW99, Theorem 4.8]. For this, it is useful to first consider some preliminary definitions and lemmas.

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

$$\hat{f} : TM \mapsto \mathbb{R} : \xi \mapsto f(R\xi) \tag{12}$$

and, in accordance with (5), \hat{f}_x denotes the restriction of \hat{f} to $T_x M$. We denote by $B_\delta(0_x) = \{\xi \in T_x M : \|\xi\| < \delta\}$ the open ball in $T_x M$ of radius δ centered at 0_x , and $B_\delta(x)$ stands for $\{y \in M : \text{dist}(x, y) < \delta\}$ where dist denotes the Riemannian distance. 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 (see [NW99, Thm 4.7] or [CGT00, Thm 6.4.5]), we first show that at least one accumulation point of $\{x_k\}$ is stationary. The convergence result requires that $m_{x_k}(\eta_k)$ is a sufficiently good approximation of $\hat{f}_{x_k}(\eta_k)$. In [CGT00, Thm 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 assumption¹, which leads us to consider the following definition.

¹It seems that $f \in C^1$ is not enough: there is apparently a gap in the proof of [NW99, Thm 4.7].

Definition 4.1 (radially L - C^1 function) Let $\hat{f} : TM \rightarrow \mathbb{R}$ be as in (12). We say that \hat{f} is radially Lipschitz continuously differentiable if there exist numbers $\beta_{RL} > 0$ and $\delta_{RL} > 0$ such that, for all $x \in M$, for all $\xi \in TM$ with $\|\xi\| = 1$, and for all $t < \delta_{RL}$, it holds

$$\left| \frac{d}{d\tau} \hat{f}_x(\tau\xi)|_{\tau=t} - \frac{d}{d\tau} \hat{f}_x(\tau\xi)|_{\tau=0} \right| \leq \beta_{RL} t. \quad (13)$$

Another important assumption in the global convergence result in \mathbb{R}^n is that the approximate solution η_k of the trust-region subproblem (8) 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 (8) is expressed on a Euclidean space, the definition of the Cauchy point is adapted from \mathbb{R}^n without difficulty, and the bound

$$m_k(0) - m_k(\eta_k) \geq c_1 \|\text{grad} f(x_k)\| \min \left(\Delta_k, \frac{\|\text{grad} f(x_k)\|}{\|\mathcal{H}_k\|} \right), \quad (14)$$

for some constant $c_1 > 0$, is readily obtained from the \mathbb{R}^n case, where

$$\|\mathcal{H}_k\| := \sup\{\|\mathcal{H}_k \zeta\| : \zeta \in T_{x_k} M, \|\zeta\| = 1\}. \quad (15)$$

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.

Finally, we allow the approximate solution of (8) to exceed the trust-region radius by some constant multiple,

$$\|\eta_k\| \leq \gamma \Delta_k, \quad \text{for some constant } \gamma \geq 1. \quad (16)$$

With these things in place, we can state and prove the first global convergence results. Note that we have deliberately chosen to present this theorem under weak assumptions; stronger but arguably easier to check assumptions will be presented in Proposition 4.5.

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 C^1 and bounded below on the level set (25), that \hat{f} is radially L - C^1 (Definition 4.1), and that $\|\mathcal{H}_k\| \leq \beta$ for some constant β . Further suppose that all approximate solutions η_k of (8) satisfy the inequalities (14) and (16), for some positive constants c_1 and γ . We then have

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

Proof.

First, we perform some manipulation of ρ_k from (10). Notice that

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{(f(x_k) - \hat{f}_{x_k}(\eta_k)) - (m_k(0) - m_k(\eta_k))}{m_k(0) - m_k(\eta_k)} \right| \\ &= \left| \frac{m_k(\eta_k) - \hat{f}_{x_k}(\eta_k)}{m_k(0) - m_k(\eta_k)} \right|. \end{aligned} \quad (17)$$

Direct manipulations on the function $t \mapsto \hat{f}_{x_k}(t \frac{\eta_k}{\|\eta_k\|})$ yield

$$\begin{aligned} \hat{f}_{x_k}(\eta_k) &= \hat{f}_{x_k}(0_{x_k}) + \|\eta_k\| \frac{d}{d\tau} \hat{f}_{x_k}(\tau \frac{\eta_k}{\|\eta_k\|})|_{\tau=0} + \int_0^{\|\eta_k\|} \left(\frac{d}{d\tau} \hat{f}_{x_k}(\tau \frac{\eta_k}{\|\eta_k\|})|_{\tau=t} - \frac{d}{d\tau} \hat{f}_{x_k}(\tau \frac{\eta_k}{\|\eta_k\|})|_{\tau=0} \right) dt \\ &= f(x_k) + g_{x_k}(\text{grad } f(x_k), \eta_k) + \epsilon' \end{aligned}$$

where $|\epsilon'| < \int_0^{\|\eta_k\|} \beta_{RL} t dt = \frac{1}{2} \beta_{RL} \|\eta_k\|^2$ whenever $\|\eta_k\| < \delta_{RL}$, and β_{RL} and δ_{RL} are the constants in the radially L- C^1 property (13).

Therefore, it follows from the definition (8) of m_k that

$$\begin{aligned} |m_k(\eta_k) - \hat{f}_{x_k}(\eta_k)| &= \left| \frac{1}{2} g_{x_k}(\mathcal{H}_{x_k} \eta_k, \eta_k) - \epsilon' \right| \\ &\leq \frac{1}{2} \beta \|\eta_k\|^2 + \frac{1}{2} \beta_{RL} \|\eta_k\|^2 \leq \beta' \|\eta_k\|^2 \end{aligned} \quad (18)$$

whenever $\|\eta_k\| < \delta_{RL}$, where $\beta' = \max(\beta, \beta_{RL})$.

Assume for purpose of contradiction that the theorem does not hold; that is, assume there exist $\epsilon > 0$ and a positive index K such that

$$\|\text{grad } f(x_k)\| \geq \epsilon, \quad \forall k \geq K. \quad (19)$$

From (14), for $k \geq K$, we have

$$m_k(0) - m_k(\eta_k) \geq c_1 \|\text{grad } f(x_k)\| \min \left(\Delta_k, \frac{\|\text{grad } f(x_k)\|}{\|\mathcal{H}_k\|} \right) \geq c_1 \epsilon \min \left(\Delta_k, \frac{\epsilon}{\beta'} \right). \quad (20)$$

Substituting (16), (18), and (20) into (17), we have that

$$|\rho_k - 1| \leq \frac{\beta' \|\eta_k\|^2}{c_1 \epsilon \min \left(\Delta_k, \frac{\epsilon}{\beta'} \right)} \leq \frac{\beta' \gamma^2 \Delta_k^2}{c_1 \epsilon \min \left(\Delta_k, \frac{\epsilon}{\beta'} \right)} \quad (21)$$

whenever $\|\eta_k\| < \delta_{RL}$.

We can choose a value of $\hat{\Delta}$ that allows us to bound the right-hand-side of the inequality (21), when $\Delta_k \leq \hat{\Delta}$. Choose $\hat{\Delta}$ as follows:

$$\hat{\Delta} \leq \min \left(\frac{c_1 \epsilon}{2 \beta' \gamma^2}, \frac{\epsilon}{\beta'}, \delta_{RL} \right).$$

This gives us $\min \left(\Delta_k, \frac{\epsilon}{\beta'} \right) = \Delta_k$. We can now write (21) as follows:

$$|\rho_k - 1| \leq \frac{\beta' \gamma^2 \hat{\Delta} \Delta_k}{c_1 \epsilon \min \left(\Delta_k, \frac{\epsilon}{\beta'} \right)} \leq \frac{\Delta_k}{2 \min \left(\Delta_k, \frac{\epsilon}{\beta'} \right)} = \frac{1}{2}.$$

Therefore, $\rho_k \geq \frac{1}{2} > \frac{1}{4}$ whenever $\Delta_k \leq \hat{\Delta}$, so that by the workings of Algorithm 1, it follows (from the argument above) that $\Delta_{k+1} \geq \Delta_k$ whenever $\Delta_k \leq \hat{\Delta}$. It follows that a reduction of Δ_k (by a factor of $\frac{1}{4}$) can occur in Algorithm 1 only when $\Delta_k > \hat{\Delta}$.

Therefore, we conclude that

$$\Delta_k \geq \min \left(\Delta_K, \hat{\Delta}/4 \right), \quad \forall k \geq K. \quad (22)$$

Suppose now that there is an infinite subsequence \mathcal{K} such that $\rho_k \geq \frac{1}{4} > \rho'$ for $k \in \mathcal{K}$. If $k \in \mathcal{K}$ and $k \geq K$, we have from (20) that

$$\begin{aligned} f(x_k) - f(x_{k+1}) &= f_{x_k} - \hat{f}_{x_k}(\eta_k) \\ &\geq \frac{1}{4}(m_k(0) - m_k(\eta_k)) \\ &\geq \frac{1}{4}c_1\epsilon \min \left(\Delta_k, \frac{\epsilon}{\beta'} \right). \end{aligned}$$

Since f is bounded below on the level set containing these iterates, it follows from this inequality that

$$\lim_{k \in \mathcal{K}, k \rightarrow \infty} \Delta_k = 0,$$

clearly contradicting (22). Then such an infinite subsequence as \mathcal{K} cannot exist. It follows that we must have $\rho_k < \frac{1}{4}$ for all k sufficiently large, so that Δ_k will be reduced by a factor of $\frac{1}{4}$ on every iteration. Then we have, $\lim_{k \rightarrow \infty} \Delta_k = 0$, which again contradicts (22). Then our original assumption (19) must be false, giving us the desired result. \square

To 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 $x, y \in \mathbb{R}^n$,

$$\|\text{grad}f(y) - \text{grad}f(x)\| \leq \beta_1 \|y - x\|. \quad (23)$$

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 (23) naturally becomes the Riemannian distance $\text{dist}(x, y)$. For the left-hand side of (23), observe that the operation $\text{grad}f(x) - \text{grad}f(y)$ is not well-defined in general on a Riemannian manifold since $\text{grad}f(x)$ and $\text{grad}f(y)$ 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 $\alpha(t) = \text{Exp}_x(t\text{Exp}_x^{-1}y)$ such that $\alpha(0) = x$ and $\alpha(1) = y$, and we can parallel transport $\text{grad}f(y)$ along α to obtain the vector $P_\alpha^{0 \leftarrow 1} \text{grad}f(y)$ in T_xM , to yield the following definition.

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

$$\|P_\alpha^{0 \leftarrow 1} \text{grad}f(y) - \text{grad}f(x)\| \leq \beta_1 \text{dist}(y, x), \quad (24)$$

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

Note that (24) is symmetric in x and y ; indeed, since the parallel transport is an isometry, it follows that

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

For the purpose of Algorithm 1, which is a descent algorithm, condition (24) need only to be imposed for all x, y in the level set

$$\{x \in M : f(x) \leq f(x_0)\}. \quad (25)$$

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

$$\|\xi\| \geq \mu d(x, R_x\xi), \quad \forall x \in M, \forall \xi \in T_xM, \|\xi\| < \delta_\mu \quad (26)$$

Note that for the exponential retraction discussed in this paper, (26) is satisfied as an equality, with $\mu = 1$.

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 $\rho' \in (0, \frac{1}{4})$, that f is Lipschitz continuously differentiable (Definition 4.3), and that (26) is satisfied for some $\mu > 0$. It then follows that*

$$\lim_{k \rightarrow \infty} \text{grad} f(x_k) = 0.$$

Proof.

Consider any index m such that $\text{grad} f(x_m) \neq 0$. The satisfaction of (24) on the level set (25) gives us

$$\|P_\alpha^{1 \leftarrow 0} \text{grad} f(x) - \text{grad} f(x_m)\| \leq \beta_1 \text{dist}(x, x_m)$$

for any x in the level set. Define scalars

$$\epsilon = \frac{1}{2} \|\text{grad} f(x_m)\|, \quad r = \min \left(\frac{\|\text{grad} f(x_m)\|}{2\beta_1}, i(M) \right) = \min \left(\frac{\epsilon}{\beta_1}, i(M) \right)$$

Define the ball $B_r(x_m) := \{x : \text{dist}(x, x_m) < r\}$.

Then for any $x \in B_r(x_m)$, we have

$$\begin{aligned} \|\text{grad} f(x)\| &= \|P_\alpha^{0 \leftarrow 1} \text{grad} f(x)\| \\ &= \|P_\alpha^{0 \leftarrow 1} \text{grad} f(x) + \text{grad} f(x_m) - \text{grad} f(x_m)\| \\ &\geq \|\text{grad} f(x_m)\| - \|P_\alpha^{0 \leftarrow 1} \text{grad} f(x) - \text{grad} f(x_m)\| \\ &\geq 2\epsilon - \beta_1 \text{dist}(x, x_m) \\ &> 2\epsilon - \beta_1 \min \left(\frac{\|\text{grad} f(x_m)\|}{2\beta_1}, i(M) \right) \\ &\geq 2\epsilon - \frac{1}{2} \|\text{grad} f(x_m)\| \\ &= \epsilon. \end{aligned}$$

If the entire sequence $\{x_k\}_{k \geq m}$ stays inside of the ball $B_r(x_m)$, then we would have $\|\text{grad} f(x_k)\| > \epsilon$ for all $k \geq m$, which contradicts the results of Theorem 4.2. Then the sequence eventually leaves the ball $B_r(x_m)$.

Let the index $l \geq m$ be such that x_{l+1} is the first iterate after x_m outside of $B_r(x_m)$. Since $\|\text{grad}f(x_k)\| > \epsilon$ for $k = m, m+1, \dots, l$, we have

$$\begin{aligned}
f(x_m) - f(x_{l+1}) &= \sum_{k=m}^l f(x_k) - f(x_{k+1}) \\
&\geq \sum_{k=m, x_k \neq x_{k+1}}^l \rho'(m_k(0) - m_k(\eta_k)) \\
&\geq \sum_{k=m, x_k \neq x_{k+1}}^l \rho'c_1 \|\text{grad}f(x_k)\| \min\left(\Delta_k, \frac{\|\text{grad}f(x_k)\|}{\|B_k\|}\right) \\
&\geq \sum_{k=m, x_k \neq x_{k+1}}^l \rho'c_1 \epsilon \min\left(\Delta_k, \frac{\epsilon}{\beta}\right).
\end{aligned}$$

If $\Delta_k \leq \epsilon/\beta$ for all $k = m, m+1, \dots, l$, then

$$\begin{aligned}
f(x_m) - f(x_{l+1}) &= \rho'c_1 \epsilon \sum_{k=m, x_k \neq x_{k+1}}^l \Delta_k \geq \rho'c_1 \epsilon \sum_{k=m, x_k \neq x_{k+1}}^l \frac{1}{\gamma} \|\eta_k\| \\
&\geq \rho'c_1 \epsilon \frac{1}{\gamma} \sum_{k=m, x_k \neq x_{k+1}}^l \mu d(x_k, R_{x_k} \eta_k) \\
&= \rho'c_1 \epsilon \frac{\mu}{\gamma} \sum_{k=m, x_k \neq x_{k+1}}^l d(x_k, x_{k+1}) \\
&\geq \rho'c_1 \epsilon \frac{\mu}{\gamma} r = \rho'c_1 \epsilon \frac{\mu}{\gamma} \min\left(\frac{\epsilon}{\beta_1}, i(M)\right). \tag{27}
\end{aligned}$$

If not, then $\Delta_k > \epsilon/\beta$ for some $k \in \{m, \dots, l\}$, so that

$$f(x_m) - f(x_{l+1}) \geq \rho'c_1 \epsilon \frac{\epsilon}{\beta}. \tag{28}$$

Then because $\{f(x_k)\}_{k=0}^\infty$ is decreasing and bounded below, we have

$$f(x_k) \downarrow f^*, \tag{29}$$

for some $f^* > -\infty$. Then using (27) and (28), we get

$$\begin{aligned}
f(x_m) - f^* &\geq f(x_m) - f(x_{l+1}) \\
&\geq \rho'c_1 \epsilon \min\left(\frac{\epsilon}{\beta}, \frac{\epsilon\mu}{\beta_1\gamma}, \frac{i(M)\mu}{\gamma}\right) \\
&= \frac{1}{2} \rho'c_1 \|\text{grad}f(x_m)\| \min\left(\frac{\|\text{grad}f(x_m)\|}{2\beta}, \frac{\|\text{grad}f(x_m)\|\mu}{2\beta_1\gamma}, \frac{i(M)\mu}{\gamma}\right).
\end{aligned}$$

Assume for the purpose of contradiction that it is not the case that $\lim_{m \rightarrow \infty} \|\text{grad}f(x_m)\| = 0$. Then there exists $\omega > 0$ and an infinite sequence \mathcal{K} such that

$$\|\text{grad}f(x_k)\| > \omega, \quad \forall k \in \mathcal{K}.$$

Then for $k \in \mathcal{K}, k \geq m$, we have

$$\begin{aligned} f(x_k) - f^* &\geq \frac{1}{2}\rho'c_1\|\text{grad}f(x_k)\| \min\left(\frac{\|\text{grad}f(x_k)\|}{2\beta}, \frac{\|\text{grad}f(x_k)\|\mu}{2\beta_1\gamma}, \frac{i(M)\mu}{\gamma}\right) \\ &> \frac{1}{2}\rho'c_1\omega \min\left(\frac{\omega}{2\beta}, \frac{\omega\mu}{2\beta_1\gamma}, \frac{i(M)\mu}{\gamma}\right) \\ &> 0. \end{aligned}$$

However, this contradicts $\lim_{k \rightarrow \infty} (f(x_k) - f^*) = 0$, so that our hypothetical assumption must be false, and

$$\lim_{m \rightarrow \infty} \|\text{grad}f(x_m)\| = 0.$$

□

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 (26), the Lipschitz condition (24) reduces to the classical expression, which subsumes the radially L - C^1 condition.

The following proposition 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 $t \mapsto Rt\xi$. 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 $\|\text{grad}f(x)\| \leq \beta_g$ and $\|\text{Hess}f(x)\| \leq \beta_H$ for some constants β_g, β_H , and all $x \in M$. Moreover suppose that*

$$\left\| \frac{D}{dt} \frac{d}{dt} Rt\xi \right\| \leq \beta_D \tag{30}$$

for some constant β_D , for all $\xi \in TM$ with $\|\xi\| = 1$ and all $t < \delta_D$, where $\frac{D}{dt}$ denotes the covariant derivative along the curve $t \mapsto Rt\xi$ (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 (26) on R is satisfied for values of μ and δ_μ satisfying $\delta_\mu < \delta_D$ and $\frac{1}{2}\beta_D\delta_\mu < \frac{1}{\mu} - 1$.

Proof. By a standard Taylor argument (see Lemma 4.6), boundedness of the Hessian of f implies the Lipschitz- C^1 property of f .

For (26), observe that

$$\text{dist}(x, Rt\xi) \leq L(u(0, t)) \leq \int_0^t \|\dot{u}(\tau)\| d\tau$$

where $L(u(0, t))$ denotes the length of the curve u between 0 and t . Using the Cauchy-Schwarz inequality and the invariance of the metric by the connection, we have

$$\left| \frac{d}{d\tau} \|\dot{u}(\tau)\| \right| = \left| \frac{d}{d\tau} \sqrt{g_{u(\tau)}(\dot{u}(\tau), \dot{u}(\tau))} \right| = \left| \frac{g_{u(\tau)}\left(\frac{D}{dt}\dot{u}(\tau), \dot{u}(\tau)\right)}{\|\dot{u}(\tau)\|} \right| \leq \frac{\beta_D \|\dot{u}(\tau)\|}{\|\dot{u}(\tau)\|} \leq \beta_D$$

for all $t < \delta_D$. Therefore

$$\int_0^t \|\dot{u}(\tau)\| d\tau \leq \int_0^t \|\dot{u}(0)\| + \beta_D\tau d\tau = \|\xi\|t + \frac{1}{2}\beta_D t^2 = t + \frac{1}{2}\beta_D t^2,$$

which is smaller than $\frac{t}{\mu}$ if $\frac{1}{2}\beta_D t < \frac{1}{\mu} - 1$.

For the radially Lipschitz- C^1 condition, let $u(t) = Rt\xi$ and $h(t) = f(u(t)) = \hat{f}(t\xi)$ with $\xi \in T_x M$, $\|\xi\| = 1$. Then

$$\dot{h}(t) = g_{u(t)}(\text{grad } f(u(t)), \dot{u}(t))$$

and

$$\ddot{h}(t) = \frac{D}{dt} g_{u(t)}(\text{grad } f(u(t)), \dot{u}(t)) = g_{u(t)}\left(\frac{D}{dt} \text{grad } f(u(t)), \dot{u}(t)\right) + g_{u(t)}(\text{grad } f(u(t)), \frac{D}{dt} \dot{u}(t)).$$

Now, $\frac{D}{dt} \text{grad } f(u(t)) = \nabla_{\dot{u}(t)} \text{grad } f(u(t)) = \text{Hess } f(u(t))[\dot{u}(t)]$. It follows that $|\ddot{h}(t)|$ is bounded on $t \in [0, \delta_D)$ by the constant $\beta_{RL} = \beta_H(1 + \beta_D \delta_D) + \beta_g \beta_D$. Then

$$|\dot{h}(t) - \dot{h}(0)| \leq \int_0^t |\ddot{h}(\tau)| d\tau \leq t\beta_{RL}.$$

□

4.2 Local convergence

We now state local convergence properties of Algorithm 1-2 (i.e., Algorithm 1 where the trust-region subproblem (8) is solved approximately with Algorithm 2). We first state a few preparation lemmas.

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 (note that similar Taylor developments on manifolds can be found in [Smi94]).

Lemma 4.6 (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 $y \in V$,*

$$P_\gamma^{0 \leftarrow 1} \zeta_y = \zeta_x + \nabla_\xi \zeta + \int_0^1 P_\gamma^{0 \leftarrow \tau} \nabla_{\gamma'(\tau)} \zeta - \nabla_\xi \zeta d\tau \quad (31)$$

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

Proof. Start from

$$P_\gamma^{0 \leftarrow 1} \zeta_y = \zeta_x + \int_0^1 \frac{d}{d\tau} P_\gamma^{0 \leftarrow \tau} \zeta d\tau = \zeta_x + \nabla_\xi \zeta + \int_0^1 \left(\frac{d}{d\tau} P_\gamma^{0 \leftarrow \tau} \zeta - \nabla_\xi \zeta \right) d\tau$$

and use the formula for the connection in terms of the parallel transport, see [dC92, Chap. 2, Ex. 2], to obtain

$$\frac{d}{d\tau} P_\gamma^{0 \leftarrow \tau} \zeta = \frac{d}{d\epsilon} P_\gamma^{0 \leftarrow \tau} P_\gamma^{\tau \leftarrow \tau + \epsilon} \zeta = P_\gamma^{0 \leftarrow \tau} \nabla_{\gamma'} \zeta.$$

□

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

Lemma 4.7 *Let $v \in M$ and let f be a C^2 cost function such that $\text{grad } f(v) = 0$ and $\text{Hess } f(v)$ is nonsingular. Then there exist a neighborhood V of v and numbers c_0 and c_1 such that, for all $x \in V$,*

$$c_0 \text{dist}(v, x) \leq \|\text{grad } f(x)\| \leq c_1 \text{dist}(v, x). \quad (32)$$

Proof. From Taylor (Lemma 4.6), it follows that

$$P_\gamma^{0 \leftarrow 1} \text{grad } f(y) = \text{Hess } f(x)[\gamma'(0)] + \int_0^1 P_\gamma^{0 \leftarrow \tau} \text{Hess } f(\gamma(\tau))[\gamma'(\tau)] - \text{Hess } f(x)[\gamma'(0)] d\tau. \quad (33)$$

Since f is C^2 and since $\|\gamma'(\tau)\| = \text{dist}(x, y)$ for all $\tau \in [0, 1]$, we have the following bound for the integral in (33):

$$\begin{aligned} & \left\| \int_0^1 P_\gamma^{0 \leftarrow \tau} \text{Hess } f(\gamma(\tau))[\gamma'(\tau)] - \text{Hess } f(x)[\gamma'(0)] d\tau \right\| \\ &= \left\| \int_0^1 (P_\gamma^{0 \leftarrow \tau} \circ \text{Hess } f(\gamma(\tau)) \circ P_\gamma^{\tau \leftarrow 0} - \text{Hess } f(x)) [\gamma'(0)] d\tau \right\| \leq \epsilon(\text{dist}(x, y)) \text{dist}(x, y) \end{aligned}$$

where $\lim_{t \rightarrow 0} \epsilon(t) = 0$. Let λ_{\min} and λ_{\max} be the minimal and maximal eigenvalues in norm of the operator $\text{Hess } f(x)$. Since $\text{Hess } f(v)$ is nonsingular, it follows that $|\lambda_{\min}| > 0$. Take V sufficiently small so that $\epsilon(\text{dist}(x, y)) < |\lambda_{\min}|/2$ for all y in V . Then, using the fact that the parallel translation is an isometry, it follows from (33) that

$$\frac{|\lambda_{\min}|}{2} \text{dist}(x, y) \leq \|\text{grad } f(y)\| \leq 2|\lambda_{\max}| \text{dist}(x, y),$$

and the result is proven. \square

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

Lemma 4.8 *Let R be a C^2 retraction on M , let f be a C^1 cost function on M , and let v belong to M . Then there exist a neighborhood V of v and a function $\varepsilon : TV \rightarrow \mathbb{R}$ with $\lim_{\delta \rightarrow 0} \sup_{\|\xi\|=\delta} \varepsilon(\xi) = 0$, such that*

$$\|\text{grad } f(R\xi)\| = (1 + \varepsilon(\xi)) \|\text{grad } \hat{f}(\xi)\|,$$

where \hat{f} is as in (12).

Proof. Let $A(\xi)$ denote the differential of R_x at $\xi \in T_x M$. Consider a parameterization of M at v , and consider the corresponding parameterization of TM (see [dC92, Chap. 0, Example 4.1]). Using Einstein's summation convention, and denoting $\partial_i f$ by $f_{,i}$, we have

$$\hat{f}_{x,i}(\xi) = f_{,j}(R\xi) A_j^i(\xi),$$

where $A(\xi)$ stands for the differential of R_x at $\xi \in T_x M$. Then,

$$\|\text{grad } \hat{f}_x(\xi)\|^2 = \hat{f}_{x,i}(\xi) g^{ij}(x) \hat{f}_{x,j}(\xi) = f_{,k}(R_x \xi) A_i^k(\xi) g^{ij}(x) A_j^\ell(\xi) f_{,\ell}(R_x \xi)$$

and

$$\|\text{grad } f(R_x \xi)\|^2 = f_{,j}(R_x \xi) g^{ij}(R_x \xi) f_{,j}(R_x \xi).$$

The conclusion follows by a real analysis argument, invoking the smoothness properties of R and g and using $A(0_x) = \text{id}$. \square

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

Lemma 4.9 *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., $\text{grad } f(v) = 0$). Then $\text{Hess } \hat{f}_v(0_v) = \text{Hess } f(v)$.*

Proof. Let A denote the differential of R_v at 0_v . Working in a parameterization of M at v , one obtains that $(\text{Hess } \hat{f}_v(0_v))_j^i = g^{ik} \partial_j \partial_k f = (\text{Hess } f(v))_j^i$. \square

Away from the stationary points, the Hessians $\text{Hess } f(x)$ and $\text{Hess } \hat{f}_x(0_x)$ do not coincide. They do coincide if a “zero acceleration” condition (34) is imposed on the retraction. This result will not be used in the convergence analysis but we state it below for completeness.

Lemma 4.10 *Suppose that*

$$\frac{D}{dt} \left(\frac{d}{dt} R_t \xi \right) \Big|_{t=0} = 0, \quad \text{for all } \xi \in TM, \quad (34)$$

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

Proof. Observe that $D^2 f(x)[\xi, \xi] = \frac{d^2}{dt^2} f(\text{Exp}_x t\xi) \Big|_{t=0}$ and $D^2 \hat{f}(0_x)[\xi, \xi] = \frac{d^2}{dt^2} f(R_x t\xi) \Big|_{t=0} = \frac{d}{dt} (df \frac{d}{dt} R_x t\xi) \Big|_{t=0} = \nabla_\xi df \xi + df \frac{D}{dt} (\frac{d}{dt} R_x t\xi) \Big|_{t=0}$. The result follows from the definitions of the Hessians and the one-to-one correspondance between symmetric bilinear forms and quadratic forms. \square

We now state and prove the local convergence results. We first show 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 4.11 (local convergence to local minima) *Consider Algorithm 1-2—i.e., the Riemannian trust-region algorithm where the trust-region subproblems (8) are solved using the truncated CG algorithm with stopping criterion (11)—with all the assumptions of Theorem 4.2. Let v be a nondegenerate local minimum of f , i.e., $\text{grad } f(v) = 0$ and $\text{Hess } f(v)$ is positive definite. Assume that $\|\mathcal{H}_k^{-1}\|$ is bounded and that (26) holds for some $\mu > 0$ and $\delta_\mu > 0$.*

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 .

Proof. Take $\delta_1 > 0$ with $\delta_1 < \delta_\mu$ such that $B_{\delta_1}(v)$ is a normal neighborhood of v , which contains only v as stationary point, and such that $f(x) > f(v)$ for all $x \in \bar{B}_{\delta_1}(v)$. (Such a δ_1 exists...). If we can exclude that $\{x_k\}$ leaves $B_{\delta_1}(v)$, then we are done, because of Theorem 4.2 and because $f(x_k)$ is nonincreasing.

Take δ_2 so small that for all $x \in B_{\delta_2}(v)$, it holds that $\|\eta^*(x)\| \leq \mu(\delta_1 - \delta_2)$, where η^* is the (unique) solution of $\mathcal{H}\eta^* = -\text{grad } f(x)$; such a δ_2 exists because of Lemma 4.7 and the bound on $\|\mathcal{H}_k^{-1}\|$. Consider a level set \mathcal{L} of f such that $\mathcal{L}' := \mathcal{L} \cap B_{\delta_1}(v) \subset B_{\delta_2}(v)$; invoke that $f \in C^1$ to show that such a level set exists. Then, for all $x \in \mathcal{L}'$, we have

$$\text{dist}(x, x_+) \leq \frac{1}{\mu} \|\eta^{tCG}(x, \Delta)\| \leq \frac{1}{\mu} \|\eta^*\| \leq (\delta_1 - \delta_2),$$

where we used the fact that $\|\eta\|$ is increasing along the truncated CG process [Ste83, Thm 2.1]. It follows from the equation above that x_+ is in $B_{\delta_1}(v)$. Moreover, since $f(x_+) \leq f(x)$, it follows that $x_+ \in \mathcal{L}'$. Thus \mathcal{L}' is invariant. But its only stationary point is v . So there is a subsequence of $\{x_k\}$ that goes to v . But since v is the global minimum in \mathcal{L}' , the whole sequence must go to v . \square

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

Theorem 4.12 (rate of convergence) Consider Algorithm 1-2. Suppose that f is a C^2 cost function on M and that

$$\|\mathcal{H}_k - \text{Hess } \hat{f}_{x_k}(0_k)\| \leq \beta_{\mathcal{H}} \|\text{grad } f(x_k)\|, \quad (35)$$

that is, \mathcal{H}_k is a sufficiently good approximation of $\text{Hess } \hat{f}_{x_k}(0_{x_k})$. Let $v \in M$ be a nondegenerate local minimum of f , (i.e., $\text{grad } f(v) = 0$ and $\text{Hess } f(v)$ is positive definite). Further assume that $\text{Hess } \hat{f}_{x_k}$ is Lipschitz-continuous at 0_x uniformly in x in a neighborhood of v , i.e., 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

$$\|\text{Hess } \hat{f}_{x_k}(\xi) - \text{Hess } \hat{f}_{x_k}(0_{x_k})\| \leq \beta_{L2} \|\xi\|, \quad (36)$$

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 Algorithm 1-2 converging to v , there exists $K > 0$ such that for all $k > K$,

$$\text{dist}(x_{k+1}, v) \leq c (\text{dist}(x_k, v))^{\min\{\theta+1, 2\}} \quad (37)$$

with $\theta > 0$ as in (11).

Proof.

We will show below 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 , it holds

$$c_0 \text{dist}(v, x_k) \leq \|\text{grad } f(x_k)\| \leq c_1 \text{dist}(v, x_k), \quad (38)$$

$$\|\eta_k\| \leq c_4 \|\text{grad } m_{x_k}(0)\| \leq \tilde{\Delta}, \quad (39)$$

$$\rho_k > \rho' \quad (40)$$

$$\|\text{grad } f(R_{x_k} \xi)\| \leq c_5 \|\text{grad } \hat{f}_{x_k}(\xi)\|, \quad (41)$$

$$\|\text{grad } m_{x_k}(\xi) - \text{grad } \hat{f}_{x_k}(\xi)\| \leq c_3 \|\xi\|^2 + c'_3 \|\text{grad } f(x_k)\| \|\xi\|, \quad (42)$$

$$\|\text{grad } m_{x_k}(\eta_k)\| \leq c_2 \|\text{grad } m_{x_k}(0)\|^{\theta+1}, \quad (43)$$

where $\{\eta_k\}$ is the sequence of update vectors corresponding to $\{x_k\}$.

With these results at hand the proof is concluded as follows. For all $k > K$, it follows from (38) and (40) that

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

from (41) and (39) that

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

from (39) and (42) and (43) that

$$\begin{aligned} \|\text{grad } \hat{f}_{x_k}(\eta_k)\| &\leq \|\text{grad } m_{x_k}(\eta_k) - \text{grad } \hat{f}_{x_k}(\eta_k)\| + \|\text{grad } m_{x_k}(\eta_k)\| \\ &\leq (c_3 c_4^2 + c'_3 c_4) \|\text{grad } m_{x_k}(0)\|^2 + c_2 \|\text{grad } m_{x_k}(0)\|^{1+\theta}, \end{aligned}$$

and from (38) that

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

Consequently, since $1 + \theta \leq 2$, taking K larger if necessary so that $\text{dist}(v, x_k) < 1$ for all $k > K$, it follows that

$$\begin{aligned} \text{dist}(v, x_{k+1}) &\leq \frac{1}{c_0} c_5 ((c_3 c_4^2 + c'_3 c_4) c_1^2 (\text{dist}(v, x_k))^2 + c_2 c_1^{1+\theta} (\text{dist}(v, x_k))^{1+\theta}) \\ &\leq \frac{1}{c_0} c_5 ((c_3 c_4^2 + c'_3 c_4) c_1^2 + c_2 c_1^{1+\theta}) (\text{dist}(v, x_k))^{\min\{2, 1+\theta\}} \end{aligned}$$

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

It remains to prove the bounds (38)-(43).

Equation (38) comes from Lemma 4.7 and is due to the fact that v is a nondegenerate critical point.

We prove (39). Since $\{x_k\}$ converges to the nondegenerate local minimum v where $\text{Hess } \hat{f}_v(0_v) = \text{Hess } f(v)$ (see Lemma 4.9) and since $\text{Hess } f(v)$ is positive definite with $f \in C^2$, it follows from the approximation condition (35) and from (38) that there exist $c_4 > 0$ such that $\|\mathcal{H}_k^{-1}\| < c_4$ for all k greater than some K . Given a $k > K$, let η^* be the solution of $\mathcal{H}_{x_k} \eta^* = -\text{grad } m_{x_k}(0)$. It follows that $\|\eta^*\| \leq c_4 \|\text{grad } m_{x_k}(0)\|$. Then, since the sequence of η_k^j 's constructed by the tCG inner iteration (Algorithm 2) is strictly increasing in norm (see [Ste83, Theorem 2.1]) and would eventually reach η^* at $j = d$, it follows that (39) holds. The second inequality in (39) comes for any given $\tilde{\Delta}$ by choosing K larger if necessary.

We prove (40). Let γ_k denote $\|\text{grad } f(x_k)\|$. From the definition (10) of ρ_k , from the assumption (35) that $\|\mathcal{H}_k - \text{Hess } \hat{f}_{x_k}\| \leq \beta_{\mathcal{H}} \gamma_k$, and from the Lipschitz assumption (36) on the Hessian of \hat{f} , it follows by a classical Taylor argument in the Euclidean space $T_x M$ that

$$\rho_k = \frac{m_k(0_k) - m_k(\eta_k) + \varepsilon(\|\eta_k\|^3)}{m_k(0_k) - m_k(\eta_k)} = 1 + \frac{\varepsilon(\|\eta_k\|^3)}{m_k(0_k) - m_k(\eta_k)},$$

where $0 \leq \varepsilon(t) \leq \frac{\beta_{L2} + \beta_{\mathcal{H}} \gamma_k}{6} t$ for all $t < \delta_2$. It then follows from $\|\eta_k\| \leq \Delta_k$, from the bound (39) and from the Cauchy decrease hypothesis (14), that

$$|\rho_k - 1| \leq \frac{(\beta_{L2} + \beta_{\mathcal{H}} \gamma_k) (\min\{\Delta_k, c_4 \gamma_k\})^3}{6 \gamma_k \min\{\Delta_k, \gamma_k / \beta\}} \quad (44)$$

where β is an upper bound on the norm of \mathcal{H}_k . Either, Δ_k is active in the denominator of (44), in which case we have

$$|\rho_k - 1| \leq \frac{(\beta_{L2} + \beta_{\mathcal{H}} \gamma_k) (\min\{\Delta_k, c_4 \gamma_k\})^3}{6 \gamma_k \Delta_k} \leq \frac{(\beta_{L2} + \beta_{\mathcal{H}} \gamma_k) \Delta_k c_4^2 \gamma_k^2}{6 \gamma_k \Delta_k} \leq \frac{(\beta_{L2} + \beta_{\mathcal{H}} \gamma_k) c_4^2}{6} \gamma_k.$$

Or, γ_k / β is active in the denominator of (44), in which case we have

$$|\rho_k - 1| \leq \frac{(\beta_{L2} + \beta_{\mathcal{H}} \gamma_k) (\min\{\Delta_k, c_4 \gamma_k\})^3}{6 \gamma_k^2 / \beta} \leq \frac{(\beta_{L2} + \beta_{\mathcal{H}} \gamma_k) c_4^3 \gamma_k^3}{6 \gamma_k^2 / \beta} \leq \frac{(\beta_{L2} + \beta_{\mathcal{H}} \gamma_k) c_4^3 \beta}{6} \gamma_k.$$

In both cases, since $\lim_{k \rightarrow \infty} \gamma_k = 0$ in view of (38), it follows that $\lim_{k \rightarrow \infty} \rho_k = 1$.

Equation (41) comes from Lemma 4.8.

We prove (42). It follows from Taylor's formula (Lemma 4.6, where the parallel translation becomes the identity since the domain of \hat{f}_{x_k} is the Euclidean space $T_{x_k} M$) that

$$\text{grad } \hat{f}_{x_k}(\xi) = \text{grad } \hat{f}_{x_k}(0_{x_k}) + \text{Hess } \hat{f}_{x_k}(0_{x_k})[\xi] + \int_0^1 \left(\text{Hess } \hat{f}_{x_k}(\tau \xi) - \text{Hess } \hat{f}_{x_k}(0_{x_k}) \right) [\xi] d\tau.$$

The conclusion comes by the Lipschitz condition (36) and the approximation condition (35).

Finally, equation (43) comes from the stopping criterion (11) of the inner iteration. More precisely, the truncated CG loop (Algorithm 2) terminates if either $g(\delta_j, \mathcal{H}_{x_k} \delta_j) \leq 0$, or $\|\eta_{j+1}\| \geq \Delta$, or the criterion (11) is satisfied. Since $\{x_k\}$ converges to v and $\text{Hess } f(v)$ is positive-definite, it follows that \mathcal{H}_{x_k} is positive-definite for all k greater than a certain K . Therefore, for all $k > K$, the criterion $g(\delta_j, \mathcal{H}_{x_k} \delta_j) \leq 0$ is never satisfied. In view of (39) and (40), it can be shown that the trust-region is eventually inactive. Therefore, increasing K if necessary, the criterion $\|\eta_{j+1}\| \geq \Delta$ is never satisfied for all $k > K$. In conclusion, for all $k > K$, the stopping criterion (11) is satisfied each time a computed η_k is returned by the tCG loop. Therefore, the tCG loop behaves as a classical linear CG method; see e.g. [NW99, Section 5.1]. Consequently, $\text{grad } m_{x_k}(\eta_j) = r_j$ for all j . Choose K such that for all $k > K$, $\|\text{grad } f(x_k)\| = \|\text{grad } m_{x_k}(0)\|$ is so small—it converges to zero in view of (38)—that the stopping criterion (11) yields

$$\|\text{grad } m_{x_k}(\eta_j)\| = \|r_j\| \leq \|r_0\|^{1+\theta} = \|\text{grad } m_{x_k}(0)\|^{1+\theta} \text{ or } k \geq d. \quad (45)$$

If the second condition in (45) is active, then it means that the linear CG process has been completed, so $\text{grad } m_{x_k}(\eta_k^j) = 0$, and (43) trivially holds. On the other hand, if the first condition in (45) is active, then we obtain (43) with $c_2 = 1$. □

4.3 Discussion

The main convergence result (Theorem 4.4) shows that RTR-tCG (Algorithm 1-2) converges to a set of stationary points of the cost function for *all* initial conditions. This is an improvement on the pure Newton method, for which only local convergence results exist. However, the convergence theory falls short of showing that the algorithm always converges to a local minimum. This is not surprising: since we have ruled out the possibility of checking positive-definiteness the Hessian of the cost function, we have no way of testing whether a stationary point is a local minimum or not (note as an aside that even checking positive-definiteness of the Hessian is not always sufficient for determining if a stationary point is a local minimum or not: if the Hessian is singular and nonnegative definite, then no conclusion can be drawn). In fact, for the vast majority of optimization methods, only convergence to stationary points can be secured unless some specific assumptions (like convexity) are made; see e.g. [Pol97, Chap. 1].

Nevertheless, it is observed in numerical experiments (Section 6) with random initial conditions that the algorithm systematically converges to a local minimum; convergence to a saddle point is only observed on specifically crafted problems, for example when the iteration is started on a point that is a saddle point in computer arithmetic. This is due to the fact that the algorithm is a descent method, i.e., $f(x_{k+1}) < f(x_k)$ whenever $x_{k+1} \neq x_k$. Therefore, convergence to saddle points or local minima is unstable under perturbations.

Notice moreover that most algorithms which are known to have “global convergence” actually fail to produce a desired solution if the initial condition belongs to some zero-measure set. A classical example is the power method (see [Par80]) which fails to compute the dominant eigenvalue if the initial condition has a vanishing component along the corresponding eigenvector. A similar result for the RTR-tCG algorithm is still an open question: it is not known whether the region of attraction of the saddle points (in exact arithmetic) is a thin set. But since computations are not performed in exact arithmetic and since the numerical noise has a favorable effect on the convergence behaviour, this issue seems to carry little practical interest.

4.3.1 Randomization

To conclude this discussion, we mention the possibility of using a stochastic version of RTR-tCG. The initial condition η^0 in the tCG algorithm is selected from a uniform distribution in some small neighbourhood of 0_{x_k} in $T_{x_k}M$ —this can be thought of as artificial numerical noise. Then the output η of the tCG algorithm is compared against the Cauchy point η^C (which can be computed at little additional cost). If $m_{x_k}(\eta) < m_{x_k}(\eta^C)$, then η is returned; otherwise, η^C is returned.

This stochastic version is interesting both from a theoretical and practical point of view. From a theoretical viewpoint, it yields convergence to a local minimum with probability one for all initial conditions. Indeed, convergence to stationary points still holds since the algorithm improves on the Cauchy point; but accumulation points must be local minima because the instability of the saddle points or local maxima will eventually be revealed with probability one by the random perturbations.

From a practical point of view, the randomized version is efficient in kicking the iteration away from saddle points: due to the trust-region approach, the iterates quickly escape from the saddle points.

The randomization we have described applies to the general RTR-tCG method. Practical applications may lend themselves to other forms of randomization. For example, if a Rayleigh quotient has to be minimized on a Grassmann manifold (see Section 5.3), then a possibility mentioned in [ST00, Section 3.2] is to use Rutishauser’s randomization technique [Rut70]; it consists in appending a random vector to the current subspace, computing the Ritz pairs and discarding the one with largest Ritz value.

5 Applications

In this section we illustrate how the RTR-tCG method (Algorithm 1-2) applies to various practical cases. The problem of computing an extreme eigenspace of a large-scale generalized symmetric/positive-definite eigenproblem (Section 5.3) is particularly interesting; preliminary numerical experiments show that an instance of our RTR-tCG algorithm matches and sometimes dramatically outperforms its competitors.

5.1 Symmetric eigenvalue decomposition

This first example is not expected to yield a competitive algorithm, except possibly under specifically chosen conditions, but its illustrative value makes it worth being considered.

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 $g(A, B) = \text{trace}(A^T B)$ on $\mathbb{R}^{n \times n}$ induces on O_n the metric

$$g_Q(Q\Omega_1, Q\Omega_2) = \text{trace}(\Omega_1^T \Omega_2). \quad (46)$$

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

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

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

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

where A and N are given $n \times n$ symmetric matrices. For $N = \text{diag}(\mu_1, \dots, \mu_n)$, $\mu_1 < \dots < \mu_n$, 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 (46), we obtain

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

from which it follows

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

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.

Note that the formula for the Hessian can be obtained from the formula (7). 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 $\text{Hess } f(Q)[Q\Omega] = Q\text{skew}(\Omega[Q^T Q Q, N] + [\Omega^T Q^T A Q + Q^T A Q \Omega, N])$, which yields the same result as above.

5.2 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

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

Consider the cost function

$$f(U, V) = \text{trace}(U^T AVN)$$

on $O_n \times O_p$, where $N = [\text{diag}(\mu_1, \dots, \mu_p) | 0_{p \times (n-p)}]$, $\mu_1 < \dots < \mu_p < 0$. 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 $\text{skew}(B) = (B - B^T)/2$, we obtain

$$\text{grad } \hat{f}_{(U,V)}(0, 0) = \text{grad } f(U, V) = (U \text{skew}(U^T AVN), -V \text{skew}(NU^T AV)),$$

and

$$\begin{aligned} \text{Hess } \hat{f}_{(U,V)}(0, 0)[(U\Omega_U, V\Omega_V)] &= \text{Hess } f(U, V)[(U\Omega_U, V\Omega_V)] \\ &= (U(\text{skew}(\Omega_U \text{skew}(U^T AVN)) + \text{skew}(\Omega_U^T U^T AVN) + \text{skew}(U^T AV \Omega_V N)) , \\ &\quad -V(\text{skew}(\Omega_V \text{skew}(NU^T AV) + \text{skew}(N \Omega_V^T U^T AV) + \text{skew}(NU^T AV \Omega_V))). \end{aligned}$$

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. Then the pencil (A, B) is said to be symmetric positive definite, abbreviated S/PD [Ste01, Chap. 3, Def. 4.1]. 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 $A\mathcal{Y} \subseteq B\mathcal{Y}$ [Dem00]. 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 $\lambda_1 \leq \dots \leq \lambda_p < \lambda_{p+1} \leq \dots \leq \lambda_n$ be the eigenvalues of the pencil (A, B) . We consider the problem of computing the (unique) eigenspace \mathcal{V} of A associated to the p leftmost eigenvalues (in other words, \mathcal{V} is characterized by $\mathcal{V} = \text{colsp}(V)$ where $AV = V \text{diag}(\lambda_1, \dots, \lambda_p)$ and $V^T V = I$). 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; it is sufficient to replace A by $-A$ throughout and notice that the leftmost eigenspace of $-A$ is the rightmost eigenspace of A .

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

$$f(\text{colsp}(Y)) = \text{trace}((Y^T AY)(Y^T BY)^{-1}) \quad (47)$$

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

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 $\text{Grass}(p, n)$. 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, for example). The first action to take is thus to devise a matrix representation of the elements of $\text{Grass}(p, n)$ 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 for example in Machado and Salavessa [MS85]. Another possibility is to rely on the definition of $\text{Grass}(p, n)$ as a quotient of Lie groups; we refer to [EAS98] and references therein for details. Yet another possibility is to rely on coordinate charts on Grassmann (see, e.g., [HM94, Section C4]); this approach is appealing because it uses a minimal set of variables, but it has the drawback of relying on arbitrarily fixed reference points.

A fourth way, which we will follow here, is to consider $\text{Grass}(p, n)$ 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]; we refer to this paper for all details. 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.

In this approach, a tangent vector ξ to $\text{Grass}(p, n)$ at a point $\mathcal{Y} = \text{colsp}(Y)$ is represented by a *horizontal lift* $\xi_{\circ Y}$ as follows. First, given a real function h on $\text{Grass}(p, n)$, define its lift by $h_{\circ Y} = h(\text{colsp}(Y))$. Then the horizontal lift of ξ is uniquely defined by the following two conditions: (i) $Y^T \xi_{\circ Y} = 0$ and (ii) $Dh(\mathcal{Y})[\xi] = Dh_{\circ}(Y)[\xi_{\circ Y}]$ for all real functions h on $\text{Grass}(p, n)$. Therefore, the *horizontal space*

$$H_Y = \{Z \in \mathbb{R}^{n \times p} : Y^T Z = 0\} \quad (48)$$

represents the tangent space $T_{\mathcal{Y}}\text{Grass}(p, n)$.

The canonical metric on $\text{Grass}(p, n)$ is then defined by [AMS04, Section 3.3]

$$g_{\mathcal{Y}}(\xi, \zeta) = \text{trace}((Y^T Y)^{-1} \xi_{\circ Y}^T \zeta_{\circ Y}). \quad (49)$$

The simplest possible retraction is arguably

$$R_{\mathcal{Y}}\xi = \text{colsp}(Y + \xi_{\circ Y}). \quad (50)$$

The following proposition shows that (50) is a well-defined retraction that agrees with the Riemannian exponential up to the second order. Therefore, the “zero acceleration” condition (34) is satisfied by (50), and thus $\text{Hess } f(\mathcal{Y}) = \text{Hess } \hat{f}_{\mathcal{Y}}(0)$, where (slightly abusing notations by failing to distinguish tangent vectors and their representations)

$$\hat{f}_{\mathcal{Y}}(Z) = f(R_{\mathcal{Y}}(Z)) = f(\text{colsp}(Y + Z)), \quad Y^T Z = 0 \quad (51)$$

with $f(\text{colsp}(Y))$ as in (47).

Proposition 5.1 *Consider the Grassmann manifold $\text{Grass}(p, n)$ endowed with its canonical metric (49). Let \mathcal{Y} be in $\text{Grass}(p, n)$ and η be in $T_{\mathcal{Y}}\text{Grass}(p, n)$. Then $\text{Exp}_{\mathcal{Y}}\eta = R_{\mathcal{Y}}(\eta + E(\eta))$, where R is as in (50) and $\|E(\eta)\| \leq c\|\eta\|^3$ for some c and all η sufficiently small.*

Proof. Pick Y such that $\mathcal{Y} = \text{colsp}(Y)$ and $Y^T Y = I$. Let $\eta_{\circ Y} = U\Sigma V^T$ be a singular value decomposition. Then [AMS04, Thm. 3.6]

$$\begin{aligned} \text{Exp}_{\mathcal{Y}}\eta &= \text{colsp}(Y + \eta_{\circ Y} V \Sigma^{-1} \tan \Sigma V^T) \\ &= \text{colsp}(Y + \eta_{\circ Y} + \eta_{\circ Y} O(\Sigma^2)) \\ &= \text{colsp}(Y + \eta_{\circ Y} + O(\|\eta_{\circ Y}\|^3)). \end{aligned}$$

□

We then need formulas for the gradient and the Hessian of the Rayleigh cost function (47). Using the formulas in [AMS04, Section 4.3], we obtain

$$\frac{1}{2}(\text{grad } f)_{\circ Y} = P_{BY,Y}AY(Y^TBY)^{-1}Y^TY, \quad (52)$$

where

$$P_{U,V} = I - U(V^TU)^{-1}V^T \quad (53)$$

denotes the projector parallel to the span of U onto the orthogonal complement of the span of V . Then, using the material in [AMS04, Section 3.5] and the formula $\text{Hess } f[\xi] = \nabla_{\xi} \text{grad } f$ [dC92, Chap. 6, Ex. 11], we obtain

$$\begin{aligned} \frac{1}{2}(\text{Hess } f(\mathcal{Y})[\xi])_{\circ Y} &= \frac{1}{2}P_{Y,Y}D(\text{grad } f)_{\circ}(Y)[\xi_{\circ Y}] \\ &= P_{BY,Y} (AP_{Y,BY}\xi_{\circ Y}(Y^TBY)^{-1}Y^TY - B\xi_{\circ Y}(Y^TBY)^{-1}Y^TAY(Y^TBY)^{-1}Y^TY) \\ &\quad - P_{Y,Y}BY(Y^TBY)^{-1}\xi_{\circ Y}^T \frac{1}{2}(\text{grad } f)_{\circ Y} - \frac{1}{2}(\text{grad } f)_{\circ Y}(Y^TY)^{-1}\xi_{\circ Y}^T BY(Y^TBY)^{-1}Y^TY. \end{aligned} \quad (54)$$

Since $\text{Hess } f(\mathcal{Y}) = \text{Hess } \hat{f}_{\mathcal{Y}}(0_{\mathcal{Y}})$, this expression could also have been obtained by considering the second order term the Taylor expansion of $\hat{f}_{\mathcal{Y}}$ around 0.

We drop the terms in (54) involving $\text{grad } f$ to obtain

$$\mathcal{H}_{\circ Y}[\xi_{\circ Y}] = P_{BY,Y} (AP_{Y,BY}\xi_{\circ Y}(Y^TBY)^{-1}Y^TY - B\xi_{\circ Y}(Y^TBY)^{-1}Y^TAY(Y^TBY)^{-1}Y^TY). \quad (55)$$

It follows that the approximation condition (35) is satisfied. Note that for $B = I$ we recover the formulas given in [EAS98, Section 4.8].

We have now all the necessary elements at hand to apply the RTR-tCG algorithm to the function f defined in (47). Since f is smooth and the Grassmann manifold is compact, it follows that the gradient and the Hessian of f are bounded. Consequently, all the hypotheses of Proposition 4.5 hold, and therefore the algorithm converges for all initial conditions to a set of stationary points of f (Theorem 4.4). It can be shown (as a simple generalization of [AMSV02, Prop. 4.1]) that the stationary points of f are the eigenspaces of (A, B) . However, only the leftmost eigenspace \mathcal{V} (which is unique in view of our assumption that $\lambda_p \neq \lambda_{p+1}$) is numerically stable, so convergence to \mathcal{V} is expected to occur in practice; with the randomized version of the algorithm described in Section 4.3, convergence to \mathcal{V} occurs with probability one. Moreover, since \mathcal{V} is a nondegenerate local minimum, it follows that the rate of convergence is $\min\{\theta + 1, 2\}$, where θ is the parameter appearing in the stopping criterion (11) of the inner (tCG) iteration.

Next we show how the RTR-tCG algorithm for the Rayleigh quotient (47) connects with other eigenvalue algorithms. In particular, we show connections with an Krylov-type algorithm recently proposed by Golub and Ye [GY02] and with the Tracemin algorithm of Sameh and Wisniewski [SW82, ST00].

5.3.1 Grassmann manifold with noncanonical metric

To make the connections clearer, it is useful to abandon the canonical metric (49) on the Grassmann manifold and use instead a noncanonical metric defined below. A reward is that the expression of the exact $\text{Hess } \hat{f}_{\mathcal{Y}}(0)$ becomes simpler than (54).

First, redefine the horizontal space as

$$H_Y = \{Z \in \mathbb{R}^{n \times p} : Y^T BZ = 0\}. \quad (56)$$

This yields a new definition of horizontal lift. The horizontal lift of ξ , which we now denote by $\xi_{\uparrow Y}$ is uniquely defined by the following two conditions: (i) $Y^T B\xi_{\uparrow Y} = 0$ and (ii) $Dh(\mathcal{Y})[\xi] = Dh \circ \text{colsp}(Y)[\xi_{\uparrow Y}]$ for all real functions h on $\text{Grass}(p, n)$. The only difference with the definitions of $\xi_{\circlearrowleft Y}$ and $\xi_{\uparrow Y}$ is the presence of B in point (i).

We then define a noncanonical metric on $\text{Grass}(p, n)$ as

$$g_{\mathcal{Y}}(\xi, \zeta) = \text{trace} \left((Y^T B Y)^{-1} \xi_{\uparrow Y}^T \zeta_{\uparrow Y} \right). \quad (57)$$

From now on, the definitions of the gradient, Hessian and Riemannian connection will be with respect to the metric (57). We also define a new retraction

$$R_{\mathcal{Y}}(\xi) = \text{colsp}(Y + \xi_{\uparrow Y}) \quad (58)$$

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

Using the same Rayleigh cost function (47) as before, we obtain

$$\begin{aligned} \hat{f}_{\mathcal{Y}}(\xi) &= f(R_{\mathcal{Y}}(\xi)) = \text{trace} \left(\left((Y + \xi_{\uparrow Y})^T B (Y + \xi_{\uparrow Y}) \right)^{-1} \left((Y + \xi_{\uparrow Y})^T A (Y + \xi_{\uparrow Y}) \right) \right) \\ &= \text{trace} \left((Y^T B Y)^{-1} Y^T A Y \right) + 2 \text{trace} \left((Y^T B Y)^{-1} \xi_{\uparrow Y}^T A Y \right) \\ &\quad + \text{trace} \left((Y^T B Y)^{-1} \xi_{\uparrow Y}^T (A \xi_{\uparrow Y} - B \xi_{\uparrow Y} (Y^T A Y)) \right) + \text{HOT} \\ &= \text{trace} \left((Y^T B Y)^{-1} Y^T A Y \right) + 2 \text{trace} \left((Y^T B Y)^{-1} \xi_{\uparrow Y}^T P_{BY, BY} A Y \right) \\ &\quad + \text{trace} \left((Y^T B Y)^{-1} \xi_{\uparrow Y}^T P_{BY, BY} (A \xi_{\uparrow Y} - B \xi_{\uparrow Y} (Y^T A Y)) \right) + \text{HOT}, \end{aligned} \quad (59)$$

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 (57), we obtain

$$(\text{grad } f(\mathcal{Y}))_{\uparrow Y} = \left(\text{grad } \hat{f}_{\mathcal{Y}}(0) \right)_{\uparrow Y} = 2P_{BY, BY} A Y \quad (60)$$

and

$$\left(\text{Hess } \hat{f}_{\mathcal{Y}}(0_{\mathcal{Y}})[\xi] \right)_{\uparrow Y} = 2P_{BY, BY} (A \xi_{\uparrow Y} - B \xi_{\uparrow Y} (Y^T A Y)). \quad (61)$$

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

We choose to take

$$\mathcal{H}_{\mathcal{Y}} = \text{Hess } \hat{f}_{\mathcal{Y}}(0_{\mathcal{Y}}). \quad (62)$$

Therefore, the approximation condition (35) is trivially satisfied. The model (8) is thus

$$\begin{aligned} m_{\mathcal{Y}}(\xi) &= f(\mathcal{Y}) + g_{\mathcal{Y}}(\text{grad } f(\mathcal{Y}), \xi) + \frac{1}{2} g_{\mathcal{Y}}(\mathcal{H}_{\mathcal{Y}} \xi, \xi) \\ &= \text{trace} \left((Y^T B Y)^{-1} Y^T A Y \right) + 2 \text{trace} \left((Y^T B Y)^{-1} \xi_{\uparrow Y}^T A Y \right) \\ &\quad + \text{trace} \left((Y^T B Y)^{-1} \xi_{\uparrow Y}^T (A \xi_{\uparrow Y} - B \xi_{\uparrow Y} (Y^T B Y)^{-1} Y^T A Y) \right). \end{aligned} \quad (63)$$

Since the Rayleigh cost function (47) is smooth on $\text{Grass}(p, n)$ —recall that B is positive definite—and since $\text{Grass}(p, n)$ 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 (37). (Since the metric (57) is different from the canonical metric (49), the formulas given in [AMS04] do not apply.) But since B is fixed and positive definite, the distances induced by the noncanonical metric (57) and by the canonical metric (49) are asymptotically equivalent, and therefore for a given sequence both distances yield the same rate of convergence.

We have now all the required information to use the RTR-tCG method (Algorithm 1-2) for the Rayleigh cost function (47) on the Grassmann manifold $\text{Grass}(p, n)$ endowed with the noncanonical metric (57). This yields the following practical version of the inner iteration. (We omit the horizontal lift notation for conciseness.) We use the notation

$$\overline{\mathcal{H}}_Y[Z] = P_{BY, BY}(AZ - BZ(Y^T BY)^{-1}Y^T AY). \quad (64)$$

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 $j = 0, 1, 2, \dots$ *until a stopping criterion is satisfied, perform the iteration:*

if $\text{trace}(\delta_j^T \overline{\mathcal{H}}_Y[\delta_j]) \leq 0$

Compute $\tau > 0$ such that $\eta = \eta^j + \tau \delta_j$

satisfies $\text{trace}(\eta^T \eta) = \Delta$;

return η ;

Set $\alpha_j = \text{trace}(r_j^T r_j) / \text{trace}(\delta_j^T \overline{\mathcal{H}}_Y[\delta_j])$;

Set $\eta^{j+1} = \eta^j + \alpha_j \delta_j$;

if $\text{trace}((\eta^{j+1})^T \eta^{j+1}) \geq \Delta$

Compute $\tau \geq 0$ such that $\eta = \eta^j + \tau \delta_j$ satisfies $\text{trace}(\eta^T \eta) = \Delta$;

return η ;

Set $r_{j+1} = r_j + \alpha \overline{\mathcal{H}}_Y[\delta_j]$;

Set $\beta_{j+1} = \text{trace}(r_{j+1}^T r_{j+1}) / \text{trace}(r_j^T r_j)$;

Set $\delta_{j+1} = -r_{j+1} + \beta_{j+1} \delta_j$;

end (for).

According to the formula (58) for the retraction, the returned η yields a candidate new iterate Y_+ by the formula

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

where M is chosen such that $Y_+^T BY_+ = I$. 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 (63) and \hat{f} as in (59).

The resulting algorithm converges to eigenspaces of (A, B) —which are the stationary points of the cost function (47)—, 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 minimum (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 (11) of the inner (tCG)

iteration. Note that the cost function has the symmetry property $\hat{f}_\nu(\xi) = \hat{f}_\nu(-\xi)$; this suggests that the rate of convergence is $\min\{\theta + 1, 3\}$, which is compatible with the results observed in numerical experiments.

5.3.2 General connection with Krylov methods

We start with a well-known property of the conjugate gradient algorithm. Consider the general RTR-tCG method (Algorithm 1-2). Let x be the current iterate and let $\{\eta^j\}_{j=0,\dots,m}$ be the sequence constructed by the inner iteration (Algorithm 2). Then, for all $j \in \{0, 1, \dots, m\}$,

$$\eta^j = \arg \min_{\eta \in K_j(\mathcal{H}_x, \text{grad } f(x))} m_x(\eta)$$

where m_x is as in (8) and

$$K_j(\mathcal{H}, \xi) \equiv \text{lin} \{ \xi, \mathcal{H}[\xi], \mathcal{H}[\mathcal{H}[\xi]], \dots, \mathcal{H}^j[\xi] \}, \quad (65)$$

where “lin { }” is the set of linear combinations of its arguments. We deliberately avoid using the classical name “span” because it is often used to denote the column space of a matrix argument, and it is essential here to make the distinction between the two concepts.

We now consider the case of the Rayleigh cost function (47). Let $\{\eta^j\}_{j=0,\dots,m}$ be a sequence of $n \times p$ matrices generated by the practical tCG (Algorithm 3). First notice that $m_Y(\eta)$, given in (63), is a second-order approximation of

$$\hat{f}_Y(\eta) = f(R_Y(\eta)) = f(\text{colsp}(Y + \eta_{\uparrow Y})) = f_{\uparrow}(Y + \eta_{\uparrow Y})$$

Therefore, $Y + \eta^j$ minimizes a second-order approximation of f_{\uparrow} over the space

$$Y + K_{j-1}(\overline{\mathcal{H}}_Y, r_0) = Y + \text{lin} \left\{ r_0, \overline{\mathcal{H}}_Y[r_0], \dots, \overline{\mathcal{H}}_Y^{j-1}[r_0] \right\}. \quad (66)$$

The connections with Krylov subspace methods relies on the following result.

Proposition 5.2 *Define*

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

as in (64) and

$$r_0 = P_{BY, BY} AY$$

as in Algorithm 3. Given a linear operator \mathcal{O} from $\mathbb{R}^{n \times p}$ to itself, define

$$K_j(\mathcal{O}, Z) = \text{lin} \{ Z, \mathcal{O}[Z], \dots, \mathcal{O}^j[Z] \} = \left\{ \sum_{i=0}^j \alpha_i \mathcal{O}^i Z, \alpha_i \in \mathbb{R} \right\}$$

and

$$\mathcal{K}_j(\mathcal{O}, Z) = \text{colsp}(Z, \mathcal{O}[Z], \dots, \mathcal{O}^j[Z]).$$

(Observe that K yields a linear subspace of $\mathbb{R}^{n \times p}$ while \mathcal{K} yields a linear subspace of \mathbb{R}^n .)

Then, for all i in $\{1, \dots, p\}$,

$$(Y + K_{j-1}(\overline{\mathcal{H}}_Y, r_0)) e_i \subset \mathcal{K}_j(\overline{\mathcal{H}}_Y, Y).$$

Proof. The inclusion $(Y + K_{j-1}(\overline{\mathcal{H}}_Y, r_0))e_i \subset \text{colsp}(Y, \mathcal{K}_{j-1}(\overline{\mathcal{H}}_Y, r_0))$ is direct. The equality $\text{colsp}(Y, \mathcal{K}_{j-1}(\overline{\mathcal{H}}_Y, r_0)) = \mathcal{K}_j(\overline{\mathcal{H}}_Y, Y)$ is direct because $r_0 = \overline{\mathcal{H}}_Y[Y]$. \square

This shows that the search space $(Y + K_{j-1}(\overline{\mathcal{H}}_Y, r_0))$ of the tCG method (Algorithm 3) is a subspace of the “generalized” Krylov space $(\mathcal{K}_j(\overline{\mathcal{H}}_Y, Y))^p$. We use the term “generalized” because the operator $\overline{\mathcal{H}}_Y$, instead of being a simple matrix multiplication on the left as usually, is a Sylvester operator. It is not clear how such a Krylov space can be reliably built, since it is not invariant by reorthogonalizations during the process.

However, there are (at least) two cases where $\mathcal{K}_j(\overline{\mathcal{H}}_Y, Y)$ is equal to a classical Krylov space $\mathcal{K}_j(C, Y) = \text{colsp}(Y, CY, \dots, C^j Y)$ for some matrix C : (i) the case $p = 1$, where the matrix $(Y^T B Y)^{-1} Y^T A Y$ becomes a scalar and can thus be put on the left, and (ii) the case $B = I$. We investigate them next.

5.3.3 Connection with Golub and Ye’s inverse-free Krylov method for (A, B)

We first consider the case $p = 1$. Then the generalized Krylov subspace $\mathcal{K}_j(\overline{\mathcal{H}}_Y, Y)$ appearing in Proposition 5.2 becomes

$$\mathcal{K}_j \left(P_{By, By} \left(A - \frac{y^T A Y}{y^T B y} B \right), y \right).$$

The distinction between K and \mathcal{K} becomes irrelevant when $p = 1$, and it is possible to prove a stronger version of Proposition 5.2, namely

$$(y + K_{j-1}(\overline{\mathcal{H}}_y, r_0))\mathbb{R} = \mathcal{K}_j(\overline{\mathcal{H}}_y, y) \setminus \{z : y^T B z = 0\}.$$

This establishes a connection with the “Inverse free Krylov subspace method for (A, B) ” (IFKS) proposed by Golub and Ye [GY02, Alg. 1]. The Krylov subspace used in IFKS is

$$\mathcal{K}_m \left(A - \frac{y^T A y}{y^T B y} B, y \right),$$

which is closely related to the one used by our tCG method. Another difference is that Alg. 1 in [GY02] restarts with the minimizer of the *exact* Rayleigh quotient in the subspace $\mathcal{K}_m \left(A - \frac{y^T A y}{y^T B y} B, y \right)$, while our tCG method minimizes the quadratic model (63).

In spite of this, RTR-tCG often clearly outperforms IFKS, because of the following two major advantages: (i) the CG approach keeps the storage space minimal, which is important when the problem is of very large dimension; (ii) the truncated CG scheme provides a stopping criterion for the inner iteration, producing an adaptive m that yields superlinear convergence.

It should be theoretically possible to use a stopping criterion for the inner iteration of IFKS, but this is impractical since the leftmost Ritz vector of the pencil of reduced matrices would need to be computed at each inner step; this operation would be expensive since the reduced pencil does not have a triangular structure. An alternative was proposed in Scott [Sco81] where the leftmost Ritz pair is computed for the reduced matrix $A - \frac{y^T A y}{y^T B y} B$ instead of the reduced pencil $\left(A - \frac{y^T A y}{y^T B y} B, B \right)$. Scott proposes heuristics for stopping the inner iteration that “seem to work well in practice”. Superlinear convergence is not proven, in contrast with the proven superlinear convergence of RTR-tCG guaranteed by Theorem 4.12.

It is interesting that Proposition 5.2 points to a block version of IFKS, where the operation $x \mapsto (A - \frac{y^T A Y}{y^T B y} B)x$ used to expand the Krylov space would be replaced by $X \mapsto AX - X(Y^T A Y)^{-1}(Y^T A Y)$. It is not clear how this could be efficiently carried out in practice.

5.3.4 Connection with restarted block Lanczos

We now consider the case $B = I$ and p arbitrary. For this situation, relations between the conjugate gradient and (block) Lanczos algorithms were investigated in [Cul78].

Golub and Underwood [GU77] (see also Cullum and Donath [CD74] for the maximization version) proposed a block version of the Lanczos algorithm, which we call Ritz-Restarted Block Lanczos (RRBL). This algorithm induces a subspace iteration (i.e., an iteration on $\text{Grass}(p, n)$) that can be written as follows.

Algorithm 4 (RRBL) *Data: symmetric $n \times n$ matrix A .*

Parameter: length m of the Krylov sequences.

Input: Current iterate $\mathcal{Y} \in \text{Grass}(p, n)$.

Output: New iterate $\mathcal{Y}_+ \in \text{Grass}(p, n)$.

(a) *Pick an orthonormal $n \times p$ matrix Y that spans \mathcal{Y} .*

(b) *Generate an orthonormal basis Q for the Krylov space $\mathcal{K} = \text{colsp}(Y, AY, \dots, A^{m-1}Y)$.*

(c) *Compute the matrix Rayleigh quotient $M = Q^T A Q$ which represents the projection of A into \mathcal{K} .*

(d) *Compute X , an orthonormal basis for the p -dimensional dominated eigenspace of M .*

(e) *Define $\mathcal{Y}_+ = \text{colsp}(QX)$.*

It is shown in [Cul78, Lemma 1] that the subspaces

$$S_j = \text{colsp}(Y, AY, \dots, A^j Y)$$

and

$$S'_j = \text{colsp}(Y, r_0, Ar_0, \dots, A^{j-1}r_0)$$

are identical, where

$$r_0 = P_{Y,Y} A Y.$$

Moreover, assuming the normalization $Y^T Y = I$, the expression (64) becomes

$$\bar{\mathcal{H}}_Y[Z] = P_{Y,Y}(AZ - Z(Y^T A Y)),$$

from which it can be deduced that

$$S'_j = \text{colsp}(Y, r_0, \bar{\mathcal{H}}_Y[r_0], \dots, \bar{\mathcal{H}}_Y^{j-1}[r_0])$$

However, RRBL and RTR-tCG for (47) are not equivalent. RRBL restarts by minimizing the Rayleigh quotient $\tilde{Y} \mapsto \text{trace}((\tilde{Y}^T A \tilde{Y})(\tilde{Y}^T \tilde{Y})^{-1})$ under the constraint that the (linearly independent) columns of \tilde{Y} belong to S_m . Instead, RTR-tCG minimizes the *model*

$$m_{\diamond Y}(\eta_{\diamond Y}) = \text{trace}((Y^T Y)^{-1} \eta_{\diamond Y}^T (\text{grad } f)_{\diamond Y}) + \frac{1}{2} \text{trace}((Y^T Y)^{-1} \eta_{\diamond Y}^T \mathcal{H}_{\diamond Y}[\eta_{\diamond Y}])$$

under the stronger constraints that $Y + \eta_{\diamond Y}$ belongs to

$$\text{lin} \{ Y, (\text{grad } f)_{\diamond Y}, \mathcal{H}_{\diamond Y}(\text{grad } f)_{\diamond Y}, \dots, \mathcal{H}_{\diamond Y}^{m-1}(\text{grad } f)_{\diamond Y} \}$$

where $\text{lin} \{ \}$ denotes the linear envelope of the arguments, that is, $\text{lin} \{ Z_1, \dots, Z_q \} = \{ \sum_{i=1}^q \alpha_i Z_i : \alpha_1, \dots, \alpha_q \in \mathbb{R} \}$.

Since RTR-tCG only uses a model of the cost and since its search space is more restricted, it is tempting to conclude that RRBL does at least as well as RTR-tCG. But in fact, this property only holds over one iteration. Namely, if the two methods start their *inner* iteration on the *same point*, then the *next* (outer) iterate of RRBL will have a smaller (i.e., better) value of the Rayleigh quotient cost function than RTR-tCG. This does not rule out the possibility that the whole sequence of outer iterates generated by RTR-tCG converge faster than RRBL.

Moreover, RTR-tCG has two important advantages over RRBL. First, the necessary storage space remains small, whereas RRBL has to store a complete basis of the Krylov subspace S_m , for a total of mp vectors of size n . Second, the RTR-tCG uses an adaptive m : the tCG framework naturally provides criteria for stopping the inner iteration; global and superlinear convergence are guaranteed by the convergence theory developed in Section 4.

Note that the concept of an adaptive Krylov length m has been used before for the same problem by Morgan and Scott [MS93]. Their algorithm, however, is plagued by the necessity of storing a Krylov basis; since m is expected to grow large as the outer iteration comes near to the solution, the necessary storage space may become prohibitively large. The RTR-tCG method does not suffer from this drawback.

5.3.5 Connection with Tracemin

The trace minimization method proposed by Sameh and Wisniewski [SW82, ST00] is constructed as follows. From the current B -orthonormal basis Y , define the next iterate as $Y + \Delta$ where Δ is the solution of

$$\arg \min_{Y^T B \Delta = 0} \text{trace}(Y + \Delta)^T A(Y + \Delta), \quad (67)$$

where A is also assumed to be positive definite (i.e., $A \succ 0$ and $B \succ 0$). It is shown in [SW82, ST00] that the solution of (67) is obtained by solving

$$(P_{BY, BY} A P_{BY, BY}) \Delta = P_{BY, BY} A Y, \quad Y^T B \Delta = 0,$$

with P defined as in (53); a CG process is studied for solving this equation and a stopping criterion is proposed.

It is interesting to notice that the function $(Y + \Delta)^T A(Y + \Delta)$ involved in (67) is *not* the second-order development of the Rayleigh quotient (47) using the B -horizontal representation. Instead, the correct second order development is given by (63) (in which $\xi_{\uparrow Y}$ replaces Δ). Assuming that Y is normalized such that $Y^T B Y = I$, and removing the redundant occurrences of $P_{BY, BY}$, the expression (63) simplifies to

$$\begin{aligned} f(\text{colsp}(Y)) + 2\text{trace}(\xi_{\uparrow Y}^T A Y) + \text{trace}(\xi_{\uparrow Y}^T (A \xi_{\uparrow Y} - B \xi_{\uparrow Y} Y^T A Y)) \\ = \text{trace}(Y + \xi_{\uparrow Y})^T A(Y + \xi_{\uparrow Y}) - \text{trace}(\xi_{\uparrow Y}^T B \xi_{\uparrow Y} Y^T A Y), \end{aligned} \quad (68)$$

where $\xi_{\uparrow Y}$ is constrained by $Y^T B \xi_{\uparrow Y} = 0$. Note that the functions in (67) and (68) only differ by the term $\text{trace}(\xi_{\uparrow Y}^T B \xi_{\uparrow Y} Y^T A Y)$.

The problems of minimizing (67) or (68) are quite different in nature. In the former case, it is shown in [SW82, ST00] that the only stationary point $(Y + \Delta)$ is a minimum and, after B -orthonormalization, it yields a Y_+ that satisfies $\text{trace}(Y_+^T A Y_+) \leq \text{trace}(Y^T A Y)$. In the latter case, the only stationary point $(Y + \xi_{\uparrow Y})$ of the model can be viewed as the solution of a *pure* Newton

step; there is thus no guarantee that the cost, or even the quadratic approximation, is decreasing. A way to overcome this difficulty is to use a truncated CG method for approximating the minimizer of the model within some trust region. This is precisely what our RTR-tCG algorithm does.

Tracemin and RTR-tCG both have excellent global convergence properties. For both iterations, unstable convergence cannot be ruled out but is not expected to occur in practice, and can be prevented by a randomization technique.

However, the two methods differ by their rate of convergence. Because of the “missing term” in (67), the basic Tracemin algorithm only achieves linear convergence, and the authors propose heuristics in the form of dynamic shifts to speed up the convergence; the workings of the heuristics are not yet understood rigorously. In contrast, thanks to the utilization of the exact quadratic model, the RTR-tCG algorithm achieves superlinear convergence.

5.3.6 Connections with Newton methods

Several methods are related with our RTR-tCG algorithm in the sense that they attempt to find the stationary point of the model (63), which is the solution of the “Newton” equation

$$P_{BY,BY}(A\xi_{\uparrow Y} - B\xi_{\uparrow Y}(Y^T BY)^{-1}Y^T AY) = -P_{BY,BY}AY, \quad Y^T B\xi_{\uparrow Y} = 0, \quad (69)$$

where $P_{U,V} = I - U(V^T U)^{-1}V^T$.

The connections between these methods are often hidden because (69) may take several equivalent forms. For example, (69) is equivalent to

$$P_{BY,Y}(A\xi_{\uparrow Y} - B\xi_{\uparrow Y}(Y^T BY)^{-1}Y^T AY) = -P_{BY,Y}AY, \quad Y^T B\xi_{\uparrow Y} = 0.$$

It is common to choose a Y that diagonalizes the block Rayleigh quotient, that is

$$(Y^T BY)^{-1}Y^T AY = \text{diag}(\rho_1, \dots, \rho_p). \quad (70)$$

Then, with the notations

$$\overline{\xi_{\uparrow Y}} = [\delta_1, \dots, \delta_p], \quad Y = [y_1, \dots, y_p],$$

the Newton equation (69) becomes

$$P_{BY,BY}((A - \rho_i B)\delta_i + Ay_i) = 0, \quad Y^T B\delta_i = 0, \quad i = 1, \dots, p. \quad (71)$$

Equation (69) is also equivalent to the following structured linear system

$$\begin{bmatrix} A - \rho_i B & BY \\ Y^T B & 0 \end{bmatrix} \begin{bmatrix} \delta_i \\ \ell_i \end{bmatrix} = \begin{bmatrix} -(A - \rho_i B)y_i \\ 0 \end{bmatrix}, \quad i = 1, \dots, p. \quad (72)$$

Pick $Y \in \text{colsp}^{-1}(Y)$ and $U \in \text{ST}(n-p, n)$ such that $(Y|U)^T B(Y|U) = I_n$. Then (69) is equivalent to

$$U^T AUK - KY^T AY = -U^T AY, \quad \xi_{\uparrow Y} = UK. \quad (73)$$

For the numerical computation of U , we refer e.g. to section 16.2 in [NW99]. Note that sparsity of A is in general not preserved in the matrix $U^T AU$ of size $(n-p) \times (n-p)$; one will therefore attempt to avoid explicitly forming the product $U^T AU$.

These equivalent formulations highlight connections with several methods, which differ for example by variations on the choice of the shifts ρ and the way the δ 's are utilized to obtain the new iterate; see for example [PW79, Cha84, Shu86, Dem87, Smi94, Smi97, EAS98, LST98, Fat98, LE02, ADM⁺02, MA03, SE02]. Note that the formulation (72) relates to saddle point problems for which a vast literature exists; see [SS98, GG03] and references therein.

5.3.7 Other connections

Closely related to Algorithm 3 is the CG-3 method mentioned in [EAS98, Section 3.5.1]. Instead of staying in the same tangent space during the inner iteration, CG-3 directly applies the update vector to obtain a new point on the Grassmann manifold. The current search direction is carried along using parallel transport, the new gradient is computed, and the new search direction is chosen using the Polak-Ribière or Fletcher-Reeves formula. There is no notion of trust-region or truncated CG in [EAS98], since global convergence issues were not considered.

Finally, we mention that recently proposed algorithms by Nikpour *et al.* [NMMA04] and Monneau and Torik [MT99] display properties similar to those of the truncated CG approach presented here, namely, good global and local convergence without the need of factorizing A or B . The methodology, however, is different. Instead of minimizing the classical Rayleigh quotient on the Grassmann manifold, the methods in [NMMA04, MT99] attempt to find a minimizer in the Euclidean space $\mathbb{R}^{n \times p}$ of a modified Rayleigh quotient; the modification is specifically chosen such that the minimizer carries useful information about the desired extreme eigenspace.

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 Matlab implementations of the RTR algorithm. The applications tested include the symmetric eigenvalue decomposition (Section 5.1)(EVD-RTR), the singular value decomposition (Section 5.2)(SVD-RTR), and the computation of an extreme eigenspace of a symmetric definite matrix pencil (Section 5.3). The subsections that follow demonstrate the convergence properties of these algorithms, in addition to a competitive test of the numerical efficiency of the extreme generalized eigenvalue algorithm.

6.1 Full Singular Value and Eigenvalue Decompositions

The SVD-RTR algorithm as outlined in Section 5.2 was tested to illustrate the convergence properties of the method. The matrix A used was a 100×40 matrix with elements randomly selected from a normal distribution. The left and right bases U and V were initialized by generating random matrices of the appropriate 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 2 shows the error in the computed singular values at each iteration of the algorithm. As the SVD-RTR algorithm 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 $\hat{\Sigma} = U^T A V$. 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.

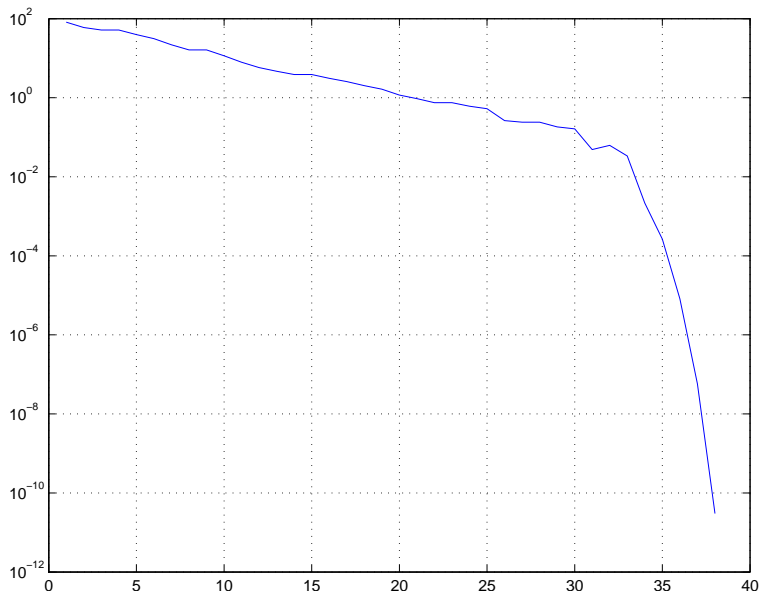


Figure 2: Illustration of the convergence for the SVD-RTR experiment. The vertical axis gives the measure $\|\hat{\Sigma} - \Sigma\|_F$ and the horizontal axis indicates the number of iterations of Algorithm 1.

Similar numerical experiments were performed on the EVD-RTR algorithm (Section 5.1), using a randomly generated, symmetric matrix A . As for the SVD-RTR, the initial eigenvectors were chosen by orthogonalizing a $n \times n$ matrix with normal random entries. Similar results were obtained from this experiment. Table 1 shows the evolution of the error for a sample problem. The error is measured by subtracting the current eigenvector estimates from the actual eigenvectors of the matrix A , adjusting the sign where necessary. As with the SVD-RTR, these results clearly illustrate a superlinear rate of convergence for the EVD-RTR algorithm.

6.2 Extreme Eigenspace Computation

We also performed numerical experiments using a Matlab implementation of the algorithm for computing the extreme eigenspace of a symmetric definite matrix pencil (Algorithm 3). Two sets of experiments were run. The first set of experiments examines the convergence properties of Algorithm 3. The second set of experiments seeks to compare the numerical efficiency of Algorithm 3 against that of competing methods.

6.2.1 Convergence Properties

For the first set of tests, the convergence of Algorithm 3 is compared against that of the basic trace minimization algorithm (Tracemin) of Sameh and Tong [ST00]. Also compared is the Rayleigh Quotient Iteration (RQI) (see [Par80]).

Iteration	$\ \hat{Q} - Q\ _F$
1	3.6261e+00
2	3.5190e+00
3	3.3669e+00
4	3.1517e+00
5	2.9403e+00
6	2.6889e+00
7	2.3367e+00
8	2.3367e+00
9	2.1774e+00
10	2.0716e+00
11	2.0145e+00
12	1.6337e+00
13	1.6337e+00
14	1.3697e+00
15	7.1374e-01
16	4.4062e-01
17	6.8965e-02
18	3.4756e-04
19	7.6915e-08
20	1.5271e-11

Table 1: Table illustrating convergence for the EVD-RTR experiment.

Figures 3 and 4 show the results of these experiments. The matrix B was chosen as the identity, while the matrix A was chosen as $A = SS^T$, where S was an $n \times n$ matrix with elements randomly chosen from a normal distribution. The experiment was run with $n = 100$, computing only the first ($p = 1$) eigenvector of A . The θ parameter in the inner stopping criterion 11 was set to $\theta = 2.0$, striving for a cubic rate of convergence.

Two experiments were run with this setup. In the first (Figure 3), the initial iterate for the algorithms was chosen close to the leftmost eigenvector of A , to illustrate the local convergence of the algorithms. The Tracemin algorithm experiences a linear convergence rate, as is expected. Note that the results for both the RQI and RTR algorithms are compatible with a cubic rate of convergence.

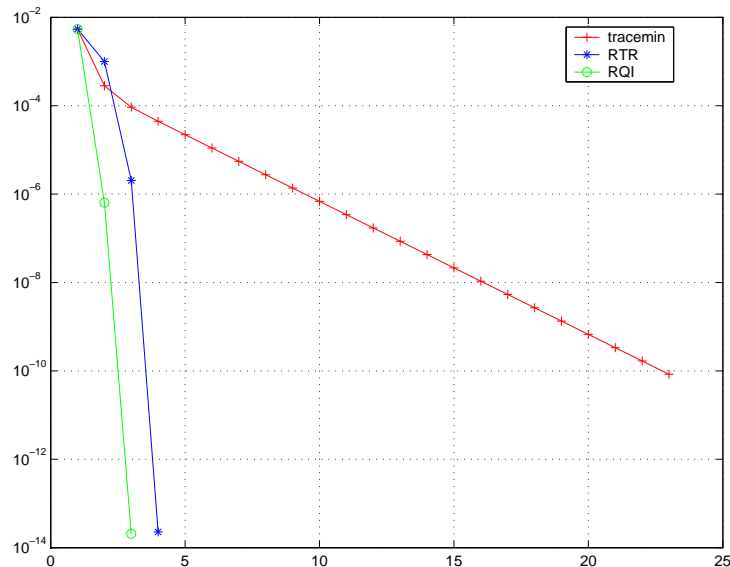


Figure 3: Convergence of three algorithms (Tracemin,RTR,RQI) with respect to the number of *outer* iterations, for an initial guess near the solution.

In the second experiment, the initial iterate for the algorithms was chosen far from the leftmost eigenvector of A . Examining Figure 4, first note that the RQI algorithm does not converge to the leftmost eigenvector. This is the expected result, as the convergence of the RQI algorithm is dependent on the initial iterate. The Tracemin algorithm experiences a linear convergence, as before. Finally, the RTR algorithm experiences a superlinear convergence.

This shows the robustness of the RTR algorithm under a variety of circumstances. The convergence close to the solution is competitive with that of the RQI. At the same time, the RTR experiences a global convergence behaviour like in Tracemin, but lacking in the RQI.

6.2.2 Numerical Efficiency

Whereas the previous set of experiments demonstrated the convergence properties of the RTR algorithm, the next set of experiments seeks to compare the numerical efficiency of the algorithm

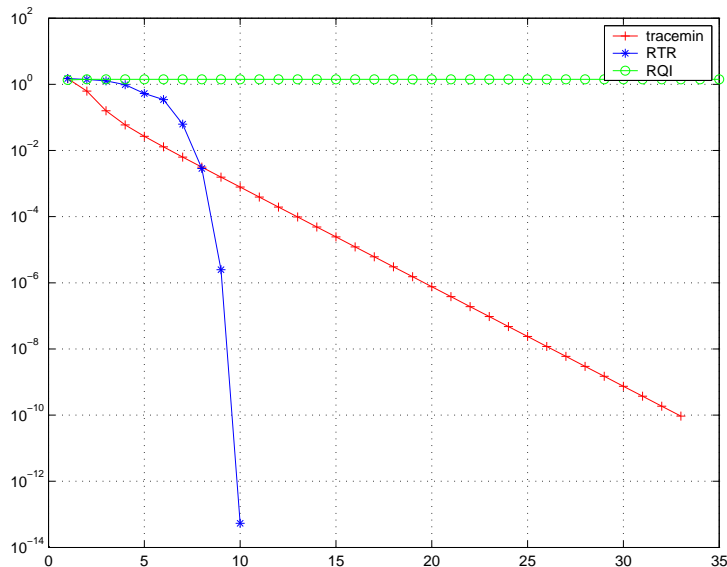


Figure 4: Convergence of three algorithms (Tracemin,RTR,RQI) with respect to the number of *outer* iterations, for an initial guess far from the solution.

against that of current competitors. As each of these methods is inverse-free, the dominant cost in many applications will be the number of multiplications by the matrices A and B .

In both experiments, a symmetric positive-definite matrix A is generated with equally spaced eigenvectors. The θ parameter in (11) is set to $\theta = 2.0$, striving for a quadratic rate of convergence. The dimension of the matrices A and B is $n = 100$.

In the first experiment, the RTR algorithm is compared against the Krylov subspace method described by Golub and Ye [GY02, Alg. 1]. This algorithm, as described by the authors, computes only the leftmost generalized eigenvector of the matrices (A, B) , so that $p = 1$. The symmetric positive definite matrix B is constructed by squaring a randomly generated matrix, as described in the preceding section. The size of the Krylov subspace constructed at each step, controlled by a term m , was set to $m = 5$, allowing the algorithm twice as much memory as used by the RTR algorithm. The Golub-Ye algorithm was implemented without preconditioning.

Figure 5 shows the results. The superlinear convergence of the RTR algorithm was already demonstrated in the previous section. Here we see that in terms of the number of matrix-vector multiplications, RTR outperforms the Krylov method in [GY02], even for mild accuracy requirements. The method of Golub and Ye has been shown to yield faster convergence when a preconditioner is used; future experiments will consider the relative performance of the preconditioned Krylov method against a version of the RTR algorithm with a preconditioned tCG iteration.

The second experiment compares the RTR method against the Block Lanczos method of Golub and Underwood [GU77]. This method computes the leftmost p eigenvectors, but only for the simple symmetric eigenvalue problem ($B = I$). In this experiment, the block size p was set to 5. The variable m controlling the size of the basis for the Krylov subspace was set to $m = 5$, giving the Block Lanczos algorithm the same amount of memory available to the RTR algorithm.

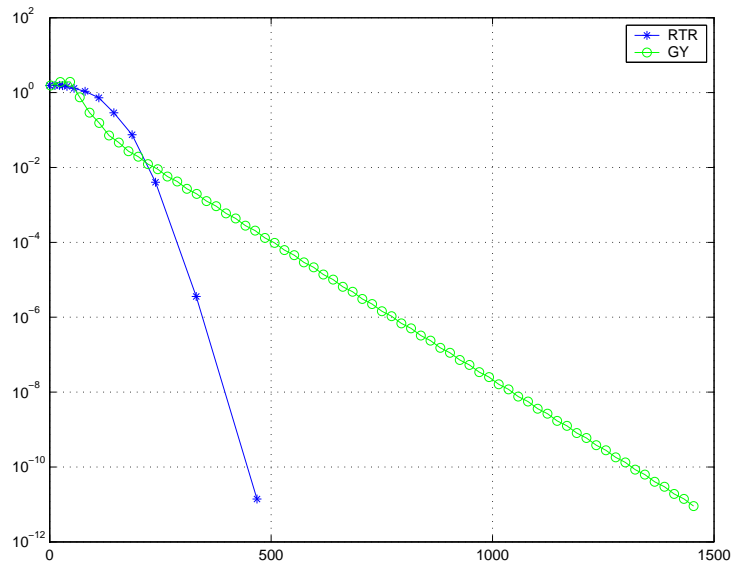


Figure 5: Numerical efficiency of RTR and GY as measured by the number of matrix-vector multiplications by A and B .

Figure 6 shows that the RTR algorithm is competitive with the Block Lanczos algorithm, requiring approximately the same number of matrix-vector multiplications against A to achieve the accuracy.

These experiments show that the Riemannian Trust-Region algorithm, applied to the trace minimization problem, yields a competitive method for computing an extreme eigenspace of a generalized eigenvalue problem.

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 .

An application to the computation of an extreme eigenspace of a symmetric/positive-definite matrix pencil has been presented in detail. It has been shown that for certain problems the algorithm outperforms an inverse-free Krylov subspace method recently proposed in [GY02]. 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 other new computational algorithms and to new convergence results for existing algorithms.

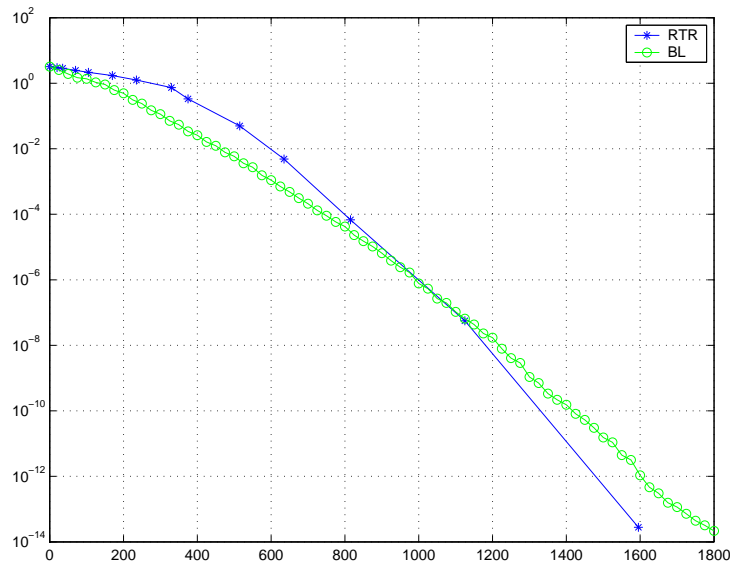


Figure 6: Numerical efficiency of RTR and BL as measured by the number of matrix-vector multiplications by A .

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 with applications in numerical linear algebra*, Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTNS2004), Leuven, Belgium, 5-9 July 2004, 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.

- [AMSV02] P.-A. Absil, R. Mahony, R. Sepulchre, and P. Van Dooren, *A Grassmann-Rayleigh quotient iteration for computing invariant subspaces*, SIAM Review **44** (2002), no. 1, 57–73.
- [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.
- [BSS88] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Math. Programming **40** (1988), no. 3, (Ser. A), 247–263.
- [CD74] J. Cullum and W. E. Donath, *A block generalization of the symmetric s-step Lanczos algorithm*, Tech. Report RC 4845 (21570), IBM Thomas J. Watson Research Center, Yorktown Heights, New York, May 14 1974.
- [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.
- [Cha84] F. Chatelin, *Simultaneous Newton’s iteration for the eigenproblem*, Computing, Suppl. **5** (1984), 67–74.
- [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.
- [Cul78] J. Cullum, *The simultaneous computation of a few of the algebraically largest and smallest eigenvalues of a large, sparse, symmetric matrix*, BIT **18** (1978), no. 3, 265–275.
- [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.
- [Dem87] J. W. Demmel, *Three methods for refining estimates of invariant subspaces*, Computing **38** (1987), 43–57.
- [Dem00] J. Demmel, *Generalized hermitian eigenproblems (Section 2.3)*, 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. 14–18.
- [DM79] J. E. Dennis, Jr. and H. H. W. Mei, *Two new unconstrained optimization algorithms which use function and gradient values*, J. Optim. Theory Appl. **28** (1979), no. 4, 453–482.
- [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.

- [Fat98] J.-L. Fattebert, *A block Rayleigh quotient iteration with local quadratic convergence*, Electron. Trans. Numer. Anal. **7** (1998), 56–74.
- [Gab82] D. Gabay, *Minimizing a differentiable function over a differential manifold*, Journal of Optimization Theory and Applications **37** (1982), no. 2, 177–219.
- [GG03] Gene H. Golub and Chen Greif, *On solving block-structured indefinite linear systems*, SIAM J. Sci. Comput. **24** (2003), no. 6, 2076–2092 (electronic). MR 2004h:65030
- [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 (electronic).
- [GU77] G. H. Golub and R. Underwood, *The block Lanczos method for computing eigenvalues*, Mathematical software, III (Proc. Sympos., Math. Res. Center, Univ. Wisconsin, Madison, Wis., 1977), Academic Press, New York, 1977, pp. 361–377. Publ. Math. Res. Center, No. 39.
- [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.
- [GY02] G. H. Golub and Q. Ye, *An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems*, SIAM J. Sci. Comput. **24** (2002), no. 1, 312–334 (electronic).
- [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 (Section 9.4)*, 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.
- [LE02] E. Lundström and L. Eldén, *Adaptive eigenvalue computations using Newton’s method on the Grassmann manifold*, SIAM J. Matrix Anal. Appl. **23** (2002), no. 3, 819–839.
- [LST98] R. Lösche, H. Schwetlick, and G. Timmerman, *A modified block Newton iteration for approximating an invariant subspace of a symmetric matrix*, Linear Algebra Appl. **275-276** (1998), 381–400.
- [MA03] R. Mahony and P.-A. Absil, *The continuous-time Rayleigh quotient flow on the sphere*, Linear Algebra Appl. **368C** (2003), 343–357.
- [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.
- [MS83] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput. **4** (1983), 553–572.
- [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.
- [MS85] A. Machado and I. Salavessa, *Grassmannian manifolds as subsets of Euclidean spaces*, Res. Notes in Math. **131** (1985), 85–102.
- [MS93] Ronald B. Morgan and David S. Scott, *Preconditioning the Lanczos algorithm for sparse symmetric eigenvalue problems*, SIAM J. Sci. Comput. **14** (1993), no. 3, 585–593. MR 93k:65032
- [MT99] M. Mongeau and M. Torki, *Computing eigenelements of real symmetric matrices via optimization*, Tech. Report MIP 99-54, Université Paul Sabatier, Toulouse, 1999, to appear in Computational Optimization and Applications.
- [NMMA04] Maziar Nikpour, Jonathan H. Manton, Iven M. Y. Mareels, and Vadim Adamyan, *Algorithms for extreme eigenvalue problems*, Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems (MTNS2004), Leuven, Belgium, 5-9 July 2004, 2004.
- [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.
- [Par80] B. N. Parlett, *The symmetric eigenvalue problem*, Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1980, republished by SIAM, Philadelphia, 1998.
- [Pol97] Elijah Polak, *Optimization*, Applied Mathematical Sciences, vol. 124, Springer-Verlag, New York, 1997, Algorithms and consistent approximations. MR 98g:49001
- [Pow70] M. J. D. Powell, *A new algorithm for unconstrained optimization*, Nonlinear Programming (Proc. Sympos., Univ. of Wisconsin, Madison, Wis., 1970), Academic Press, New York, 1970, pp. 31–65.
- [PW79] G. Peters and J. H. Wilkinson, *Inverse iteration, ill-conditioned equations and Newton’s method*, SIAM Review **21** (1979), no. 3, 339–360.
- [Rut70] H. R. Rutishauser, *Simultaneous iteration method for symmetric matrices*, Numerische Mathematik **16** (1970), 205–223.
- [Sco81] David S. Scott, *Solving sparse symmetric generalized eigenvalue problems without factorization*, SIAM J. Numer. Anal. **18** (1981), no. 1, 102–110. MR 82d:65039

- [SE02] V. Simoncini and L. Eldén, *Inexact Rayleigh quotient-type methods for eigenvalue computations*, BIT **42** (2002), no. 1, 159–182.
- [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*, 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.
- [Smi97] P. Smit, *Numerical analysis of eigenvalue algorithms based on subspace iterations*, Ph.D. thesis, CentER, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands, 1997.
- [Sor97] D. C. Sorensen, *Minimization of a large-scale quadratic function subject to a spherical constraint*, SIAM J. Optim. **7** (1997), no. 1, 141–161.
- [SS98] Vivek Sarin and Ahmed Sameh, *An efficient iterative method for the generalized Stokes problem*, SIAM J. Sci. Comput. **19** (1998), no. 1, 206–226 (electronic), Special issue on iterative methods (Copper Mountain, CO, 1996). MR 99b:65132
- [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*, Society for Industrial and Applied Mathematics, Philadelphia, 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*, 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.