SIAM J. NUMER. ANAL.
Vol. 47, No. 2, pp. 997–1018

ACCELERATED LINE-SEARCH AND TRUST-REGION METHODS*

P.-A. ABSIL† AND K. A. GALLIVAN‡

Abstract. In numerical optimization, line-search and trust-region methods are two important classes of descent schemes, with well-understood global convergence properties. We say that these methods are "accelerated" when the conventional iterate is replaced by any point that produces at least as much of a decrease in the cost function as a fixed fraction of the decrease produced by the conventional iterate. A detailed convergence analysis reveals that global convergence properties of line-search and trust-region methods still hold when the methods are accelerated. The analysis is performed in the general context of optimization on manifolds, of which optimization in $\mathbb{R}^n$ is a particular case. This general convergence analysis sheds new light on the behavior of several existing algorithms.

Key words. line search, trust region, subspace acceleration, sequential subspace method, Riemannian manifold, optimization on manifolds, Riemannian optimization, Arnoldi, Jacobi–Davidson, locally optimal block preconditioned conjugate gradient (LOBPCG)

AMS subject classifications. 65B99, 65K05, 65J05, 65F15, 90C30

DOI. 10.1137/08072019X

1. Introduction.

Let f be a real-valued function defined on a domain M, and let $\{x_k\}$ be a sequence of iterates generated as follows: for every k, some $x_{k+1/2} \in \mathcal{M}$ is generated (possibly implicitly) using a descent method that has global convergence to stationary points of f; then x_{k+1} is chosen arbitrarily in the sublevel set $\{x \in M : f(x) \leq f(x_{k+1/2})\}$. We term "acceleration" the fact of choosing x_{k+1} rather than $x_{k+1/2}$ as the new iterate. The question addressed in this paper is whether the inclusion of the acceleration step preserves global convergence, i.e., whether $\{x_k\}$ converges to stationary points. We prove that the answer is positive for a wide class of methods.

The initial motivation for engaging in this general convergence analysis was to obtain a unifying convergence theory for several well-known eigenvalue algorithms. For example, the Jacobi–Davidson approach [38] is a popular technique for computing an eigenpair (eigenvalue and eigenvector) of a matrix A. It is an iterative method where the computation of the next iterate x_{k+1} from the current iterate x_k can be decomposed into two steps. The Jacobi step consists of solving (usually, approximately) a Newton-like equation to obtain an update vector η_k. Whereas in a classical Newton method the new iterate x_{k+1} is defined as $x_k + \eta_k$, the Davidson step uses the update vector η_k to expand a low-dimensional subspace and selects x_{k+1} as the "best" approximation (in some sense) of the sought eigenvector of A within the subspace. A key to the success of this approach is that the problem of computing x_{k+1} within the

*Received by the editors April 3, 2008; accepted for publication (in revised form) September 16, 2008; published electronically February 13, 2009. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors. This work was supported in part by the US National Science Foundation under grant OCI0324944 and by the School of Computational Science of Florida State University.

http://www.siam.org/journals/sinum/47-2/72019.html

†Département d'ingénierie mathématique, Université catholique de Louvain, 1348 Louvain-la-Neuve, Belgium (absil@inma.ucl.ac.be, http://www.inma.ucl.ac.be/~absil).

‡Department of Mathematics, Florida State University, Tallahassee, FL 32306-4510 (kgallivan@fsu.edu, http://www.math.fsu.edu/~gallivan).

subspace can be viewed as a reduced-dimensional eigenvalue problem, which can be solved efficiently when the dimension of the subspace is small. In certain situations, notably when x_{k+1} is chosen as the Ritz vector associated with an extreme Ritz value, the Davidson step can be interpreted as an acceleration step in the sense given above.

The reader primarily interested in eigenvalue algorithms can thus think of the purpose of this paper as formulating and analyzing this Jacobi–Davidson concept in the broad context of smooth optimization, i.e., the minimization of a smooth real-valued cost function over a smooth domain. The "Jacobi" step, instead of being restricted to (inexact) Newton methods, is expanded to cover general line-search and trust-region techniques. The "Davidson" step, or acceleration step, is also made more general: any iterate x_{k+1} is accepted provided that it produces a decrease in the cost function that is at least equal to a prescribed fraction of the decrease produced by the Jacobi update; minimizing the cost function over a subspace that contains the Jacobi update is just one way of achieving this goal.

This new analysis, while requiring only rather straightforward modifications of classical proofs found in the optimization literature, is very general and powerful. In particular, our global convergence analysis yields novel global convergence results for some well-known eigenvalue methods. Moreover, the proof technique is less *ad hoc* than the proofs and derivations usually found in the numerical linear algebra literature, since it simply relies on showing that the methods fit in the broad optimization framework.

What we mean by a smooth domain is a (smooth) manifold. Since the work of Gabay [17], there has been a growing interest for the optimization of smooth cost functions defined on manifolds. Major references include [22, 40, 34, 14, 3]. These differential-geometric techniques have found applications in various areas, such as signal processing, neural networks, computer vision, and econometrics (see, e.g., [6]). The concept of a manifold generalizes the notion of a smooth surface in a Euclidean space. It can thus be thought of as a natural setting for smooth optimization. Roughly speaking, a manifold is a set that is locally smoothly identified with open subsets of $\mathbb{R}^d$, where d is the dimension of the manifold. When the manifold is given to us as a subset of $\mathbb{R}^n$ described by equality constraints, the differential-geometric approach can be viewed as an "informed way" of doing constrained optimization. The resulting algorithms have the property of being feasible (i.e., the iterates satisfy the constraints). In several important cases, however, the manifold is not available as a subset of $\mathbb{R}^n$ but rather as a quotient space. Usually, the fundamental reason why the quotient structure appears is in order to take into account an inherent invariance in the problem. Smooth real-valued functions on quotient manifolds lend themselves as well to differential-geometric optimization techniques. We refer the reader to [6] for a recent overview of this area of research.

The reader solely interested in unconstrained optimization in $\mathbb{R}^n$ should bear in mind that this situation is merely a particular case of the differential-geometric optimization framework considered here. We frequently mention in the text how unconstrained optimization in $\mathbb{R}^n$ is subsumed.

Line-search and trust-region methods are two major techniques for unconstrained optimization in $\mathbb{R}^n$ (see, e.g., [30]). Line-search techniques were proposed and analyzed on manifolds by several authors; see, e.g., [33, 34, 22, 40, 41, 6]. A trust-region framework, based on a systematic use of the concept of retraction, for optimizing functions defined on abstract Riemannian manifolds was proposed more recently [2, 6, 9]. Under reasonable conditions, which hold in particular for smooth cost functions on compact Riemannian manifolds, the trust-region method was shown to converge

to stationary points of the cost function (this is an extension of a well-known result for trust-region methods in $\mathbb{R}^n$). Furthermore, if the trust-region subproblems are (approximately) solved using a truncated conjugate gradient (CG) method with a well-chosen stopping criterion, then the method converges locally superlinearly to the nondegenerate local minima of the cost function. However, these favorable global and local convergence properties do not yield any information on the number of iterates needed, from a given initial point, to reach the local superlinear regime; and, indeed, problems can be crafted where this number of iterates is prohibitively high. The same can be said about the retraction-based line-search approach considered here. Acceleration techniques can be viewed as a way of improving the speed of convergence of those methods.

The acceleration idea is closely related to the subspace expansion concept in Davidson's method for the eigenvalue problem [12] (see also the more recent results in [38, 16, 15]), but the constraints we impose on the acceleration step are weaker than in Davidson-type algorithms. Our approach is also reminiscent of the *sequential subspace method* (SSM) of Hager [20, 25]. Whereas the latter uses subspace acceleration for the purpose of approximately solving trust-region subproblems, we use it as an outermost iteration wrapped around line-search and trust-region methods. The sequential subspace optimization algorithm of Narkiss and Zibulevsky [31] fits in the same framework.

The paper is organized as follows. In section 2, we define the concept of acceleration. The background in optimization on manifolds is recalled in section 3, with a particular emphasis on the case where the manifold is simply $\mathbb{R}^n$. We show global convergence properties for accelerated line-search (section 4) and trust-region (section 5) methods on Riemannian manifolds (of which the classical $\mathbb{R}^n$ is a particular case). Section 6 gives a local convergence result. In section 7, these results are exploited to show global convergence properties of subspace acceleration methods. In particular, a conceptually simple accelerated conjugate gradient method, inspired from the work of Knyazev [26] for the symmetric eigenvalue problem, is proposed, and its global convergence is analyzed. Applications are mentioned in section 8, and conclusions are drawn in section 9.

A preliminary version of this paper appeared in the technical report [4], where the retraction-based line-search scheme and the acceleration concept were introduced.

2. Accelerated optimization methods. In this section, we define the concept of acceleration and briefly discuss acceleration strategies. An important acceleration technique, which consists of minimizing the cost function over an adequately chosen subspace, will be further discussed in section 7.

2.1. Definition. Let f be a cost function defined on an optimization domain M. Given a current iterate $x_k \in M$, line-search and trust-region methods generate a new iterate in M; call it $x_{k+1/2}$. *Accelerating* the method consists of picking a new iterate $x_{k+1} \in M$ that produces at least as much of a decrease in the cost function as a fixed fraction of the decrease produced by $x_{k+1/2}$. In other words, x_{k+1} must satisfy

$$f(x_k) - f(x_{k+1}) \geq c \left(f(x_k) - f(x_{k+1/2})\right) \tag{1}$$

for some constant $c > 0$ independent of k.

2.2. Acceleration strategies. This relaxation on the choice of the new iterate introduces leeway for exploiting information that may improve the behavior of the method. For example, x_{k+1} can be determined by minimizing f over some well-

chosen subset of the domain M, built using information gained over the iterations. This idea is developed in section 7.

Moreover, a wide variety of "hybrid" optimization methods fit in the framework of (1). For example, let $\mathcal{A}$ be a line-search or trust-region algorithm, and let $\mathcal{B}$ be any descent method. If, for all k, $x_{k+1/2}$ is obtained from x_k by $\mathcal{A}$ and x_{k+1} is obtained from $x_{k+1/2}$ by $\mathcal{B}$, then the sequence $\{x_k\}$ is generated by an accelerated line-search or trust-region algorithm. Likewise, for all k, let $x_{k+1/2}$ be obtained from x_k by $\mathcal{A}$, let $\tilde{x}_{k+1/2}$ be obtained from x_k by $\mathcal{B}$, and let $x_{k+1} = x_{k+1/2}$ if $f(x_{k+1/2}) \leq f(\tilde{x}_{k+1/2})$ and $x_{k+1} = \tilde{x}_{k+1/2}$ otherwise; then the sequence $\{x_k\}$ is again generated by an accelerated line-search or trust-region method.

Note that, until we reach section 7 on subspace acceleration, we make no assumption other than (1) on how x_{k+1} is chosen from $x_{k+1/2}$. We also point out that values of c in the open interval $(0, 1)$ do not correspond to acceleration in the intuitive sense of the term since $f(x_{k+1})$ is possibly greater than $f(x_{k+1/2})$. Actually, all practical accelerated methods considered in section 8 satisfy (1) with $c = 1$. However, we consider the general case $c > 0$ because it may be useful in some situations and the global convergence analysis for $c > 0$ is not significantly more complicated than for $c = 1$.

3. Preliminaries on Euclidean and Riemannian optimization. In this paper, we assume that the optimization domain M is a (finite-dimensional) Riemannian manifold. The particularization to unconstrained optimization in $\mathbb{R}^n$ is made explicit whenever we feel that it improves readability.

Loosely speaking, a manifold is a topological set covered by mutually compatible local parameterizations. We refer, e.g., to [13, 6] for details. An important type of manifolds are those subsets of $\mathbb{R}^n$ with a tangent space of constant dimension defined at each point (simple examples are spheres and $\mathbb{R}^n$ itself). If the tangent spaces T_xM are equipped with an inner product $\langle \cdot, \cdot \rangle_x$ that varies smoothly with x, then the manifold is called *Riemannian*. In this paper, we consider the problem of minimizing a real function f (the *cost function*) defined on a Riemannian manifold M.

Classical unconstrained optimization in $\mathbb{R}^n$ corresponds to the case $M = \mathbb{R}^n$. The tangent space to $\mathbb{R}^n$ at any point $x \in \mathbb{R}^n$ is canonically identified with $\mathbb{R}^n$ itself: $T_x\mathbb{R}^n \simeq \mathbb{R}^n$. The canonical Riemannian structure on $\mathbb{R}^n$ is its usual Euclidean vector space structure, where the inner product at $x \in \mathbb{R}^n$ defined by $\langle \xi, \zeta \rangle := \xi^T \zeta$ for all $\xi, \zeta \in T_x\mathbb{R}^n \simeq \mathbb{R}^n$.

The major problem to overcome is that manifolds are in general not flat so that the sum of two elements of M or their multiplication by scalars is not defined. A remedy advocated in [2] is to locally "flatten" the manifold onto the tangent space $T_{x_k}M$ at the current iterate x_k. This is done by means of a *retraction*, a concept proposed by Shub [32, 3].

DEFINITION 3.1 (retraction). *A* retraction *on a manifold M is a mapping R from the tangent bundle TM into M with the following properties (let R_x denote the restriction of R to T_xM):*

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

Instead of the third condition, it is equivalent to require that $\frac{\mathrm{d}}{\mathrm{d}t} R_x(t\xi_x)\big|_{t=0} = \xi_x$ for all $\xi_x \in T_xM$.

We do not necessarily assume that R is defined on the whole tangent bundle TM, but we make the blanket assumption that its evaluation never fails in the algorithms. Note that the third condition implies that R_x is defined on a neighborhood of the origin of T_xM for all $x \in M$; this guarantees that, given $\eta_x \in T_xM$, $R_x(t\eta_x)$ is well-defined at least on some nonempty interval $-\epsilon < t < \epsilon$.

On a Riemannian manifold, it is always possible to choose the retraction R as the exponential mapping (which is defined everywhere when the manifold is *complete*). Using the exponential, however, may not be computationally sensible. The concept of retraction gives the possibility of choosing more efficient substitutes (see [3, 6]). Given a cost function f on a manifold M equipped with a retraction R, we define the lifted cost function at $x \in M$ as

$$\hat{f}_x : T_xM \to \mathbb{R} : \xi \mapsto f(R_x(\xi)). \tag{2}$$

When $M = \mathbb{R}^n$, the natural retraction is given by

$$R_x(\xi) := x + \xi, \tag{3}$$

and $\hat{f}_x$ satisfies $\hat{f}_x(\xi) = f(x + \xi)$ for all $x \in \mathbb{R}^n$ and all $\xi \in T_x\mathbb{R}^n \simeq \mathbb{R}^n$.

Given a current iterate x_k on M, any line-search or trust-region method applied to $\hat{f}_{x_k}$ produces a vector η_k in $T_{x_k}M$. In a line-search method, η_k is used as a search direction: a point is sought on the curve $t \mapsto R_{x_k}(t\eta_k)$ that satisfies some conditions on the cost function (e.g., a line minimizer or the Armijo condition). In a trust-region method [2], η_k defines a proposed new iterate $R_{x_k}(\eta_k)$. In both cases, the optimization method yields a proposed new iterate $x_{k+1/2}$ in M. Below we study the convergence properties of such schemes when they are accelerated in the sense of (1).

4. Accelerated line-search methods. Line-search methods (without acceleration) on a manifold M endowed with a retraction R are based on the update formula

$$x_{k+1} = R_{x_k}(t_k\eta_k),$$

where η_k is in $T_{x_k}M$ and t_k is a scalar. The two issues are to select the search direction η_k and then the step length t_k. To obtain global convergence results, some restrictions have to be imposed on η_k and t_k. The following definition concerning η_k is adapted from [10].

DEFINITION 4.1 (gradient-related). *A sequence* $\{\eta_k\}$, $\eta_k \in T_{x_k}M$, *is* gradient-related *if, for any subsequence* $\{x_k\}_{k\in\mathcal{K}}$ *in* M *that converges to a nonstationary point, the corresponding subsequence* $\{\eta_k\}_{k\in\mathcal{K}}$ *is bounded and satisfies*

$$\limsup_{k\to\infty,\ k\in\mathcal{K}} \langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k} < 0.$$

When $M = \mathbb{R}^n$ with its canonical Euclidean structure, we have $\operatorname{grad} f(x) = \begin{bmatrix}\partial_1 f(x) & \cdots & \partial_n f(x)\end{bmatrix}^T$ and $\langle \operatorname{grad} f(x), \eta\rangle = \eta^T \operatorname{grad} f(x)$, where we used the canonical identification $T_x\mathbb{R}^n \simeq \mathbb{R}^n$. (One must bear in mind that when we use the identification $T_x\mathbb{R}^n \simeq \mathbb{R}^n$, we lose the information on the foot x of the tangent vector. In order to specify the foot, we say that $\{\eta_k\} \subseteq \mathbb{R}^n$ is gradient-related *to* $\{x_k\}$.)

There is a relation between the gradient relatedness of $\{\eta_k\}$ and the angle between η_k and the steepest-descent direction. Let $\angle(-\operatorname{grad} f(x_k), \eta_k) = \arccos \frac{\langle -\operatorname{grad} f(x_k), \eta_k\rangle_{x_k}}{\|\operatorname{grad} f(x_k)\|_{x_k}\|\eta_k\|_{x_k}}$ denote the angle between η_k and the steepest-descent direction $-\operatorname{grad} f(x_k)$. Let $\{\eta_k\}$ be such that $c_1 \le \|\eta_k\|_{x_k} \le c_2$ for some $0 < c_1 < c_2 < \infty$ and all k. Then the condition $\angle(-\operatorname{grad} f(x_k), \eta_k) \ge \theta$ for some fixed $\theta > \frac{\pi}{2}$ and all k is sufficient for the

sequence $\{\eta_k\}$ to be gradient-related to $\{x_k\}$. In particular, assume that η_k is obtained by solving a linear system $\mathcal{A}_k\eta_k = -\operatorname{grad} f(x_k)$, where $\mathcal{A}_k$ is a linear symmetric positive-definite transformation of $T_{x_k}M$. Then $\cos\angle(-\operatorname{grad} f(x_k), \eta_k) \geq \kappa^{-1}(\mathcal{A}_k)$, where $\kappa(\mathcal{A}_k)$ denotes the condition number of $\mathcal{A}_k$. Hence if the smallest eigenvalue of $\mathcal{A}_k$ is bounded away from zero and the largest eigenvalue of $\mathcal{A}_k$ is bounded, then $\{\eta_k\}$ is bounded away from zero and infinity and the condition number of $\mathcal{A}_k$ is bounded, and thus $\{\eta_k\}$ is gradient-related. (Note that the condition that the linear operator $\mathcal{A} : T_xM \to T_xM$ is symmetric positive-definite means that $\langle u, \mathcal{A}v\rangle_x = \langle \mathcal{A}u, v\rangle_x$ for all $u, v \in T_xM$, and $\langle u, \mathcal{A}u\rangle_x > 0$ for all nonzero $u \in T_xM$. In the case of $\mathbb{R}^n$ endowed with its canonical inner product, this corresponds to the classical definitions of symmetry and positive definiteness for the matrix representing the operator $\mathcal{A}$.)

The next definition, related to the choice of the step length t_k, relies on Armijo's backtracking procedure [7] (or see [10]) to find a point at which there is sufficient decrease of the cost function.

Definition 4.2 (Armijo point). *Given a differentiable cost function f on a Riemannian manifold M with retraction R, a point $x \in M$, a nonzero descent vector $\eta \in T_xM$ (i.e., $\langle \operatorname{grad} f(x), \eta\rangle_x < 0$), a scalar $\overline{\alpha} > 0$ such that the segment $[0, \overline{\alpha}]\eta \subseteq T_xM$ is included in the domain of R, and scalars $\beta \in (0,1)$ and $\sigma \in (0,1)$, the* Armijo vector *is defined as $\eta^A = \beta^m\overline{\alpha}\eta$, where m is the first nonnegative integer such that*

$$f(x) - f(R_x(\beta^m\overline{\alpha}\eta)) \geq -\sigma \langle \operatorname{grad} f(x), \beta^m\overline{\alpha}\eta\rangle_x. \tag{4}$$

The Armijo point *is $R_x(\beta^m\overline{\alpha}\eta) \in M$.*

It can be shown, using the classical Armijo theory for the lifted cost function $\hat{f}_x$, that there is always an m such that (4) holds, and hence the definition is legitimate. A similar definition was proposed in [41] for the particular case where the retraction is the exponential mapping. When $M = \mathbb{R}^n$ with its canonical Euclidean structure, the definition reduces to the classical situation described, e.g., in [10].

We propose the following accelerated Riemannian line-search algorithm.

Algorithm 1. Accelerated Line Search (ALS)

Require: Riemannian manifold M; continuously differentiable scalar field f on M; retraction R from TM to M as in Definition 3.1; scalars $\overline{\alpha} > 0$, $c, \beta, \sigma \in (0,1)$.

Input: Initial iterate $x_0 \in M$.

Output: Sequence of iterates $\{x_k\} \subseteq M$ and search directions $\{\eta_k\} \subseteq TM$.

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: Pick a descent vector η_k in $T_{x_k}M$ such that $t\eta_k$ is in the domain of R for all $t \in [0, \overline{\alpha}]$.

3: Select $x_{k+1} \in M$ such that

$$f(x_k) - f(x_{k+1}) \geq c\left(f(x_k) - f\left(R_{x_k}\left(\eta^A\right)\right)\right), \tag{5}$$

where η^A is the Armijo vector (Definition 4.2 with $x := x_k$ and $\eta := \eta_k$).

4: **end for**

Observe that Algorithm 1, as well as most other algorithms in this paper, describes a *class* of numerical algorithms; one could call it an *algorithm model*. The purpose of this analysis paper is to give (strong) convergence results for (broad) classes of algorithms. For Algorithm 1, we have the following convergence result, whose proof closely follows [10, Proposition 1.2.1]. The result is, however, more general in three

aspects. (1) Even when the optimization domain is $\mathbb{R}^n$, the line search is not necessarily done on a straight line, because the choice of the retraction is not restricted to the natural retraction (3) in $\mathbb{R}^n$. (2) Even in the case of $\mathbb{R}^n$, points other than the Armijo point can be selected, as long as they satisfy the acceleration condition (5). (3) Finally, the optimization domain can be any Riemannian manifold.

THEOREM 4.3. *Let $\{x_k\}$ be an infinite sequence of iterates generated by Algorithm* 1 *(ALS), and assume that the generated sequence $\{\eta_k\}$ of search directions is gradient-related (Definition* 4.1*). Then every limit point of $\{x_k\}$ is a stationary point of f.*

Proof. The proof is by contradiction. Suppose that there is a subsequence $\{x_k\}_{k\in\mathcal{K}}$ converging to some x^* with $\operatorname{grad} f(x^*) \neq 0$. Since $\{f(x_k)\}$ is nonincreasing, it follows that $\{f(x_k)\}$ converges to $f(x^*)$. Hence $f(x_k) - f(x_{k+1})$ goes to zero. By the construction of the algorithm,

$$f(x_k) - f(x_{k+1}) \geq -c\sigma\alpha_k \langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k},$$

where $\alpha_k\eta_k$ is the Armijo vector. Since $\{\eta_k\}$ is gradient-related, it follows that $\{\alpha_k\}_{k\in\mathcal{K}} \to 0$. It follows that for all k greater than some $\overline{k}$, $\alpha_k < \overline{\alpha}$, which means that $\alpha_k = \beta^m\overline{\alpha}$ for some $m \geq 1$, which implies that the previously tried step size $\beta^{m-1}\overline{\alpha} = \alpha_k/\beta$ did not satisfy the Armijo condition. In other words,

$$f(x_k) - f\left(R_{x_k}(\alpha_k/\beta)\eta_k\right) < -\sigma(\alpha_k/\beta)\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k} \quad \forall k \in \mathcal{K},\ k \geq \overline{k}.$$

Denoting

$$\tilde{\eta}_k = \frac{\eta_k}{\|\eta_k\|} \quad \text{and} \quad \tilde{\alpha}_k = \frac{\alpha_k\|\eta_k\|}{\beta}, \tag{6}$$

the inequality above reads

$$\frac{\hat{f}_{x_k}(0) - \hat{f}_{x_k}(\tilde{\alpha}_k\tilde{\eta}_k)}{\tilde{\alpha}_k} < -\sigma\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k} \quad \forall k \in \mathcal{K},\ k \geq \overline{k},$$

where $\hat{f}$ is defined as in (2). The mean value theorem yields

$$-\langle \operatorname{grad} \hat{f}_{x_k}(t\tilde{\eta}_k), \tilde{\eta}_k\rangle_{x_k} < -\sigma\langle \operatorname{grad} f(x_k), \eta_k\rangle_{x_k} \quad \forall k \in \mathcal{K},\ k \geq \overline{k}, \tag{7}$$

where t is in the interval $[0, \tilde{\alpha}_k]$. Since $\{\alpha_k\}_{k\in\mathcal{K}} \to 0$ and since η_k is gradient-related, hence bounded, it follows that $\{\tilde{\alpha}_k\}_{k\in\mathcal{K}} \to 0$. Moreover, since $\tilde{\eta}_k$ has unit norm and its foot x_k converges on the index set $\mathcal{K}$, it follows that $\{\eta_k\}_{k\in\mathcal{K}}$ is included in some compact subset of the tangent bundle TM, and therefore there exists an index set $\tilde{\mathcal{K}} \subseteq \mathcal{K}$ such that $\{\tilde{\eta}_k\}_{k\in\tilde{\mathcal{K}}} \to \tilde{\eta}^*$ for some $\tilde{\eta}^* \in T_{x^*}M$ with $\|\tilde{\eta}^*\| = 1$. We now take the limit in (7) over $\tilde{\mathcal{K}}$. Since the Riemannian metric is continuous (by definition), $f \in C^1$, and $\operatorname{grad} \hat{f}_{x_k}(0) = \operatorname{grad} f(x_k)$ (because of point 3 in Definition 3.1, see [6, equation (4.4)]), we obtain

$$-\langle \operatorname{grad} f(x^*), \tilde{\eta}^*\rangle_{x^*} \leq -\sigma\langle \operatorname{grad} f(x^*), \tilde{\eta}^*\rangle_{x^*}.$$

Since $0 < \sigma < 1$, it follows that $\langle \operatorname{grad} f(x^*), \tilde{\eta}^*\rangle_{x^*} \geq 0$. On the other hand, from the fact that $\{\eta_k\}$ is gradient-related, one obtains that $\langle \operatorname{grad} f(x^*), \tilde{\eta}^*\rangle_{x^*} < 0$, a contradiction. ∎

More can be said under compactness assumptions, using a standard topological argument. (The purpose of the compactness assumption is to ensure that every subsequence of $\{x_k\}$ has at least one limit point.)

COROLLARY 4.4. *Let $\{x_k\}$ be an infinite sequence of iterates generated by Algorithm 1 (ALS), and assume that the generated sequence $\{\eta_k\}$ of search directions is gradient-related (Definition 4.1). Assume that there is a compact set $\mathcal{C}$ such that $\{x_k\} \subseteq \mathcal{C}$. (This assumption holds in particular when the sublevel set $\mathcal{L} = \{x \in M : f(x) \leq f(x_0)\}$ is compact: the iterates all belong to the sublevel set since f is nonincreasing. It also holds when M itself is compact.) Then* $\lim_{k\to\infty} \|\operatorname{grad} f(x_k)\| = 0$.

Proof. The proof is by contradiction. Assume the contrary; i.e., there is a subsequence $\{x_k\}_{k\in\mathcal{K}}$ and $\epsilon > 0$ such that $\|\operatorname{grad} f(x_k)\| > \epsilon$ for all $k \in \mathcal{K}$. Since $\{x_k\} \subseteq \mathcal{C}$, with $\mathcal{C}$ compact, it follows that $\{x_k\}_{k\in\mathcal{K}}$ has a limit point x^* in $\mathcal{C}$ (Bolzano–Weierstrass theorem). By continuity of grad f, one has $\|\operatorname{grad} f(x^*)\| \geq \epsilon$, i.e., x^* is not stationary, a contradiction with Theorem 4.3. ∎

5. Accelerated trust-region algorithm. We first briefly recall the basics of the Riemannian trust-region scheme (RTR) proposed in [2]. Let M be a Riemannian manifold with retraction R. Given a cost function $f : M \to \mathbb{R}$ and a current iterate $x_k \in M$, we use R_{x_k} to locally map the minimization problem for f on M into a minimization problem for the cost function $\hat{f}_{x_k}$ defined as in (2). The Riemannian metric g turns $T_{x_k}M$ into a Euclidean space endowed with the inner product $g_{x_k}(\cdot,\cdot)$, which makes it possible to consider the following *trust-region subproblem* in the Euclidean space $T_{x_k}M$:

$$\min_{\eta\in T_{x_k}M} m_{x_k}(\eta) \quad \text{subject to } \langle \eta, \eta\rangle_{x_k} \leq \Delta_k^2, \tag{8a}$$

where

$$m_{x_k}(\eta) \equiv f(x_k) + \langle \operatorname{grad} f(x_k), \eta\rangle_{x_k} + \frac{1}{2}\langle \mathcal{H}_{x_k}\eta, \eta\rangle_{x_k}, \tag{8b}$$

Δ_k is the *trust-region radius*, and $\mathcal{H}_{x_k} : T_{x_k}M \to T_{x_k}M$ is some symmetric linear operator, i.e., $\langle \mathcal{H}_{x_k}\xi, \chi\rangle_{x_k} = \langle \xi, \mathcal{H}_{x_k}\chi\rangle_{x_k}$, $\xi, \chi \in T_xM$. Note that m_{x_k} need not be the exact quadratic Taylor expansion of $\hat{f}_{x_k}$ about zero, since $\mathcal{H}_k$ is freely chosen.

Next, an approximate solution η_k to the trust-region subproblem (8) is produced. For the purpose of obtaining global convergence results, the η_k need not be the exact solution provided it produces a sufficient decrease of the model, as specified later. The decision to accept or not the candidate $R_{x_k}(\eta_k)$ and to update the trust-region radius is based on the quotient

$$\rho_k = \frac{f(x_k) - f(R_{x_k}(\eta_k))}{m_{x_k}(0_{x_k}) - m_{x_k}(\eta_k)} = \frac{\hat{f}_{x_k}(0_{x_k}) - \hat{f}_{x_k}(\eta_k)}{m_{x_k}(0_{x_k}) - m_{x_k}(\eta_k)} \tag{9}$$

measuring the agreement between the model decrease and the function decrease at the proposed iterate.

The following algorithm differs from the RTR algorithm of [2] only below the line "**if** $\rho_k > \rho'$." (The specific rules for accepting the proposed new iterate and updating the trust-region radius come from [30]; they form a particular instance of the rules given in [11].)

Next, we study the global convergence of Algorithm 2. We show that, under some assumptions on the cost function, the model and the quality of η_k, it holds

ALGORITHM 2. ACCELERATED TRUST REGION (ATR)

Require: Riemannian manifold M; scalar field f on M; retraction R from TM to M as in Definition 3.1. Parameters: $\bar{\Delta} > 0$, $\Delta_0 \in (0, \bar{\Delta})$, and $\rho' \in [0, \frac{1}{4})$, $c \in (0, 1)$, $c_1 > 0$.

Input: Initial iterate $x_0 \in M$.

Output: Sequence of iterates $\{x_k\}$.

for $k = 0, 1, 2, \ldots$ **do**
  Obtain η_k by (approximately) solving (8).
  Evaluate ρ_k from (9);
  if $\rho_k < \frac{1}{4}$ **then**
    $\Delta_{k+1} = \frac{1}{4}\Delta_k$
  else if $\rho_k > \frac{3}{4}$ and $\|\eta_k\| = \Delta_k$ **then**
    $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$
  else
    $\Delta_{k+1} = \Delta_k$;
  end if
  if $\rho_k > \rho'$ **then**
    Select $x_{k+1} \in M$ such that

$$f(x_k) - f(x_{k+1}) \geq c\,(f(x_k) - f(R_{x_k}(\eta_k)))\,; \tag{10}$$

  else
    Select $x_{k+1} \in M$ such that

$$f(x_k) - f(x_{k+1}) \geq 0; \tag{11}$$

  end if
end for

that the gradient of the cost function goes to zero at least on a subsequence of $\{x_k\}$. This is done by slightly modifying the corresponding development given in [2] to take acceleration into account.

We need the following definition.

DEFINITION 5.1 (radially L-C^1 function). *Let $\hat{f} : TM \to \mathbb{R}$ be as in (2). We say that $\hat{f}$ is* radially Lipschitz continuously differentiable *if there exist reals $\beta_{RL} > 0$ and $\delta_{RL} > 0$ such that, for all $x \in M$, for all $\xi \in TM$ with $\|\xi\| = 1$, and for all $t < \delta_{RL}$, it holds that*

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

For the purpose of Algorithm 2, which is a descent algorithm, this condition needs only to be imposed in the level set

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

We also require the approximate solution η_k of the trust-region subproblem (8) to produce a sufficient decrease in the model. More precisely, η_k must produce at least as much of a decrease in the model function as a fixed fraction of the so-called Cauchy decrease; see [30, 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

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

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

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

In particular, the Steihaug–Toint truncated CG method (see, e.g., [37, 30, 11]) satisfies this bound (with $c_1 = \frac{1}{2}$, see [30, Lemma 4.5]) since it first computes the Cauchy point and then attempts to improve the model decrease.

With these things in place, we can state and prove the following global convergence result.

THEOREM 5.2. *Let $\{x_k\}$ be a sequence of iterates generated by Algorithm 2 (ATR) with $\rho' \in [0, \frac{1}{4})$. Suppose that f is C^1 and bounded below on the level set (13), that $\hat{f}$ is radially L-C^1 (Definition 5.1), and that $\|\mathcal{H}_k\| \leq \beta$ for some constant β. Further suppose that all approximate solutions η_k of (8) satisfy the Cauchy decrease inequality (14) for some positive constant c_1. We then have*

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

Proof. Here is a brief outline of the proof for the reader's convenience. We will assume for contradiction that the norm of the gradient is bounded away from zero. Then a key to reaching a contradiction is that the trust-region does not shrink to zero (21). This is ensured by showing that ρ_k is greater than $\frac{1}{2}$ whenever Δ_k is smaller than a global value (20). This result itself is obtained by imposing that the discrepancy between the model and the cost function is uniformly quadratic (17) and that the denominator of ρ_k is bounded below by a ramp function of Δ_k (14).

We now turn to the detailed proof. First, we perform some manipulation of ρ_k from (9):

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{(f(x_k) - \hat{f}_{x_k}(\eta_k)) - (m_k(0) - m_k(\eta_k))}{m_k(0) - m_k(\eta_k)} \right| \\ &= \left| \frac{m_k(\eta_k) - \hat{f}_{x_k}(\eta_k)}{m_k(0) - m_k(\eta_k)} \right|. \end{aligned} \tag{16}$$

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

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

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 (12). Therefore, it follows from the

definition (8b) of m_k that

$$\begin{aligned}\left|m_k(\eta_k) - \hat{f}_{x_k}(\eta_k)\right| &= \left|\frac{1}{2}\langle \mathcal{H}_{x_k}\eta_k, \eta_k\rangle_{x_k} - \epsilon'\right| \\ &\leq \frac{1}{2}\beta\|\eta_k\|^2 + \frac{1}{2}\beta_{RL}\|\eta_k\|^2 \leq \beta'\|\eta_k\|^2 \end{aligned} \tag{17}$$

whenever $\|\eta_k\| < \delta_{RL}$, where $\beta' = \max(\beta, \beta_{RL})$. Assume for the purpose of contradiction that $\liminf_{k\to\infty} \|\operatorname{grad} f(x_k)\| \neq 0$; that is, assume that there exist $\epsilon > 0$ and a positive index K such that

$$\|\operatorname{grad} f(x_k)\| \geq \epsilon \quad \forall k \geq K. \tag{18}$$

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

$$m_k(0) - m_k(\eta_k) \geq c_1\|\operatorname{grad} f(x_k)\| \min\left(\Delta_k, \frac{\|\operatorname{grad} f(x_k)\|}{\|\mathcal{H}_k\|}\right) \geq c_1\epsilon \min\left(\Delta_k, \frac{\epsilon}{\beta'}\right). \tag{19}$$

Substituting (17) and (19) into (16), we have that

$$|\rho_k - 1| \leq \frac{\beta'\|\eta_k\|^2}{c_1\epsilon \min\left(\Delta_k, \frac{\epsilon}{\beta'}\right)} \leq \frac{\beta'\Delta_k^2}{c_1\epsilon \min\left(\Delta_k, \frac{\epsilon}{\beta'}\right)} \tag{20}$$

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 (20) when $\Delta_k \leq \hat{\Delta}$. Choose $\hat{\Delta}$ as follows:

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

This gives us $\min(\Delta_k, \frac{\epsilon}{\beta'}) = \Delta_k$. We can now write (20) as follows:

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

Therefore, $\rho_k \geq \frac{1}{2} > \frac{1}{4}$ whenever $\Delta_k \leq \hat{\Delta}$ so that, by the workings of Algorithm 2, it follows 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 2 only when $\Delta_k > \hat{\Delta}$. Therefore, we conclude that

$$\Delta_k \geq \min\left(\Delta_K, \hat{\Delta}/4\right) \quad \forall k \geq K. \tag{21}$$

Consequently, $\rho_k \geq \frac{1}{4}$ must hold infinitely many times (otherwise $\{\Delta_k\}$ would go to zero by the workings of the algorithm). So there exists 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 $k \geq K$, it follows from (19) and (10) that

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

Since, moreover, $f(x_k) - f(x_{k+1}) \geq 0$ for all $k \notin \mathcal{K}$, it follows that $f(x_k) \to -\infty$, a contradiction since f is bounded below on the level set containing $\{x_k\}$. ∎

The convergence result of Theorem 5.2 is essentially identical to the corresponding result for the non-accelerated Riemannian trust-region method (see [2] or [6]), which itself is a natural generalization of a convergence result of the classical (non-accelerated) trust-region method in $\mathbb{R}^n$. In the classical convergence theory of trust-region methods in $\mathbb{R}^n$ (see, e.g., [30, 11]), this result is followed by another theorem stating that, under further assumptions, $\lim_{k\to\infty} \|\operatorname{grad} f(x_k)\| = 0$; i.e., the gradient of the cost function goes to zero on the *whole* sequence of iterates. This result also has a natural generalization for the non-accelerated Riemannian trust-region method (see [2, Theorem 4.4] or [6, Theorem 7.4.4]). It is an open question whether this result extends verbatim to the accelerated case. At least we can say that the proof cannot be adapted in a simple way: the condition that there exist $\mu > 0$ and $\delta_\mu > 0$ such that

$$\|\xi\| \geq \mu \operatorname{dist}(x, R_x(\xi)) \quad \text{for all } x \in \mathcal{M}, \text{for all } \xi \in T_x\mathcal{M},\ \|\xi\| \leq \delta_\mu, \tag{22}$$

no longer implies that $\|\eta_k\| \geq \mu \operatorname{dist}(x_k, x_{k+1})$ when acceleration comes into play. A simple fix is to require that there exists $\mu > 0$ such that the iterates satisfy

$$\|\eta_k\| \geq \mu \operatorname{dist}(x_k, x_{k+1}) \quad \text{for all } k. \tag{23}$$

We then obtain the following result. (We refer to [2, 6] for the concept of Lipschitz continuous differentiability of f on the Riemannian manifold M; the definition reduces to the classical one when the manifold is $\mathbb{R}^n$. The extension of the proof of [6, Theorem 7.4.4] to a proof of Theorem 5.3 is left to the reader.)

THEOREM 5.3. *Let $\{x_k\}$ be a sequence of iterates generated by Algorithm 2 (ATR). Suppose that all of the assumptions of Theorem 5.2 are satisfied. Further suppose that $\rho' \in (0, \frac{1}{4})$, that f is Lipschitz continuously differentiable, and that (23) is satisfied for some $\mu > 0$. It then follows that*

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

6. Local convergence. We now briefly comment on how accelerating an optimization method may affect its order of convergence. Consider an algorithm that converges locally with order q to a local minimum v of the cost function f; that is,

$$\operatorname{dist}(x_+, v) \leq c_0\ (\operatorname{dist}(x, v))^q$$

for some $c_0 > 0$ and all x in some neighborhood of v, where x_+ stands for the next iterate computed from the current iterate x. If the algorithm is accelerated in the sense of (1), then local convergence to v is no longer guaranteed without further hypotheses; i.e., the algorithm may converge to stationary points other than v. However, for sequences of iterates of the accelerated algorithm that converge to v, we have the following result.

PROPOSITION 6.1. *Let v be a nondegenerate minimizer of $f \in C^3(M)$, where M is a Riemannian manifold. Consider a descent algorithm that converges locally with order $q > 1$ to v. If $\{x_k\}$ is a sequence of iterates of an accelerated version of the descent algorithm, in the sense of (1) with $c = 1$, and $\{x_k\}$ converges to v, then it does so with order q.*

Proof. We work in a coordinate system around v. Abusing notation, we use the same symbols for points of M and their coordinate representations. There is a neighborhood $\mathcal{U}$ of v such that, for all $x \in \mathcal{U}$, we have

$$\frac{1}{2}\lambda_m \|x - v\|^2 \leq f(x) - f(v) \leq 2\lambda_M \|x - v\|^2,$$

where $\lambda_M \geq \lambda_m > 0$ denote the largest and smallest eigenvalues, respectively, of the Hessian of f at v (they are positive since v is a nondegenerate minimizer). Since $c = 1$, it follows from (1) that $f(x_{k+1}) \leq f(x_{k+1/2})$. Moreover, by the equivalence of norms, there is a neighborhood $\mathcal{U}_1$ of v and constants c_1 and c_2 such that, for all $x \in \mathcal{U}_1$, $\frac{1}{c_1}\mathrm{dist}(x, v) \leq \|x - v\| \leq c_2\mathrm{dist}(x, v)$. Since the original descent algorithm converges locally with order q to v, there exists a nonempty open ball $B_\epsilon(v)$ such that, whenever $x_k \in B_\epsilon(v)$, it holds that $x_{k+1/2} \in B_\epsilon(v)$ with $\mathrm{dist}(x_{k+1/2}, v) \leq c_0\ (\mathrm{dist}(x_k, v))^q$. Moreover, ϵ can be chosen such that $B_\epsilon(v) \subseteq \mathcal{U} \cap \mathcal{U}_1$. Since $\{x_k\}$ converges to v, there is K such that, for all $k > K$, x_k belongs to $B_\epsilon(v)$. We have, for all $k > K$,

$$\begin{aligned}(\mathrm{dist}(x_{k+1}, v))^2 &\leq c_1^2\|x_{k+1} - v\|^2 \\ &\leq c_1^2\frac{2}{\lambda_m}(f(x_{k+1}) - f(v)) \leq c_1^2\frac{2}{\lambda_m}(f(x_{k+1/2}) - f(v)) \\ &\leq c_1^2\frac{4}{\lambda_m}\lambda_M\|x_{k+1/2} - v\|^2 \leq c_1^2\frac{4}{\lambda_m}\lambda_M c_0 c_2^2\left(\mathrm{dist}(x_{k+1/2}, v)\right)^2 \\ &\leq c_1^2\frac{4}{\lambda_m}\lambda_M c_0^2 c_2^2\left(\mathrm{dist}(x_k, v)\right)^{2q}. \qquad \square\end{aligned}$$

7. Sequential subspace optimization methods. We consider sequential subspace optimization methods in the form given in Algorithm 3 below. It generalizes the sequential subspace optimization (SESOP) algorithm of [31] to Riemannian manifolds.

ALGORITHM 3. SESOP

Require: Riemannian manifold M; continuously differentiable scalar field f on M; retraction R from TM to M as in Definition 3.1.
Input: Initial iterate $x_0 \in M$.
Output: Sequence of iterates $\{x_k\} \subseteq M$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Select a subspace $\mathcal{S}_k \subseteq T_{x_k}M$.
3: Find $\xi_k = \arg\min_{\xi \in \mathcal{S}_k} f(R_{x_k}(\xi))$.
4: Set $x_{k+1} = R_{x_k}(\xi_k)$.
5: **end for**

If $\mathcal{S}_k$ is chosen in step 2 such that $\mathcal{S}_k$ contains η_k, where η_k is as in Algorithm 1 (ALS) (resp., Algorithm 2 (ATR)), then Algorithm SESOP becomes an instance of Algorithm 1 (resp., Algorithm 2), with $c = 1$. The SESOP framework thus provides a strategy for accelerating line-search and trust-region methods.

When $M = \mathbb{R}^n$ with its natural retraction, Algorithm 3 becomes Algorithm 4 below, which can be found in [31] in an almost identical formulation. Observe that

ALGORITHM 4. $\mathbb{R}^n$-SESOP

Require: Continuously differentiable scalar field f on $\mathbb{R}^n$.
Input: Initial iterate $x_0 \in \mathbb{R}^n$.
Output: Sequence of iterates $\{x_k\} \subseteq \mathbb{R}^n$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Select a real matrix W_k with n rows.
3: Find $y^* = \arg\min_y f(x + W_k y)$.
4: Set $x_{k+1} = x_k + W_k y^*$.
5: **end for**

if $x_k \in \mathrm{col}(W_k)$, where $\mathrm{col}(W)$ denotes the subspace spanned by the columns of W, then x_{k+1} admits the expression

$$x_{k+1} = \operatorname*{arg\,min}_{x \in \mathrm{col}(W_k)} f(x). \tag{24}$$

DEFINITION 7.1 (gradient-related sequence of subspaces). *A sequence $\{\mathcal{S}_k\}$ of subspaces of $T_{x_k}M$ is* gradient-related *if there exists a gradient-related sequence $\{\eta_k\}$ such that $\eta_k \in \mathcal{S}_k$ for all k; equivalently, for any subsequence $\{x_k\}_{k\in\mathcal{K}}$ that converges to a nonstationary point, we have*

$$\limsup_{k\to\infty,\ k\in K} \left(\inf_{\eta\in\mathcal{S}_k,\ \|\eta\|=1} \langle \operatorname{grad} f(x_k), \eta \rangle \right) < 0.$$

When $M = \mathbb{R}^n$, the condition that $\mathcal{S}_k$ be a subspace of $T_{x_k}M$ reduces to $\mathcal{S}_k$ being a subspace of $\mathbb{R}^n$ (in view of the canonical identification $T_x\mathbb{R}^n \simeq \mathbb{R}^n$).

PROPOSITION 7.2. *Let $\{x_k\}$ be an infinite sequence of iterates generated by Algorithm 3 (SESOP). Assume that the sequence $\{\mathcal{S}_k\}$ produced by Algorithm 3 is gradient-related (Definition 7.1). Then every limit point of $\{x_k\}$ is a stationary point of f. Assume further that $\{x_k\}$ is included in some compact set $\mathcal{C}$. Then* $\lim_{k\to\infty} \|\operatorname{grad} f(x_k)\| = 0$.

Proof. The proof is a direct consequence of the convergence analysis of Algorithm 1 (ALS). ∎

We now discuss a detailed procedure for selecting $\mathcal{S}_k$ in Algorithm 3 (SESOP). It generalizes an idea in [26], which can be traced back to [39]. We denote by $P_\gamma^{t\leftarrow t_0}\zeta$ the vector of $T_{\gamma(t)}M$ obtained by parallel transporting a vector $\zeta \in T_{\gamma(t_0)}M$ along a curve γ. We refer, e.g., to [13, 6] for details on parallel translation. In $\mathbb{R}^n$, the natural parallel translation is simply given by $P_\gamma^{t\leftarrow t_0}\zeta = \zeta$ (where the ζ on the left-hand side is viewed as an element of $T_{\gamma(0)}M$ and the ζ on the right-hand side is viewed as an element of $T_{\gamma(t)}M$).

The name *conjugate gradient* is justified by the following property. Let M be the Euclidean space $\mathbb{R}^n$ with retraction $R_x(\xi) := x + \xi$. Let f be given by $f(x) = \frac{1}{2}x^TAx$, where A is a symmetric positive-definite matrix. Then Algorithm 5 reduces to the classical linear CG method. This result is a consequence of the minimizing properties of the CG method. Again in the Euclidean case, but for general cost functions, Algorithm 5 can be viewed as a "locally optimal" nonlinear CG method: instead of computing a search direction ξ_k as a correction of $-\operatorname{grad} f(x_k)$ along ξ_{k-1} (as is done in classical CG methods), the vector ξ_k is computed as a minimizer over the space spanned by $\{-\operatorname{grad} f(x_k), \xi_{k-1}\}$. For the general Riemannian case, assuming that the retraction is chosen as the Riemannian exponential, Algorithm 5 can be thought of as a locally optimal version of the Riemannian CG algorithms proposed by Smith [34] (see also [14]).

By construction, the sequence $\{\mathcal{S}_k\}$ in Algorithm 5 is gradient-related. The following result thus follows from Proposition 7.2.

PROPOSITION 7.3. *Let $\{x_k\}$ be an infinite sequence of iterates generated by Algorithm 5. Then every limit point of $\{x_k\}$ is a stationary point of f. Assume further that $\{x_k\} \subseteq \mathcal{C}$ for some compact set $\mathcal{C}$. Then* $\lim_{k\to\infty} \|\operatorname{grad} f(x_k)\| = 0$.

This result still holds if the parallel transport in Algorithm 5 is replaced by any *vector transport* as defined in [6]; indeed, the sequence $\{\mathcal{S}_k\}$ is still gradient-related by construction. Moreover, we point out that since Algorithm 5 is based on CG, it tends to display fast local convergence.

ALGORITHM 5. ACCELERATED CONJUGATE GRADIENT (ACG)

Require: Riemannian manifold M; continuously differentiable scalar field f on M; retraction R from TM to M as in Definition 3.1.
Input: Initial iterate $x_0 \in M$.
Output: Sequence of iterates $\{x_k\}$.
1: $\xi_0 := 0$; $x_1 := x_0$;
2: **for** $k = 1, 2, \ldots$ **do**
3: Compute ξ_k as a minimizer of $\hat{f}_{x_k}$ over $\mathcal{S}_k := \operatorname{span}\left\{P_\gamma^{1\leftarrow 0}\xi_{k-1}, \operatorname{grad} f(x_k)\right\}$ where $\gamma(t) := R_{x_{k-1}}(t\,\xi_{k-1})$;
4: Compute $x_{k+1} = R_{x_k}(\xi_k)$;
5: **end for**

8. Applications. Several occurences of Algorithms 1 (ALS), 2 (ATR), and 3 (SESOP) appear in the literature, e.g., in [20], [31], and in several eigenvalue algorithms. Indeed, it is well-known that subspace acceleration can remarkably improve the efficiency of eigensolvers; see, for example, the numerical comparison in [6, Figure 4.3] between a steepest descent algorithm and an accelerated version thereof, equivalent to locally optimal block preconditioned conjugate gradient (LOBPCG). Since, moreover, subspace acceleration is easy to perform for the eigenvalue problem, there are few methods that do not exploit it.

In the context of this analysis paper, we will focus on showing that the theory developed in the previous sections leads to convergence results for certain well-known algorithms. Some of these convergence results are new, to the best of our knowledge. In other cases, we recover results that have already been established, but the acceleration-based proof technique is novel and arguably more streamlined.

8.1. Lanczos algorithm. In a Ritz-restarted Lanczos algorithm for computing the leftmost eigenpair of a symmetric matrix A, the next iterate x_{k+1} is chosen as a minimizer of the Rayleigh quotient over the subspace $\mathcal{K}_m(x_k) := \operatorname{span}\{x_k, Ax_k, A^2x_k, \ldots, A^m x_k\}$, $m \geq 1$. Recall that the Rayleigh quotient of A is the function

$$f : \mathbb{R}_0^n \to \mathbb{R} : x \mapsto \frac{x^T A x}{x^T x}.$$

Its stationary points are the eigenvectors of A, and at those points it takes the value of the corresponding eigenvalue. (Note, however, that $f(x) = \lambda_i$, where λ_i is an eigenvalue of A, does not imply that x is an eigenvector of A, unless λ_i is an extreme eigenvalue of A.) Since x_k belongs to $\mathcal{K}_m(x_k)$, we are in the situation (24), and thus the Ritz-restarted Lanczos algorithm is an instance of Algorithm 3 (SESOP) (specifically, of Algorithm 4 ($\mathbb{R}^n$-SESOP)). The gradient of the Rayleigh quotient at x_k is collinear with $Ax_k - f(x_k)x_k$, which belongs to $\mathcal{K}_m(x_k)$, and hence $\{\mathcal{K}_m(x_k)\}$ is gradient-related to $\{x_k\}$. It follows from Theorem 7.2 that every limit point of $\{x_k\}$ is an eigenvector of A, regardless of x_0. Taking into account the properties of the Rayleigh quotient f along with the fact that $\{x_k\}$ is a descent sequence for f, it follows that $\{x_k\}$ converges to the eigenspace associated to an eigenvalue of A. The same conclusion holds for the Ritz-restarted Krylov method proposed by Golub and Ye [19] for the symmetric definite generalized eigenvalue problem. In other words, we recovered [19, Theorem 3.2].

8.2. LOBPCG. Knyazev's LOBPCG method [26], in combination with a symmetric positive-definite preconditioner, is a popular algorithm for computing approx-

imations to the smallest eigenvalues and eigenvectors of the eigenproblem

$$Au = Bu\lambda,$$

where A and B are real symmetric positive-definite matrices of order n. Here we consider LOBPCG as formulated in [21, Algorithm 1] (with some changes in the notation), and we show, using Theorem 4.3, that the limit points of $\{\mathrm{col}(X_k)\}$ are invariant subspaces of the pencil (A, B). Moreover, invariant subpaces that do not correspond to the smallest eigenvalues are "unstable," in the sense explained below.

The LOBPCG algorithm is described in Algorithm 6. In the algorithm, $(Y, \Theta) = \mathrm{RR}(S, p)$ performs a Rayleigh–Ritz analysis where the pencil (S^TAS, S^TBS) has eigenvectors Y and eigenvalues Θ, i.e.,

$$S^TASY = S^TBSY\Theta \quad \text{and} \quad Y^TS^TBSY = I_{b\times b},$$

where $I_{b\times b}$ is the identity matrix of size $b \times b$. The first p pairs with smallest Ritz values are returned in Y and in the diagonal matrix Θ in a nondecreasing order. Note that we consider the formulation [21, Algorithm 1] because it is simple to state and comprehend. However, it should be kept in mind that the matrix $[X_k, H_k, P_k]$ may become singular or ill-conditioned [21]. Therefore, in practical implementations, it is recommended to rely on the robust representation given in [21, Algorithm 2]. The convergence results obtained below also hold in this case.

Algorithm 6. LOBPCG [26, 21] without soft-locking

Require: Symmetric positive-definite matrices A and B of order n; symmetric positive-definite preconditioner N; block-size p.

Select an initial guess $\tilde{X} \in \mathbb{R}^{n\times p}$.
$X_0 = \tilde{X}Y$ where $(Y, \Theta_0) = \mathrm{RR}(\tilde{X}, p)$.
$R_k = AX_0 - MX_0\Theta_0$.
$P_k = [\,]$.
for $k = 0, 1, 2, \ldots$ **do**
  Solve the preconditioned linear system $NH_k = R_k$.
  Let $S = [X_k, H_k, P_k]$ and compute $(Y_k, \Theta_{k+1}) = \mathrm{RR}(S, p)$.
  $X_{k+1} = [X_k, H_k, P_k]Y_k$.
  $R_{k+1} = AX_{k+1} - MX_{k+1}\Theta_{k+1}$.
  $P_{k+1} = [0, H_k, P_k]Y_k$.
end for

In the case $p = 1$, it takes routine manipulations to check, using Proposition 7.2 with the Rayleigh quotient as the cost function, that all of the limit points of $\{X_k\}$ are eigenvectors of the pencil (A, B). We now consider the general case $p \geq 1$ in detail.

Let $\mathbb{R}_*^{n\times p}$ denote the set of all full-rank $n \times p$ real matrices. Observe that $\mathbb{R}_*^{n\times p}$ is an open subset of $\mathbb{R}^{n\times p}$ (it is thus an open submanifold of the linear manifold $\mathbb{R}^{n\times p}$, see [6]) and that $T_X\mathbb{R}_*^{n\times p} \simeq \mathbb{R}^{n\times p}$ for all $X \in \mathbb{R}_*^{n\times p}$. In $\mathbb{R}_*^{n\times p}$, consider the inner product defined by

$$\langle Z_1, Z_2\rangle_X = 2\,\mathrm{trace}\left(\left(X^TBX\right)^{-1} Z_1^TZ_2\right), \quad X \in \mathbb{R}_*^{n\times p},\ Z_1, Z_2 \in T_X\mathbb{R}_*^{n\times p}. \tag{25}$$

(The factor of 2 is included here to prevent factors of 2 from appearing in the formula of the gradient below. This is still a valid inner product, and it turns $\mathbb{R}_*^{n\times p}$ into a

Riemannian manifold.) Consider the cost function

$$f : \mathbb{R}_*^{n\times p} \to \mathbb{R} : X \mapsto \operatorname{trace}\left(\left(X^TBX\right)^{-1} X^TAX\right). \tag{26}$$

This generalized Rayleigh quotient was studied, e.g., in [6] (when $B = I$, it reduces to the *extended Rayleigh quotient* of [22]). It satisfies the property $f(XW) = f(X)$ for all $X \in \mathbb{R}_*^{n\times p}$ and all W invertible of size $p \times p$. A matrix $X \in \mathbb{R}_*^{n\times p}$ is a stationary point of f if and only if its column space is an invariant subspace of the pencil (A, B). The value of f at an invariant subspace is the sum of the corresponding eigenvalues. The stationary points whose column space is the rightmost invariant subspace of (A, B) (i.e., the one corresponding to the largest eigenvalues) are global maximizers of f. The stationary points whose column space is the leftmost invariant subspace of (A, B) (i.e., the one corresponding to the smallest eigenvalues) are global minimizers of f. All of the other stationary points are saddle points.

The fact that $\mathbb{R}_*^{n\times p}$ is $\mathbb{R}^{n\times p}$ with infinitely many elements excerpted makes it difficult to view LOBPCG as an instance of Algorithm 3 (SESOP). Instead, we view it as an instance of Algorithm 1 (ALS). The gradient of f with respect to the Riemannian metric (25) is

$$\operatorname{grad} f(X) = AX - BX\left(X^TBX\right)^{-1} X^TAX;$$

see, e.g., [6, equation (6.37)]. Referring to Algorithm 6, we have $H_k = N^{-1}\operatorname{grad} f(X_k)$ and

$$\langle \operatorname{grad} f(X_k), -H_k\rangle_{X_k} = \left\| N^{-\frac{1}{2}} \operatorname{grad} f(X_k)\right\|_F^2,$$

from which it follows that $\{-H_k\}$ is gradient-related to $\{X_k\}$ (Definition 4.1). We consider the retraction given by $R_X(Z) = X + Z$, $X \in \mathbb{R}_*^{n\times p}$, $Z \in T_X\mathbb{R}_*^{n\times p} \simeq \mathbb{R}^{n\times p}$. The Armijo point along $-H_k$ takes the form

$$X_{k+1/2} = X_k - \alpha_k H_k$$

for some $\alpha_k > 0$. Hence

$$X_{k+1/2} = [X_k, H_k, P_k]Y$$

for some Y. Without preconditioning ($N = I$), $X_{k+1/2}$ is full-rank (i.e., it belongs to $\mathbb{R}_*^{n\times p}$) for any α_k. Indeed, we have that $X_k^TX_{k+1/2} = X_k^T(I-\alpha_kA)X_k+\alpha_kX_k^TAX_k = X_k^TX_k$ is full-rank. (Observe that all iterates are B-orthogonal, hence of full rank.) With the preconditioner, however, this property is no longer guaranteed. Nevertheless, given A, B and N symmetric positive-definite matrices of order n, it is possible to find $\overline{\alpha}$ such that $X - \alpha N^{-1}\operatorname{grad} f(X)$ has full rank for all B-orthonormal X and all $\alpha \in [0, \overline{\alpha}]$. (This is because $\{X \in \mathbb{R}^{n\times p} : X^TBX = I\}$ is a compact subset of $\mathbb{R}^{n\times p}$ and $\mathbb{R}^{n\times p} \setminus \mathbb{R}_*^{n\times p}$ is a closed subset of $\mathbb{R}^{n\times p}$ that do not intersect, and hence their distance does not vanish.) With this $\overline{\alpha}$, LOBPCG becomes an instance of Algorithm 1 (ALS), provided we show that the acceleration bound (5) holds for some $c > 0$. It does hold for $c = 1$, as a consequence of the following result.

LEMMA 8.1. *In the context of Algorithm* 6, *we have*

$$\begin{aligned} f(X_{k+1}) &= \min\left\{f([X_k, H_k, P_k]Y) : Y \in \mathbb{R}^{3p\times p}, Y^T[X_k, H_k, P_k]^TB[X_k, H_k, P_k]Y = I\right\} \\ &= \min\{f([X_k, H_k, P_k]Y) : Y \in \mathbb{R}^{3p\times p}, [X_k, H_k, P_k]Y \textit{ full rank}\}, \end{aligned}$$

where f denotes the Rayleigh quotient (26).

Proof. The three expressions are equal to the sum of the p leftmost eigenvalues of the pencil (U^TAU, U^TBU), where U is a full-rank matrix with $\mathrm{col}(U) = \mathrm{col}([X_k, H_k, P_k])$. □

This yields the following result.

PROPOSITION 8.2. *Let $\{X_k\}$ be a sequence of iterates generated by Algorithm* 6 *(LOBPCG). Then the following holds.*

(a) *Every limit point X_* of $\{X_k\}$ is a stationary point of f; i.e.,* $\mathrm{col}(X_*)$ *is an invariant subspace of (A, B);*

(b) $\lim_{k\to\infty} \|AX_k - BX_k\Theta_k\| = 0$, *where Θ_k is as in Algorithm* 6 *(LOBPCG);*

(c) *The limit points of* $\{\mathrm{col}(X_k)\}$ *are p-dimensional invariant subspaces of (A, B);*

(d) $\lim_{k\to\infty} f(X_k)$ *exists (where f is the generalized Rayleigh quotient* (26)*), and thus f takes the same value at all limit points of $\{X_k\}$.*

(e) *Let $\mathcal{V}$ be a limit point of* $\{\mathrm{col}(X_k)\}$ *that is not a leftmost invariant subspace of (A, B) ("leftmost" means related to the smallest eigenvalues). Then $\mathcal{V}$ is* unstable *in the following sense: there is $\epsilon > 0$ such that for all $\delta > 0$ there exists $K > 0$ and $Z \in \mathbb{R}^{n\times p}$, with $\|Z\| < \delta$, such that if X_K is perturbed to $X_K + Z$ and the algorithm is pursued from this new iterate, then the new sequence satisfies $\angle(\mathrm{col}(X_k), \mathcal{V}) > \epsilon$ for all but finitely many iterates.*

Proof. Point (a) follows from Proposition 4.3 as explained above. Point (b) follows from Corollary 4.4 since all iterates belong to the compact set $\{X \in \mathbb{R}^{n\times p} : X^TBX = I\}$. Note that $\mathrm{grad}\, f(X_k) = AX_k - BX_k\Theta_k$. Point (c) involves the topology of the quotient manifold. The result follows from the fact that the col mapping is continuous from $\mathbb{R}^{n\times p}_*$ to the Grassmann manifold of p-planes in $\mathbb{R}^n$. (The topology of the Grassmann manifolds is precisely the one that makes the col mapping continuous; see, e.g., [6] for details.) Point (d) holds because LOBPCG is a descent method for f. Point (e) can be deduced from the fact that the non-leftmost invariant subspaces of (A, B) are saddle points or maxima for f and from the fact that LOBPCG is a descent method for f. □

8.3. Jacobi–Davidson methods. The Jacobi–Davidson algorithm for computing the smallest eigenvalue and eigenvector of an $n \times n$ symmetric matrix A, as described in [38, Algorithm 1], clearly fits within Algorithm 3 (SESOP). However, without further assumptions, it is not guaranteed that $\{\mathcal{S}_k\}$ be gradient-related: it all depends on how the Jacobi correction equation is "approximately" solved. If the approximate solution can be guaranteed to be gradient-related, then it follows from Proposition 7.2 that all limit points are stationary points of the Rayleigh quotient; i.e., they are eigenvectors.

For example, consider, as in [28], the Jacobi equation in the form

$$\left(I - x_kx_k^T\right)(A - \tau I)\left(I - x_kx_k^T\right)\eta_k = -\left(I - x_kx_k^T\right)Ax_k, \quad x_k^T\eta_k = 0, \tag{27}$$

where τ is some target less than the smallest eigenvalue λ_1 of A, and assume that the approximate solution η_k is obtained with m_k steps of the CG iteration ($1 \le m_k < n$ for all k). We show that the sequence $\{\eta_k\}$ is gradient-related to $\{x_k\}$, and thus $\{\mathcal{S}_k\}$ is gradient-related to $\{x_k\}$ when $\mathcal{S}_k$ contains η_k for all k. By the workings of CG (with zero initial condition), η_k is equal to $V_{m_k}y_k$, where V_{m_k} is an orthonormal basis of the Krylov subspace $\mathcal{K}_{m_k}$ generated from $-(I - x_kx_k^T)Ax_k$ using the operator $(I - x_kx_k^T)(A - \tau I)(I - x_kx_k^T)$ and where y_k solves

$$V_{m_k}^T(A - \tau I)V_{m_k}y_k = -V_{m_k}^TAx_k. \tag{28}$$

Notice that the Krylov subspace is orthogonal to x_k and contains the gradient $(I -$

$x_k x_k^T)Ax_k$, and hence we have the identities $(I - x_k x_k^T)V_{m_k} = V_{m_k}$ and $V_{m_k} V_{m_k}^T Ax_k = V_{m_k} V_{m_k}^T (I - x_k x_k^T)Ax_k = (I - x_k x_k^T)Ax_k$. Since $A - \tau I$ is positive-definite, it follows that the condition number of the projected matrix $V_{m_k}^T (A - \tau I)V_{m_k}$ is bounded, and hence in view of (28) the angle between y_k and $-V_{m_k}^T Ax_k$ is bounded away from $\frac{\pi}{2}$, and so is the angle between $V_{m_k} y_k = \eta_k$ and $-V_{m_k} V_{m_k}^T Ax_k = -(I - x_k x_k^T)Ax_k$ because V_{m_k} is an orthonormal basis. Moreover, $\{y_k\}$ is bounded away from zero and infinity, and so is $\{\eta_k\}$. We have thus shown that the sequence $\{\eta_k\}$ is gradient-related to $\{x_k\}$ (see the discussion that follows Definition 4.1). Thus Proposition 8.2 holds, mutatis mutandis, for the Jacobi–Davidson method [38, Algorithm 1] when the Jacobi equation (27) is defined and solved approximately with CG as in [28].

The result still holds when the CG iteration for (approximately) solving (27) is preconditioned with a positive-definite preconditioner N_k. Indeed, the preconditioned CG for solving a linear system $B\eta = -g$ amounts to applying the "regular" CG method to the transformed system $\tilde{B}\tilde{\eta} = -\tilde{g}$, where $\tilde{B} = N^{-1}BN^{-1}$, $\tilde{\eta} = N\eta$, and $\tilde{g} = N^{-1}g$ (see, e.g., [18, section 10.3]). If $\tilde{\eta}_j$ is an iterate of the regular CG applied to $\tilde{B}\tilde{\eta} = -\tilde{g}$ and thus $\eta_j = N^{-1}\tilde{\eta}_j$ is the iterate of the preconditioned CG, then we have $\langle \tilde{\eta}_j, \tilde{g} \rangle = \langle N\eta_j, N^{-1}g \rangle = \langle \eta_j, g \rangle$. Thus the sequence $\{\eta\}_k$, where η_k is the approximate solution of (27) returned by the preconditioned CG, is gradient-related.

Note that the choice of τ to make $(A - \tau I)$ positive-definite in (27) is crucial in the development above. In the frequently encountered case where τ is selected as the Rayleigh quotient θ_k at x_k, it seems difficult to provide a theoretical guarantee that the approximate solution η_k of (27) is gradient-related, unless we assume that the iteration starts close enough to the minor eigenvector so that $(I - x_k x_k^T)(A - \theta_k I)(I - x_k x_k^T)$ is positive definite as a linear transformation of the orthogonal complement of x_k. (An example of the requirement that the iteration start sufficiently close to the minor eigenvector is the condition $\theta_k < \frac{\lambda_1 + \lambda_2}{2}$ in [29, Theorem 4.3].) However, in practice, it is quite clear that a solver producing a sequence $\{\eta_k\}$ that is not gradient-related would have to be particularly odd. It is thus not surprising that the global convergence properties stated in Proposition 8.2 have been empirically observed in general for eigenvalue algorithms that fit in the Jacobi–Davidson framework.

Another example (which does not fit, strictly speaking, in the Jacobi–Davidson framework, but is closely related) is when, as in [1], the Jacobi equation is solved approximately using a truncated CG algorithm and the approximate solution is accepted or rejected using a trust-region mechanism. The method becomes an instance of Algorithm 2 applied to the Rayleigh quotient cost function, and Proposition 8.2 holds, mutatis mutandis.

8.4. Sequential subspace method. All of the algorithms thus far in this section are concerned with the eigenvalue problem; however, the area of application of the convergence theory developed in this paper is not restricted to eigenvalue solvers. An example is the SSM of Hager [20] for minimizing an arbitrary quadratic function over a sphere. This algorithm is an instance of Algorithm 3 (SESOP). In [20], $\{\mathcal{S}_k\}$ is required to contain $\operatorname{grad} f(x_k)$; therefore, all limit points are stationary by Proposition 7.2. This was proven in [25], where stronger global convergence results are obtained by making additional assumptions on $\{\mathcal{S}_k\}$.

9. Concluding remarks. If we accelerate, in the sense of (1), an optimization algorithm that converges globally to stationary points of the cost function, do we preserve the global convergence result? We have answered this question positively for a wide class of line-search and trust-region methods. The global convergence of several eigenvalue algorithms follows from this result, under mild conditions, as shown

in section 8. We suspect that several other existing methods satisfy the conditions of the global convergence theorems proven in this paper.

An important practical issue in the design of accelerated algorithms is to strike a good balance of the workload between the "Jacobi-like" step (i.e., the computation of an update vector η_k) and the "Davidson-like" step (i.e., the improvement on the Jacobi update, for example, via a minimization within a subspace containing η_k). For example, at one extreme, the simplified Jacobi–Davidson in [28] simply turns off the Davidson step. Note that the algorithm in [8], where the "Jacobi" step consists of solving approximately a certain trust-region-like problem, shows promising numerical results even without using a "Davidson" step. At the other extreme, the workings of the the Jacobi–Davidson approach [38] can be exploited to let the Davidson step compensate for a crude approximation of the Jacobi update. In LOBPCG, the balance of the workload between the Jacobi-like step (computation of H_k) and the Davidson-like step (computation of X_{k+1} from $[X_k, H_k, P_k]$ by a Ritz process) depends much on the complexity of the chosen preconditioner; we refer, e.g., to [5, 27] for more information on preconditioners in LOBPCG. Note that in an eigenvalue method for a matrix A, the structure of A and the nature of the preconditioner will affect the computational burden on the Jacobi-like step, whereas the Davidson-like step, if implemented efficiently, should require only some orthogonalization routines and be largely independent of the cost of the operators. Hence, when the operators are inexpensive, it becomes more affordable to require a higher accuracy in the Jacobi-like step. We refer to [35, 24, 23, 36] for further work along these lines.

Finally, we point out that there is not necessarily a unique way of separating the instructions of an iterative loop into a Jacobi-like step and a Davidson-like step that satisfy the conditions for the global convergence analysis. For example, the application of a preconditioner can be considered as part of the Jacobi-like step or as part of the acceleration step if the preconditioning leads to an acceleration bound (1).

Acknowledgments. This work benefited in particular from discussions with Chris Baker, Bill Hager, Ekkehard Sachs, and Gerard Sleijpen. Special thanks to Chris Baker for his helpful comments on the manuscript.

REFERENCES

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

[2] P.-A. Absil, C. G. Baker, and K. A. Gallivan, *Trust-region methods on Riemannian manifolds*, Found. Comput. Math., 7 (2007), pp. 303–330.

[3] 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), pp. 359–390.

[4] P.-A. Absil and K. A. Gallivan, *Accelerated Line-search and Trust-region Methods*, Technical report FSU-SCS-2005-095, School of Computational Science, Florida State University, Tallahassee, FL, 2005.

[5] P. Arbenz, U. L. Hetmaniuk, R. B. Lehoucq, and R. S. Tuminaro, *A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods*, Internat. J. Numer. Methods Engrg., 64 (2005), pp. 204–236.

[6] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2008.

[7] L. Armijo, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3.

[8] C. G. Baker, P.-A. Absil, and K. A. Gallivan, *An implicit trust-region method on Riemannian manifolds*, IMA J. Numer. Anal., to appear.

[9] C. G. Baker, *Riemannian Manifold Trust-region Methods with Applications to Eigenproblems*, Ph.D. thesis, School of Computational Science, Florida State University, Tallahassee, FL, 2008.
[10] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.
[11] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, MPS/SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000.
[12] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices*, J. Comput. Phys., 17 (1975), pp. 87–94.
[13] M. P. do Carmo, *Riemannian geometry*, Math. Theory Appl., Birkhäuser Boston, Boston, MA, 1992. Translated from the second Portuguese edition by Francis Flaherty.
[14] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.
[15] D. R. Fokkema, G. L. G. Sleijpen, and H. A. Van der Vorst, *Accelerated inexact Newton schemes for large systems of nonlinear equations*, SIAM J. Sci. Comput., 19 (1998), pp. 657–674.
[16] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Jacobi–Davidson style QR and QZ algorithms for the reduction of matrix pencils*, SIAM J. Sci. Comput., 20 (1998), pp. 94–125.
[17] D. Gabay, *Minimizing a differentiable function over a differential manifold*, J. Optim. Theory Appl., 37 (1982), pp. 177–219.
[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins S. Math. Sci., Johns Hopkins University Press, Baltimore, MD, 1996.
[19] G. H. Golub and Q. Ye, *An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems*, SIAM J. Sci. Comput., 24 (2002), pp. 312–334.
[20] W. W. Hager, *Minimizing a quadratic over a sphere*, SIAM J. Optim., 12 (2001), pp. 188–208.
[21] U. Hetmaniuk and R. Lehoucq, *Basis selection in LOBPCG*, J. Comput. Phys., 218 (2006), pp. 324–332.
[22] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*, Comm. Control Engrg. Ser., Springer-Verlag, London, 1994.
[23] M. E. Hochstenbach and Y. Notay, *Controlling Inner Iterations in the Jacobi-Davidson Method*, SIAM J. Matrix Anal. Appl., to appear.
[24] M. E. Hochstenbach and Y. Notay, *The Jacobi-Davidson method*, GAMM Mitt. Ges. Angew. Math. Mech., 29 (2006), pp. 368–382.
[25] W. W. Hager and S. Park, *Global convergence of SSM for minimizing a quadratic over a sphere*, Math. Comp., 74 (2005), pp. 1413–1423.
[26] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.
[27] I. Lashuk, M. Argenti, E. Ovtchinnikov, and A. Knyazev, *Preconditioned eigensolver LOBPCG in Hypre and PETSc*, in Domain Decomposition Methods in Science and Engineering XVI, Lect. Notes Comput. Sci. Eng. 55, Springer-Verlag, Berlin, 2007.
[28] Y. Notay, *Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem*, Numer. Linear Algebra Appl., 9 (2004), pp. 21–44.
[29] Y. Notay, *Is Jacobi–Davidson faster than Davidson?*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 522–543.
[30] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Ser. Oper. Res., Springer-Verlag, New York, 1999.
[31] G. Narkiss and M. Zibulevsky, *Sequential Subspace Optimization Method for Large-Scale Unconstrained Problems*, Technical report CCIT 559, EE Dept., Technion, Haifa, Israel, 2005.
[32] M. Shub, *Some remarks on dynamical systems and numerical analysis*, in Dynamical Systems and Partial Differential Equations, Proceedings of the VII ELAM, L. Lara-Carrero and J. Lewowicz, eds., Equinoccio, Universidad Simón Bolívar, Caracas, 1986, pp. 69–91.
[33] S. T. Smith, *Geometric Optimization Methods for Adaptive Filtering*, Ph.D. thesis, Division of Applied Sciences, Harvard University, Cambridge, MA, 1993.
[34] S. T. Smith, *Optimization techniques on Riemannian manifolds*, in Hamiltonian and Gradient Flows, Algorithms and Control, Fields Inst. Commun. 3, American Mathematical Society, Providence, RI, 1994, pp. 113–136.
[35] A. Stathopoulos and Y. Saad, *Restarting techniques for the (Jacobi-)Davidson symmetric eigenvalue methods*, Electron. Trans. Numer. Anal., 7 (1998), pp. 163–181.
[36] A. Stathopoulos, *Nearly optimal preconditioned methods for Hermitian eigenproblems under limited memory. Part* I*: Seeking one eigenvalue*, SIAM J. Sci. Comput., 29 (2007), pp. 481–514.

[37] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.
[38] G. L. G. Sleijpen and H. A. Van der Vorst, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.
[39] I. Takahashi, *A note on the conjugate gradient method*, Inform. Process. Japan, 5 (1965), pp. 45–49.
[40] C. Udrişte, *Convex Functions and Optimization Methods on Riemannian Manifolds*, Math. Appl. 297, Kluwer Academic, Dordrecht, the Netherlands, 1994.
[41] Y. Yang, *Globally convergent optimization algorithms on Riemannian manifolds: Uniform framework for unconstrained and constrained optimization*, J. Optim. Theory Appl., 132 (2007), pp. 245–265.