



HAL
open science

Riemannian trust-region methods for strict saddle functions with complexity guarantees

Florentin Goyens, Clément W. Royer

► **To cite this version:**

Florentin Goyens, Clément W. Royer. Riemannian trust-region methods for strict saddle functions with complexity guarantees. 2024. hal-04397931v1

HAL Id: hal-04397931

<https://hal.science/hal-04397931v1>

Preprint submitted on 18 Jan 2024 (v1), last revised 13 Feb 2024 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Riemannian trust-region methods for strict saddle functions with complexity guarantees

Florentin Goyens^{1,2} and Clément W. Royer¹

¹LAMSADE, CNRS, Université Paris Dauphine-PSL, Place du Maréchal de Lattre de Tassigny, 75016 Paris, France.

²Corresponding author: goyensflorentin@gmail.com
Cite this version on <https://hal.science/hal-04397931>

January 18, 2024

Abstract

For a function with strict saddle points and strong convexity near minimizers, we show that the classical trust-region algorithm with exact subproblem minimization finds approximate (local) minimizers in polynomial-time. The accuracy parameter ε only appears in a $\log \log(\varepsilon^{-1})$ term and, in that sense, the complexity is essentially independent of the accuracy ε . Using a second-order model allows to benefit from the quadratic convergence rate of Newton's method near minimizers. This is a notable improvement over known results for generic nonconvex functions, where the worst-case number of iterations to find an approximate second-order critical point is of the order $\mathcal{O}(\varepsilon^{-3})$. Our complexity bound depends polynomially on the inverse of landscape parameters which represent negative curvature and strong convexity constants. In the context where these landscape parameters are known, we present a strict saddle trust-region method with similar complexity guarantees which minimizes the trust-region subproblem approximately. It is based on a truncated conjugate gradient scheme and uses the Lanczos method to compute directions of negative curvature. Our complexity result gives an upper bound on the total number of gradient evaluations and Hessian-vector products. All our results apply to the setting where the function is defined on a Riemannian manifold, and Riemannian algorithms are used.

Keywords: Riemannian optimization, strict saddle function, second-order method, complexity.

1 Introduction

We consider the optimization problem

$$\min_{x \in \mathcal{M}} f(x) \tag{P}$$

where \mathcal{M} is a Riemannian manifold, and $f: \mathcal{M} \rightarrow \mathbb{R}$ is twice continuously differentiable and nonconvex.

A popular way to solve Problem (P) is to use a Riemannian optimization method. These methods use differential geometry to generalize unconstrained optimization methods to minimization over smooth manifolds (Absil et al., 2008; Boumal, 2023). While convergence guarantees have historically focused on the behaviour of algorithms near minimizers (local convergence), the past decade has seen a great interest in global complexity results. They give an

upper bound on the number of iterations that algorithms perform before termination, starting from an arbitrary initial guess $x_0 \in \mathcal{M}$. These guarantees are based on approximate satisfaction of first- and second-order necessary conditions for stationarity. For a smooth function $f: \mathcal{M} \rightarrow \mathbb{R}$, second-order critical points have a zero Riemannian gradient and positive semi-definite Riemannian Hessian:

$$\operatorname{grad}f(x) = 0 \quad \text{and} \quad \operatorname{Hess}f(x) \succeq 0. \quad (1.1)$$

In practice, given positive tolerances $(\varepsilon_g, \varepsilon_H)$, the target points of our algorithms satisfy

$$\|\operatorname{grad}f(x)\| \leq \varepsilon_g \quad \text{and} \quad \lambda_{\min}(\operatorname{Hess}f(x)) \geq -\varepsilon_H, \quad (1.2)$$

where $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue of a symmetric operator.

For a general nonconvex function f , complexity guarantees for Riemannian optimization methods are of the order of $\mathcal{O}(\varepsilon^{-3})$ iterations to find an $(\varepsilon, \varepsilon)$ -second-order critical point (Boumal et al., 2019; Agarwal et al., 2021). These results are pessimistic in nature and do not reflect the good practical behaviour of second-order methods.

In an effort to reconcile theoretical guarantees with performance in practice, it becomes necessary to leverage additional structure in the function f . Many problems of the form (P) satisfy a *strict saddle* property (see Definition 2.4). It ensures that the Riemannian Hessian of f has a negative eigenvalue at every critical point that is not a local minimizer. This makes it tractable to find approximate local minimizers using local optimization algorithms, which is too difficult to guarantee for general nonconvex optimization.

Strict saddle properties hold in applications which include Burer-Monteiro factorizations of semidefinite programs (Boumal et al., 2020; Luo and Trillos, 2022), phase retrieval (Sun et al., 2018), matrix completion and factorization (Ge et al., 2016; Li et al., 2019), dictionary learning (Sun et al., 2017a; Qu et al., 2019) and others (Wright and Ma, 2022). The results listed have the additional property that all local minimizers are global minimizers. For instance, Boumal et al. (2020) show that all second-order critical points (1.1) are global minimizers for smooth Burer-Monteiro factorizations. (This implies that all saddle points have a negative eigenvalue.) These properties are often grouped under the umbrella term of *benign nonconvexity*.

It follows that an optimization algorithm which is able to “escape” strict saddle points is guaranteed to find a global minimizer. There are two families of algorithms which provably escape saddle points.

The first kind uses noise injection to perturb iterates and prove that, with high probability, the algorithm does not terminate near a strict saddle point. These are usually first-order gradient descent methods (Lee et al., 2019). They may be impractical to implement, as the noise level must be chosen carefully according to problem constants (Jin et al., 2017). First-order methods with noise injection to avoid saddle points on manifolds are described in (Crisitiello and Boumal, 2019; Sun et al., 2019).

The second kind are second-order methods that directly compute search directions in which the Hessian has a negative curvature. Their framework naturally exploits the negative curvature of strict saddle points, which leads to deterministic results and simple proofs (Wright and Recht, 2022, Section 3.6). Computing directions of negative curvature can be expensive, but practical implementations compute these directions using an iterative method based on Hessian-vector products, which is cheaper than a complete eigenvalue decomposition (Cartis et al., 2012; Royer and Wright, 2018). Our work considers methods of the second kind.

Complexity results based on landscape parameters instead of the accuracy ε appear in (Sun et al., 2018; O’Neill and Wright, 2023). In (Sun et al., 2018), the authors consider the phase retrieval problem and analyse the landscape of a least-squares cost function. They show that

the domain of the function is partitioned into three areas: large gradients, negative eigenvalues of the Hessian and neighbourhood of minima with strong local convexity. They propose a second-order algorithm for the minimization of such function and derive a worst-case number of iteration, where the accuracy ε only influences the local quadratic convergence. The algorithm is presented as a trust-region method with exact subproblem minimization where the trust-region radius does not change during the iterations. The trust-region radius must be initialized with a small value depending on problem constants that are difficult to know a priori. Similar results are obtained for the complete dictionary recovery problem in (Sun et al., 2017b).

In (O’Neill and Wright, 2023), the authors consider low-rank optimization problems for which saddle points are strict. They consider a first-order algorithm with additional computations of negative curvature directions of the Hessian to avoid saddle points. The algorithm uses parameters of the landscape to compute search directions. If the parameters are unknown, it estimates them dynamically. Under a regularity condition near minimizers, they show local linear convergence of the algorithm. Carmon et al. (2018) propose an accelerated gradient method with negative curvature detection and give convergence rate for strict saddle functions. Liu and Roosta (2023) define a relaxation of the strict saddle property, and Paternain et al. (2019) propose a Newton-like method for the unconstrained minimization of strict saddle functions. Each iteration computes a full eigenvalue decomposition of the Hessian and take the absolute value of eigenvalues. Similarly to (Sun et al., 2018), they show a worst-case complexity that depends on landscape parameters. Curtis and Robinson (2021) introduce the notion of regional complexity. They bound the number of iterations that can be performed in regions with large gradient or negative curvature.

Contributions

In this work, we study two versions of the Riemannian trust-region method (Absil et al., 2007): one with exact minimization of the subproblems, and one where the subproblems are minimized inexactly by an iterative procedure based on truncated conjugate gradients. For strict saddle functions, we provide theoretical guarantees which are closer to the true behaviour of the algorithms, and do not suffer from a crippling dependency in the accuracy ε . In our results, the accuracy ε only plays a role in the local convergence, near the minimizer. (This corresponds to what happens in practice.) The strength of our results is to use second-order methods which benefit from the local quadratic convergence of Newton’s method. Hence, quadratic convergence triggers the termination condition in only a handful of iterations, even for the smallest values of ε used in practice. In that sense, we say that the complexity results are independent of ε . The complexity estimates therefore depend on landscape parameters of the function f . These parameters—which we define below—quantify the negative curvature at strict saddle points and the strong convexity constant near minimizers (see Definition 2.4).

Our complexity result for the Riemannian trust-region with exact subproblem minimization applies to the original algorithm, without any modification. This algorithm benefits from the improved complexity without needing to know the landscape parameters and without using a scheme to estimate them. We then propose a new algorithm, a Riemannian trust-region with inexact subproblem minimization that uses the landscape parameters to compute directions that are appropriate for the local landscape. For this algorithm, the guarantees measure the *operation complexity*: the number of gradient evaluations and Hessian-vector products before termination.

Our results apply naturally to the case $\mathcal{M} = \mathbb{R}$. Many applications of strict saddle optimization include optimization over smooth constraints. Our work presents the first practical second-order algorithm that comes with complexity guarantees for strict saddle functions on

manifolds. It is also the first such analysis that applies to a generic manifold \mathcal{M} .

Outline: In Section 2, we introduce strict saddle functions on a Riemannian manifolds. In Section 3, we show that the classical Riemannian trust-region with exact subproblem minimization achieves a worst-case complexity rate which is independent of ε_H and depends on ε_g in a log log term. For a formal statement, see Theorem 3.18. In Section 4, we design a new RTR algorithm with inexact minimization of the subproblems which exploits the strict saddle structure. Finally, we summarize our work and discuss perspectives in Section 5.

2 Strict saddle functions on Riemannian manifolds

In this section, we provide a formal definition of the strict saddle property over Riemannian manifolds. To this end, we first present background material on Riemannian optimization in Section 2.1, with a focus on Lipschitz continuity. We then discuss the notion of geodesic strong convexity in Section 2.2, which plays a key role in our definition of strict saddle functions. This definition is provided along with several examples in Section 2.3. For simplicity, we state all the necessary definitions and properties using our function f and our manifold \mathcal{M} of interest.

2.1 Displacements and derivatives on Riemannian manifolds

We begin by a summary of the key concepts from Riemannian optimization that will be of use in this paper. Recall that problem (P) considers the optimization of a smooth function f over a Riemannian manifold \mathcal{M} . A key component of manifold optimization consists in introducing proper notions of displacement on the manifold. On the manifold, the notion of shortest path between two points can be expressed through that of geodesic. A *geodesic* of the manifold \mathcal{M} is a smooth curve $c: I \rightarrow \mathcal{M}$ defined on an open interval $I \subset \mathbb{R}$ such that $c''(t) = 0$ for all $t \in I$, where $c''(t)$ is the intrinsic acceleration of c defined by the covariant derivative (we refer to (Boumal, 2023, Chapter 5) for details regarding the covariant derivative). Moving along the geodesics thus guarantees to remain on the manifold.

On the other hand, optimization algorithms such as those used in this paper typically produce steps that lie outside of the manifold, in which case one must define a way to map these steps to the manifold. To this end, steps are produced within tangent spaces, and retractions are then applied to generate a point on the manifold. Formally, given a point $x \in \mathcal{M}$, the *tangent space* $T_x\mathcal{M}$ is the vector space of vectors tangent to the manifold. We equip this vector space with the inner product $\langle \cdot, \cdot \rangle_x$ (which we may write $\langle \cdot, \cdot \rangle$ when the point of interest is clear from context). One also defines a *retraction* at x , that maps the tangent space to the manifold, denoted by $R_x: T_x\mathcal{M} \rightarrow \mathcal{M}$. A classical example of such retraction is the exponential map, that follows a geodesic defined by x and s . In this paper, however, we allow for a more generic choice of retraction, with the following requirement.

A1 (Assumption 6.9 in (Boumal, 2023)). *There exists positive constants ν_S, κ_S such that for all $(x, s) \in \mathcal{M} \times T_x\mathcal{M}$, if $\|s\|_x \leq \nu_S$, then $\text{dist}(x, R_x(s)) \leq \kappa_S \|s\|_x$, where $\text{dist}(\cdot, \cdot)$ is the Riemannian distance on \mathcal{M} .*

Assumption 1 will be instrumental in controlling the distance between iterates of our algorithms.

Given a retraction, one can lift the function f to the tangent space through the following composition.

Definition 2.1. For any $x \in \mathcal{M}$, the pullback of f to the tangent space $T_x\mathcal{M}$ is the function $\hat{f}_x: T_x\mathcal{M} \rightarrow \mathbb{R}$ defined by

$$\forall s \in T_x\mathcal{M}, \quad \hat{f}_x(s) := f \circ R_x(s).$$

Since the function f is twice continuously differentiable over the manifold, we can define a *Riemannian gradient* and *Riemannian Hessian* of f at any $x \in \mathcal{M}$, which we denote by $\text{grad}f(x)$ and $\text{Hess}f(x)$, respectively. By contrast, we use the symbols ∇ and ∇^2 to the gradient and Hessian of a given function over a Euclidean space. In particular, given $x \in \mathcal{M}$ and $s \in T_x\mathcal{M}$, we will consider the gradient of the pullback function $\nabla\hat{f}_x(s) \in T_x\mathcal{M}$ as well as the Hessian $\nabla^2\hat{f}_x(s): T_x\mathcal{M} \rightarrow T_x\mathcal{M}$. Note that we distinguish these derivatives from the Riemannian derivatives of f at $R_x(s)$, denoted by $\text{grad}f(R_x(s))$ and $\text{Hess}f(R_x(s))$. However, the identification $\hat{f}_x(0) = f(x)$ and $\nabla\hat{f}_x(0) = \text{grad}f(x)$ holds by definition (Boumal, 2023, Proposition 3.59). Under additional assumptions on the retraction, we can also relate the second-order derivatives. Assumption 2 below states our main requirement.

We also require that the retraction is a second-order approximation of geodesics.

A2. The retraction mapping is a second-order retraction, i.e. for any $(x, s) \in \mathcal{M} \times T_x\mathcal{M}$, the curve $c: t \in [0, 1] \rightarrow R_x(ts)$ has zero acceleration at $t = 0$, that is $c''(0) = 0$.

Provided a second-order retraction is used, one can show that

$$\nabla^2\hat{f}_x(0) = \text{Hess}f(x) \quad \forall x \in \mathcal{M}, \tag{2.1}$$

i.e. the Hessian of the pullback function corresponds to the Riemannian Hessian of f (Boumal, 2023, Proposition 5.45).

Remark 2.1. In this paper, we have elected to use a more general retraction, which requires certain smoothness assumptions on the pullback function. However, when the exponential map is used, a different reasoning based on the manifold curvature can be employed Sun et al. (2019); Criscitiello and Boumal (2019), yielding a simple analysis. Still, our approach is more general (note that the exponential map satisfies Assumptions 1 and 2), and yields a analysis of our algorithms that resembles the Euclidean setting.

2.2 Geodesic convexity

We now provide the key definitions behind geodesic convexity, a concept that generalizes the notion of convexity in the Euclidean setting to functions defined over Riemannian manifolds. As suggested by the name, this notion is built upon that of geodesics. Geodesically convex sets and functions are indeed defined with respect to geodesics of \mathcal{M} as follows.

Definition 2.2. A subset S of \mathcal{M} is *geodesically convex* if, for every $x, y \in S$, there exists a geodesic segment $c: [0, 1] \rightarrow \mathcal{M}$ such that $c(0) = x$, $c(1) = y$ and $c(t)$ is in S for all $t \in [0, 1]$.

Definition 2.3. Given a subset S of \mathcal{M} , the function f is *geodesically convex* on S (resp. *geodesically strongly convex*) if S is geodesically convex and for every geodesic $c: [0, 1] \rightarrow \mathcal{M}$ such that $c(0) \neq c(1)$ and $c([0, 1]) \subset S$, the function $f \circ c: [0, 1] \rightarrow \mathbb{R}$ is convex (resp. strongly convex).

From Definition 2.3, we can view geodesic convexity as convexity over geodesics. Interestingly, in the case of a smooth function, one can show that geodesic strong convexity is determined by the eigenvalues of the Riemannian Hessian.

Proposition 2.1 (Theorem 11.23 in (Boumal, 2023)). *A function $f: \mathcal{M} \rightarrow \mathbb{R}$ is geodesically γ -strongly convex on the set $S \subset \mathcal{M}$ if S is a geodesically convex set and $\lambda_{\min}(\text{Hess}f(x)) \geq \gamma$ for every $x \in S$.*

Note that we will consider geodesic strong convexity over proper subsets of the manifold, since we are interested in nonconvex problems. Geodesic convex functions over the entire manifold have however been investigated, with examples arising from the use of Hadamard manifolds (Zhang and Sra, 2016).

2.3 Strict saddle property

We are now ready to define our function class of interest on the manifold \mathcal{M} , namely that of *strict saddle* functions on \mathcal{M} . Our definition follows that of (Sun et al., 2015).

Definition 2.4. *The function f is $(\alpha, \beta, \gamma, \delta)$ -strict saddle if the manifold \mathcal{M} satisfies $\mathcal{M} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$, where*

$$\mathcal{R}_1 = \{x \in \mathcal{M} : \|\text{grad}f(x)\| \geq \alpha\}$$

$$\mathcal{R}_2 = \{x \in \mathcal{M} : \lambda_{\min}(\text{Hess}f(x)) \leq -\beta\}$$

$$\mathcal{R}_3 = \{x \in \mathcal{M} : \text{there exists a local minimum } x^* \in \mathcal{M} \text{ of } f \text{ such that } \text{dist}(x, x^*) \leq \delta \text{ and } f \text{ is geodesically } \gamma\text{-strongly convex over the set } \{y \in \mathcal{M} : \text{dist}(x^*, y) < 2\delta\}\}.$$

Definition 2.4 has the following interpretation. If f is a strict saddle function on \mathcal{M} , then for any $x \in \mathcal{M}$, either the Riemannian gradient of f at \mathcal{M} is sufficiently large, the Riemannian Hessian of $x \in \mathcal{M}$ possesses a sufficiently large eigenvalue, or x is close to a local minimum of f on \mathcal{M} and f is geodesically strongly convex in the neighborhood of this local minimum. Note that the last two cases are mutually exclusive, but that the first case may occur simultaneously with one of the other two.

Remark 2.2. *Other definitions of strict saddle functions have been proposed in the literature, that mainly differ from ours in that the region \mathcal{R}_3 is defined without requiring geodesic strong convexity O’Neill and Wright (2023). One advantage of these other definitions is that they often encompass functions with non-isolated local minima, unlike ours. Indeed, Definition 2.4 does not cover functions with non-isolated local minima, since strong convexity cannot hold at minimizers in this setting. Nevertheless, non-isolated minima often arise due to rotational symmetries in the problem (Wright and Ma, 2022), in which case such rotations define equivalence classes. Reformulating the problem on the quotient manifold induced by these classes leads to a problem where minimizers become isolated, thus fitting our definition. We note that this approach was recently employed for studying Burer-Monteiro formulations of certain semidefinite programs (Luo and Trillos, 2022).*

To end this section, we provide two simple examples of strict saddle functions in the sense of Definition 2.4. Our first example, albeit not being a truly nonconvex function, illustrates the interest of the region \mathcal{R}_3 in the definition.

Example 2.1. *Suppose that $\mathcal{M} = \mathbb{R}^n$, and that $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is a geodesically γ -strongly convex function with global minimum x^* . Then, for any $\alpha > 0$, the function f is $(\alpha, 1, \gamma, \frac{2\alpha}{\gamma})$ -strict saddle. Indeed, with these constants, the region \mathcal{R}_2 is empty, while for any $x \in \mathbb{R}^n$ such that $\|\nabla f(x)\| < \alpha$ (i.e. $x \notin \mathcal{R}_1$), we have by convexity that*

$$\frac{\gamma}{2}\|x - x^*\| \leq f(x) - f(x^*) \leq -\nabla\phi(x)^\top(x - x^*) \leq \|\nabla f(x)\|\|x - x^*\| \leq \alpha\|x - x^*\|,$$

hence $\|x - x^*\| \leq \delta = \frac{2\alpha}{\gamma}$. Since f is γ -strongly convex on \mathbb{R}^n , it is in particular γ -strongly convex on $\{x \mid \|x - x^*\| \leq 2\delta\}$, showing that $x \in \mathcal{R}_3$.

Our second example previously introduced in Sun et al. (2015), illustrates the interest of the region \mathcal{R}_2 in presence of nonconvexity.

Example 2.2. Suppose that \mathcal{M} is the unit sphere in \mathbb{R}^n , denoted by \mathbb{S}^{n-1} and let $f : \mathbb{S}^{n-1} \rightarrow \mathbb{R}$ be defined by $f(x) = x^T A x$, where $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix with eigenvalues $\lambda_1 > \lambda_2 \geq \dots \geq \lambda_{n-1} > \lambda_n$. Then, there exists an absolute constant $c > 0$ such that the function f is $(c(\lambda_{n-1} - \lambda_n)/\lambda_1, c(\lambda_{n-1} - \lambda_n), c(\lambda_{n-1} - \lambda_n), 2c(\lambda_{n-1} - \lambda_n)/\lambda_1)$ -strict saddle function on \mathbb{S}^{n-1} .

To end this section, we provide our key assumption about the problem of interest.

A3. There exist positive constants $(\alpha, \beta, \gamma, \delta)$ such that the function f is $(\alpha, \beta, \gamma, \delta)$ -strict saddle on the manifold \mathcal{M} with \mathcal{R}_3 being a compact set.

Our additional assumption on \mathcal{R}_3 precludes the function from having infinitely many local minima on \mathcal{M} , and is made for simplicity. It is possible to extend our analysis to the case of an unbounded region \mathcal{R}_3 , but this comes at the cost of a significantly more involved reasoning. Note that Assumption 3 holds for both examples above.

3 Riemannian trust-region with exact subproblem solutions

In this section we analyze a classical Riemannian trust-region algorithm applied to a strict saddle function. Our goal is to leverage the strict saddle property to obtain improved complexity bounds compared to those that have previously been derived for general nonconvex functions (Agarwal et al., 2021; Boumal et al., 2019). Our algorithm is fairly standard, yet the analysis borrows from recent results on Newton-type methods from the Euclidean setting (Curtis et al., 2021). In particular, understanding the local convergence properties of such algorithms is crucial to obtaining improved bounds. Note that using the exponential map would greatly simplify the analysis, similarly to the case of Riemannian cubic regularization Agarwal et al. (2021).

Section 3.1 describes our algorithmic framework, along with key assumptions. Standard decrease lemmas are provided in Section 3.2. A local convergence analysis of the algorithm in the region of geodesic strong convexity is provided in Section 3.3. This analysis is instrumental to deriving our global convergence rate results, that are established in Section 3.4.

3.1 Algorithm and assumptions

Our Riemannian trust-region algorithm is described in Algorithm 1. At every iteration, a step s_k is computed by minimizing a given model of the function over the tangent space corresponding to the current iterate x_k . For this section, we assume that the subproblem can be solved exactly using standard approaches (Moré and Sorensen, 1983; Absil et al., 2007) (the inexact case will be addressed in Section 4). Once the step s_k has been computed, the function is evaluated at the point $R_{x_k}(s_k) \in \mathcal{M}$, and the changes in both the true function and the model are compared. Provided those are in sufficient agreement (as measured by the ratio of both quantities), the candidate point $R_{x_k}(s_k)$ is accepted as the new iterate and the trust-region radius may be kept constant (successful iteration) or possibly increased (very successful iteration). Otherwise, the current iterate is kept for the next iteration, and the trust-region radius is decreased (unsuccessful iteration).

In our algorithm, we build the model as a second-order Taylor expansion of the pullback function \hat{f}_{x_k} , namely

$$m_k : T_x \mathcal{M} \rightarrow \mathbb{R} : s \mapsto m_k(s) = f(x_k) + \langle s, g_k \rangle + \frac{1}{2} \langle s, H_k s \rangle, \quad (3.2)$$

Algorithm 1 RTR with exact subproblem minimization

- 1: **Inputs:** Tolerances $(\varepsilon_g, \varepsilon_H)$, initial point $x_0 \in \mathcal{M}$, initial and maximal trust-region radii $0 < \Delta_0 < \bar{\Delta}$, constants $0 < \eta_1 < \eta_2 < 1$ and $0 < \tau_1 < 1 < \tau_2$.
- 2: **for** $k = 1, 2, \dots$ **do**
- 3: Compute s_k as a solution to the trust-region subproblem

$$s_k \in \arg \min_{s \in \mathbb{T}_{x_k} \mathcal{M}} m_k(s) \text{ subject to } \|s\| \leq \Delta_k, \quad (3.1)$$

where m_k is the model defined by (3.2).

- 4:
 - 5: Compute $\rho_k = \frac{f(x_k) - f(R_{x_k}(s_k))}{m_k(0) - m_k(s_k)}$ and set $x_{k+1} = \begin{cases} R_k(x_k) & \text{if } \rho \geq \eta_1 \\ x_k & \text{otherwise.} \end{cases}$
 - 6:
 - 7: Set $\Delta_{k+1} = \begin{cases} \min(\tau_2 \Delta_k, \bar{\Delta}) & \text{if } \rho > \eta_2 \quad [\text{very successful}] \\ \Delta_k & \text{if } \eta_2 \geq \rho \geq \eta_1 \quad [\text{successful}] \\ \tau_1 \Delta_k & \text{otherwise.} \quad [\text{unsuccessful}] \end{cases}$
 - 8: **end for**
-

where $g_k = \nabla \hat{f}_{x_k}(0) = \text{grad}f(x_k)$ and $H_k = \nabla^2 \hat{f}_{x_k}(0)$. Such a model is called second-order accurate. In this paper, we do not consider approximate second-order accurate models, although we believe that our analysis could be extended to such models under suitable conditions (Absil et al., 2008, Eq. (7.36)). Note that when the retraction is second-order (i.e. Assumption 2 holds), it follows that $H_k = \text{Hess}f(x_k)$.

In the rest of Section 3, we will endow Algorithm 1 with complexity results. To this end, we make Lipschitz-type assumptions on the derivatives of the pullback functions Boumal et al. (2019).

A4. *There exists $L_g > 0$ such that for all iterates x_k generated by Algorithm 1, the pullback $f \circ R_{x_k}$ satisfies*

$$f(R_{x_k}(s)) \leq f(x_k) + \langle s, \text{grad}f(x_k) \rangle + \frac{L_g}{2} \|s\|^2. \quad (3.3)$$

for all $s \in \mathbb{T}_x \mathcal{M}$ such that $\|s\| \leq \Delta_k$.

A5. *There exists $L_H > 0$ such that for all iterates x_k generated by Algorithm 1, the pullback $f \circ R_{x_k}$ satisfies*

$$f(R_{x_k}(s)) \leq f(x_k) + \langle s, \text{grad}f(x_k) \rangle + \frac{1}{2} \langle s, \text{Hess}(f \circ R_{x_k})(0)[s] \rangle + \frac{L_H}{6} \|s\|^3. \quad (3.4)$$

for all $s \in \mathbb{T}_x \mathcal{M}$ such that $\|s\| \leq \Delta_k$.

Note that Assumptions 4 and 5 both apply on the trust regions encountered throughout the algorithmic process, rather than on the entire (tangent) space.

We additionally make the following assumption on the Hessian operators considered throughout the algorithm.

A6. *There exists $\kappa_H > 0$ such that for all iterates x_k generated by Algorithm 1, we have*

$$|\langle s, H_k(s) \rangle_{x_k}| \leq \kappa_H \|s\|_{x_k}^2 \quad (3.5)$$

for all $s \in \mathbb{T}_{x_k} \mathcal{M}$.

Note that one could use $\kappa_H = L_g$, however those quantities play different roles in our analysis, and we use distinct notations to better illustrate those roles.

3.2 Preliminary lemmas

In this section, we bound the model decrease in each of the regions defined by the strict saddle property, thanks to standard arguments from the theory of trust-region methods. We also provide a lower bound on the trust-region radius, which is needed to obtain complexity guarantees.

Our first result handles the case of an iterate with large enough gradient (i.e. in \mathcal{R}_1).

Lemma 3.1. *Under A3 and A6, consider the k th iterate of Algorithm 1 and suppose that $x_k \in \mathcal{R}_1$. Then,*

$$m_k(0_k) - m_k(s_k) \geq \frac{1}{2} \min \left(\Delta_k, \frac{\alpha}{\kappa_H} \right) \alpha. \quad (3.6)$$

Proof. Define s_k^C as the Cauchy point associated with the trust-region subproblem (3.1), i.e. $s_k^C = -tg_k$ with $t = \arg \min_{t \geq 0, \|t g_k\| \leq \Delta_k} m_k(-t g_k)$. A straightforward application of Boumal (2023, Lemma 6.15) gives

$$\begin{aligned} m_k(0_k) - m_k(s_k^C) &\geq \frac{1}{2} \min \left(\Delta_k, \frac{\|g_k\|}{\kappa_H} \right) \|g_k\| \\ &\geq \frac{1}{2} \min \left(\Delta_k, \frac{\alpha}{\kappa_H} \right) \alpha. \end{aligned}$$

The desired result then follows from the optimality of s_k , since $m_k(s_k) \leq m_k(s_k^C)$. \square

Our second result considers an iterate at which the Hessian possesses significant negative curvature (i.e. in \mathcal{R}_2).

Lemma 3.2. *Under A3 and A2, consider the k th iterate of Algorithm 1 and suppose that $x_k \in \mathcal{R}_2$. Then,*

$$m_k(0_k) - m_k(s_k) \geq \frac{1}{2} \beta \Delta_k^2. \quad (3.7)$$

Proof. Define $s_k^E = \Delta_k u_k$, where $u_k \in \mathbb{T}_{x_k} \mathcal{M}$ satisfies

$$\|u_k\|_{x_k} = 1, \quad \langle g_k, u_k \rangle_{x_k} \leq 0 \quad \text{and} \quad \langle u_k, H_k u_k \rangle_{x_k} \leq -\beta.$$

Note that such a vector, called an eigenstep for the problem (3.1), exists thanks to $x_k \in \mathcal{R}_2$. Then, the step s_k^E satisfies (Boumal, 2023, Lemma 6.16)

$$m_k(0_k) - m_k(s_k^E) \geq \frac{1}{2} \beta \Delta_k^2.$$

The desired result again follows from the optimality of s_k as $m_k(s_k) \leq m_k(s_k^E)$. \square

Our last decrease lemma is slightly less used in convergence rate analysis of trust-region methods, but proceeds similarly to the previous two lemmas.

Lemma 3.3. *Under A3 and A2, consider the k th iterate of Algorithm 1 and suppose that $x_k \in \mathcal{R}_3$. Then, the step s_k is uniquely defined, and satisfies*

$$m_k(0) - m_k(s_k) \geq \frac{1}{2} \gamma \|s_k\|^2. \quad (3.8)$$

Proof. Since s_k is a solution of the trust-region subproblem (3.1), there exists $\lambda_k \geq 0$ such that the following optimality conditions hold (Absil et al., 2008, Chapter 8):

$$\left(H_k + \lambda_k \text{Id}_{\mathbb{T}_{x_k} \mathcal{M}}\right) s_k = -g_k \quad (3.9)$$

$$\left\langle s, \left(H_k + \lambda_k \text{Id}_{\mathbb{T}_{x_k} \mathcal{M}}\right) [s] \right\rangle \geq 0 \quad \forall s \in \mathbb{T}_{x_k} \mathcal{M} \quad (3.10)$$

$$\|s_k\| \leq \Delta_k \quad (3.11)$$

$$\lambda_k(\Delta_k - \|s_k\|) = 0. \quad (3.12)$$

Moreover, if the inequality in (3.10) is strict for nonzero s , then the solution is unique (Absil et al., 2008, Proposition 7.3.1). To establish a decrease guarantee for s_k , we combine (3.9) and $\lambda_k \geq 0$ to obtain

$$\begin{aligned} m_k(0) - m_k(s_k) &= -\langle s_k, \text{grad}f(x_k) \rangle - \frac{1}{2} \langle s_k, H_k s_k \rangle \\ &= \langle s_k, (H_k + \lambda_k \text{Id}) s_k \rangle - \frac{1}{2} \langle s_k, H_k s_k \rangle \\ &= \frac{1}{2} \langle s_k, H_k s_k \rangle + \lambda_k \|s_k\|^2 \\ &\geq \frac{1}{2} \langle s_k, H_k s_k \rangle. \end{aligned}$$

Using now that $H_k = \text{Hess}f(x_k)$ and $x_k \in \mathcal{R}_3$, we have that $\langle s, H_k[s] \rangle \geq \gamma \|s\|^2$ for any $s \in \mathbb{T}_{x_k} \mathcal{M}$. This implies that $\langle s, (H_k + \lambda_k \text{Id}_{\mathbb{T}_{x_k} \mathcal{M}})[s] \rangle > 0$ for nonzero s , hence s_k is uniquely defined. In addition, we obtain

$$m_k(0) - m_k(s_k) \geq \frac{1}{2} \langle s_k, H_k s_k \rangle \geq \frac{\gamma}{2} \|s_k\|^2,$$

showing the desired result. \square

The three lemmas above together with the Lipschitz-type assumptions on the pullback function yield a lower bound on the trust-region radius, as shown by Lemma 3.4.

Lemma 3.4. *Let A2, A3, A4, A5 and A6 hold. Then, for any iteration index k , the trust-region radius Δ_k satisfies*

$$\Delta_k \geq \Delta_{\min} := c_{\Delta} \min \{\Delta_0, \alpha, \beta, \gamma\}, \quad (3.13)$$

$$\text{where } c_{\Delta} = \min \left[1, \frac{1 - \eta_1}{L_g + \kappa_H}, 3\tau_1 \frac{(1 - \eta_1)}{L_H} \right].$$

Proof. We begin by showing that if the trust-region radius drops below a certain threshold, then the iteration is necessarily successful. To that aim, we investigate the quantity

$$1 - \rho_k = 1 - \frac{f(x_k) - f(R_{x_k}(s_k))}{m_k(0) - m_k(s_k)} = \frac{f(R_{x_k}(s_k)) - m_k(s_k)}{m_k(0) - m_k(s_k)} \quad (3.14)$$

for the three possible cases identified by the strict saddle property.

Suppose first that $x_k \in \mathcal{R}_1$. Using (3.3), the numerator of (3.14) can be upper-bounded as follows:

$$\begin{aligned} f(R_{x_k}(s_k)) - m_k(s_k) &= f(R_{x_k}(s_k)) - f(x_k) - \langle g_k, s_k \rangle - \frac{1}{2} \langle s_k, H_k s_k \rangle \\ &\leq \frac{L_g}{2} \|s_k\|^2 + \frac{1}{2} \|H_k\| \|s_k\|^2 \\ &\leq \frac{L_g + \kappa_H}{2} \Delta_k^2. \end{aligned}$$

Meanwhile, the denominator satisfies (3.6) per Lemma 3.1. It follows that

$$1 - \rho_k \leq \frac{(L_g + \kappa_H)\Delta_k^2}{\min\left(\Delta_k, \frac{\alpha}{\kappa_H}\right)\alpha}.$$

As a result, if

$$\Delta_k \leq \min\left(\frac{\alpha}{\kappa_H}, \frac{(1 - \eta_1)\alpha}{L_g + \kappa_H}\right) = \frac{1 - \eta_1}{L_g + \kappa_H}\alpha,$$

then $1 - \rho_k \leq 1 - \eta_1$ and iteration k is successful.

Suppose now that $x_k \in \mathcal{R}_2$. In that case, A2 together with (3.4) give the following bound on the numerator of (3.14):

$$f(R_{x_k}(s_k)) - m_k(s_k) = f(R_{x_k}(s_k)) - f(x_k) - \langle g_k, s_k \rangle - \frac{1}{2} \langle s_k, H_k s_k \rangle \leq \frac{L_H}{6} \|s_k\|^3. \quad (3.15)$$

while the numerator satisfies (3.7). Combining both results then gives

$$1 - \rho_k \leq \frac{1}{6} \frac{L_H \|s_k\|^3}{1/2 \Delta_k^2 \beta} \leq \frac{1}{3} \frac{L_H \Delta_k^3}{\Delta_k^2 \beta} = \frac{1}{3} \frac{L_H}{\beta} \Delta_k.$$

Thus, if $\Delta_k \leq 3(1 - \eta_1)\beta/L_H$, we have $\rho_k \geq \eta_1$ and iteration k is successful.

Finally, suppose that $x_k \in \mathcal{R}_3$. Using the upper bound (3.15) together with the decrease (3.8), we obtain that

$$1 - \rho_k \leq \frac{L_H \|s_k\|^3}{1/2 \|s_k\|^2 \gamma} = \frac{1}{3} \frac{L_H}{\gamma} \|s_k\| \leq \frac{1}{3} \frac{L_H}{\gamma} \Delta_k.$$

As a result, if $\Delta_k \leq 3(1 - \eta_1)\beta/L_H$, then $\rho_k \geq \eta_1$ and iteration k is successful.

Overall, we have shown that the iteration k is successful as long as

$$\Delta_k \leq \min\left[\frac{(1 - \eta_1)}{L_g + \kappa_H}\alpha, \frac{3(1 - \eta_1)}{L_H}\beta, \frac{3(1 - \eta_1)}{L_H}\gamma\right],$$

in which case $\Delta_{k+1} \geq \Delta_k$. It follows from the updating rule on Δ_k that the trust-region radius is lower bounded for any $k \geq 1$ by

$$\Delta_k \geq \tau_1 \min\left[\frac{(1 - \eta_1)}{L_g + \kappa_H}\alpha, \frac{3(1 - \eta_1)}{L_H}\beta, \frac{3(1 - \eta_1)}{L_H}\gamma\right] \geq c_\Delta \min\{\alpha, \beta, \gamma\}.$$

Accounting for $k = 0$, we arrive at the desired result. \square

Lemma 3.4 will be combined with the results of Lemmas 3.1 and 3.2 to provide decrease guarantees independent on k when $\|s_k\| = \Delta_k$. Deriving such guarantees from Lemma 3.3 is significantly more involved, and is the topic of the next section.

3.3 Region of geodesic strong convexity and local convergence

In this section, we analyze the behavior of the iterates produced by Algorithm 1 that belong to \mathcal{R}_3 , thanks to the properties of the global subproblem solution. Since that solution is a Newton-type step (recall (3.9)), our approach mimics the study of Newton's method applied to strongly convex functions Boyd and Vandenberghe (2004). Indeed, we can either provide a lower bound on the decrease achieved by the steps, or show that a local convergence phase begins, during which the iterates converge quadratically towards a local minimum.

Before stating those results, we state several consequences of Assumption A3 that are helpful in analyzing Newton-type steps, starting with a Lipschitz-type inequality on the gradient of the pullback function.

Lemma 3.5. Under A3, there exists $\hat{L}_H > 0$ such that for all iterates $x_k \in \mathcal{R}_3$ produced by Algorithm 1, we have

$$\left\| \nabla \hat{f}_k(s_k) - \nabla \hat{f}_k(0) - \nabla^2 \hat{f}_k(0)[s_k] \right\| \leq \frac{\hat{L}_H}{2} \|s_k\|^2. \quad (3.16)$$

Proof. The constant \hat{L}_H exists by continuity of the derivatives over a compact set and bounded steps (Boumal, 2023, Lemma 10.57). The compactness of \mathcal{R}_3 follows from A3, while the boundedness of the steps s_k follows from $\|s_k\| \leq \Delta_k \leq \bar{\Delta}$. \square

Note that Lemma 3.5 is stated on the points of interest produced by the algorithm, similarly to the Lipschitz-type assumptions A4 and A5.

The next result describes a non-singularity condition for the pullback gradient, which is critical for small-norm steps. A similar property was used in the context of Riemannian cubic regularization, that also employs Newton-type steps (Agarwal et al., 2021).

Lemma 3.6. Under A3, for any $\kappa_R > 1$ there exists $\nu_R > 0$ such that for any iterate $x_k \in \mathcal{R}_3$ produced by Algorithm 1 such that $\|s_k\| \leq \nu_R$, we have

$$\|\text{grad}f(R_{x_k}(s))\|_{x_{k+1}} \leq \kappa_R \left\| \nabla \hat{f}(s) \right\|_{x_k}. \quad (3.17)$$

Proof. We apply (Agarwal et al., 2021, Theorem 7) to \mathcal{R}_3 as a non-empty compact subset of \mathcal{M} . This result ensures that for any $c_R > 1$, there exists a constant $\nu_R > 0$ such that, at each $x_k \in \mathcal{R}_3$,

$$\|s_k\| \leq \nu_R \quad \Leftrightarrow \quad \sigma_{\min}(\text{DR}_{x_k}(s_k)) \geq \frac{1}{\kappa_R},$$

where DR_{x_k} denotes the differential of R_{x_k} . The desired conclusion follows by combining this result with (Agarwal et al., 2021, Equation 22), that gives (when applied with $s = s_k$)

$$\left\| \nabla \hat{f}_{x_k}(s_k) \right\| \geq \sigma_{\min}(\text{DR}_{x_k}(s_k)) \|\text{grad}f(R_{x_k}(s))\|. \quad \square$$

Combining the previous two lemmas, we obtain the following result on the change in gradient norm for

Lemma 3.7. Suppose that A2 and A3 hold. Let $x_k \in \mathcal{R}_3$ be an iterate produced by Algorithm 1 such that s_k is a Newton step, i.e. $H_k s_k = -g_k$, and $\|s_k\| \leq \nu_R$ where ν_R is the constant from Lemma 3.6. Then, if the iteration is successful, we have

$$\|\text{grad}f(x_{k+1})\|_{x_{k+1}} \leq c_R \frac{L}{2} \|s_k\|_{x_k}^2 \quad (3.18)$$

where \hat{L}_H comes from Lemma 3.5.

Proof. Let $x_{k+1} = R_{x_k}(s_k)$. Using successively Lemma 3.6, Lemma 3.5, and A2, we obtain

$$\begin{aligned} \|\text{grad}f(x_{k+1})\|_{x_{k+1}} &\leq \kappa_R \left\| \nabla \hat{f}_{x_k}(s_k) \right\|_{x_k} \\ &= \kappa_R \left\| \nabla \hat{f}_{x_k}(s_k) - \text{grad}f(x_k) + \text{grad}f(x_k) \right\|_{x_k} \\ &= \kappa_R \left\| \nabla \hat{f}_{x_k}(s_k) - \text{grad}f(x_k) - H_k s_k \right\|_{x_k} \\ &= \kappa_R \left\| \nabla \hat{f}_{x_k}(s_k) - \nabla \hat{f}_{x_k}(0) - \nabla^2 \hat{f}_{x_k}(0)[s_k] \right\|_{x_k} \\ &\leq \kappa_R \frac{\hat{L}_H}{2} \|s_k\|_{x_k}^2 \end{aligned}$$

\square

Remark 3.1. *Our results apply for a general retraction R , but using the exponential map Exp greatly. In particular, thanks to (Boumal, 2023, Corollary 10.56), the result of Lemma 3.5 can be replaced by*

$$\|P_s^{-1}\text{grad}f(\text{Exp}_x(s)) - \text{grad}f(x) - \text{Hess}f(x)[s]\| \leq \frac{L_H}{2} \|s\|^2 \quad (3.19)$$

where P_s^{-1} is a parallel transport map that maps $\text{grad}f(\text{Exp}_x(s)) \in T_{\text{Exp}_x(s)}\mathcal{M}$ to $T_x\mathcal{M}$, and L_H is the Lipschitz constant of $\text{Hess}f$ from (3.4). As a result, the proof of Lemma 3.7 is also simplified, and no longer requires the decomposition in long and short steps induced by Lemma 3.6.

We are now equipped to prove a decrease guarantee on Newton steps.

Lemma 3.8. *Under the assumptions of Lemma 3.7, we have*

$$m_k(0) - m_k(s_k) \geq \frac{\gamma}{\hat{L}_H \kappa_R} \|\text{grad}f(x_{k+1})\|_{x_{k+1}}.$$

Proof. By combining (3.18) with (3.8), we obtain

$$\begin{aligned} m_k(0) - m_k(s_k) &\geq \frac{\gamma}{2} \|s_k\|_{x_k}^2 \\ &\geq \frac{\gamma}{\hat{L}_H \kappa_R} \|\text{grad}f(x_{k+1})\|_{x_{k+1}}, \end{aligned} \quad \square$$

which is the desired result.

We now turn to local convergence results. Our goal is to show that Newton steps are eventually accepted by the algorithm, and that they produce iterates with decreasing gradient norm. To begin with, we give a bound on the norm of such steps.

Lemma 3.9. *Suppose that Algorithm 1 produces an iterate $x_k \in \mathcal{R}_3$. Then,*

$$\|s_k\| \leq \frac{\|g_k\|}{\gamma}. \quad (3.20)$$

Proof. The result trivially holds if $\|s_k\| = 0$. Otherwise, using the definition of \mathcal{R}_3 together with $\lambda_k \geq 0$ and (3.9), we get

$$\gamma \|s_k\|^2 \leq \langle s_k, H_k s_k \rangle \leq \langle s_k, (H_k + \lambda \text{Id}) s_k \rangle = -\langle s_k, g_k \rangle \leq \|s_k\| \|g_k\|,$$

and we obtain (3.20) by dividing by $\|s_k\|$. □

Lemma 3.9 is a key ingredient towards showing an upper bound on the gradient that leads to a successful iteration for an iterate in \mathcal{R}_3 .

Proposition 3.10. *Under A5 and A2, suppose that Algorithm 1 generates $x_k \in \mathcal{R}_3$ such that*

$$\|\text{grad}f(x_k)\|_{x_k} < \frac{3(1 - \eta_1)\gamma^2}{L_H}. \quad (3.21)$$

Then, the corresponding iteration is successful.

Proof. First, we note that the condition for a successful step $\rho_k \geq \eta_1$ is equivalent to

$$r_k = f(R_k(s_k)) - m_k(s_k) + (1 - \eta_1)(m_k(s_k) - f(x_k)) \leq 0.$$

The proof then consists in finding an upper bound for r_k that is negative when (3.21) holds. From Lemma 3.3, we have that

$$(1 - \eta_1)(m_k(s_k) - f(x_k)) \leq (1 - \eta_1)\left(-\frac{\gamma}{2} \|s_k\|^2\right).$$

Combining this property with (3.15) and (3.20) gives

$$\begin{aligned} r_k &\leq \frac{L_H}{6} \|s_k\|^3 - \frac{\gamma}{2}(1 - \eta_1) \|s_k\|^2 = \|s_k\|^2 \left(\frac{L_H}{6} \|s_k\| - \frac{\gamma}{2}(1 - \eta_1) \right) \\ &\leq \|s_k\|^2 \left(\frac{L_H}{6} \frac{\|g_k\|}{\gamma} - \frac{\gamma}{2}(1 - \eta_1) \right). \end{aligned}$$

The condition (3.21) is equivalent to the latter bound being negative, from which we conclude that $r_k \leq 0$, hence the iteration is successful. \square

We now show that if a Newton step leads to a successful iteration with a small enough gradient, the gradient norm decreases.

Proposition 3.11. *Suppose that A2 and A3 hold. Let $x_k \in \mathcal{R}_3$ be an iterate produced by Algorithm 1 such that s_k is a Newton step, i.e. $H_k s_k = -g_k$, $\|s_k\| \leq \nu_R$ where ν_R is the constant from Lemma 3.6, and*

$$\|\text{grad}f(x_k)\|_{x_k} < \frac{2\gamma^2}{\kappa_R \hat{L}_H}. \quad (3.22)$$

Then, if the iteration is successful, the next iterate satisfies $\|\text{grad}f(x_{k+1})\|_{x_{k+1}} < \|\text{grad}f(x_k)\|_{x_k}$.

Proof. Combining (3.18), (3.20) and (3.22), we immediately obtain that

$$\|g_{k+1}\| \leq \frac{\kappa_R \hat{L}_H}{2} \|s_k\|^2 \leq \frac{\kappa_R \hat{L}_H}{2} \frac{\|g_k\|^2}{\gamma^2} \leq \frac{\kappa_R \hat{L}_H}{2\gamma^2} \|g_k\| \cdot \|g_k\| < \|g_k\|.$$

\square

We have shown that the gradient norm decreases for iterates in \mathcal{R}_3 corresponding to successful iterations with Newton steps. In order to derive a local convergence result, we now establish that the iterates remain in \mathcal{R}_3 if the gradient is small enough.

Proposition 3.12. *Suppose that A1, A2 and A3 hold. Let $x_k \in \mathcal{R}_3$ be an iterate produced by Algorithm 1 such that s_k is a Newton step, and $\|s_k\| \leq \nu_R$ where ν_R is the constant from Lemma 3.6, and that*

$$\|\text{grad}f(x_k)\| \leq \min \left(\nu_S \gamma, \frac{\gamma \delta}{\kappa_S}, \frac{\delta \gamma}{2}, \frac{2\gamma^2}{\kappa_R \hat{L}_H} \right). \quad (3.23)$$

Let $x^ \in \mathcal{M}$ be a local minimum of problem (P) such that $\text{dist}(x_k, x^*) \leq \delta$ and f is geodesically γ -strongly convex on $\{y \mid \text{dist}(y, x^*) \leq 2\delta\}$. If the iteration is successful, then $\text{dist}(x_{k+1}, x^*) < \delta$.*

Proof. We first show that $\text{dist}(x_{k+1}, x^*) \leq 2\delta$. Using (3.20), we have that

$$\|s_k\| \leq \frac{\|g_k\|}{\gamma} < \frac{\nu_S \gamma}{\gamma} = \nu_S.$$

From A1, it then follows that

$$\text{dist}(x_{k+1}, x_k) \leq \kappa_S \|s_k\| \leq \kappa_S \frac{\|g_k\|}{\gamma} \leq \kappa_S \frac{\gamma \delta}{\kappa_S \gamma} = \delta,$$

where the last inequality holds thanks to (3.23). As a result,

$$\text{dist}(x_{k+1}, x^*) \leq \text{dist}(x_{k+1}, x_k) + \text{dist}(x_k, x^*) \leq 2\delta.$$

By definition of x^* , we then know that f is γ -strongly geodesically convex over a subset of \mathcal{M} that includes x^* and x_{k+1} . Consider a geodesic segment contained in that subset of the form $c: [0, 1] \rightarrow \mathcal{M}$ with $c(0) = x_{k+1}$ and $c(1) = x^*$ such that $\text{dist}(x_{k+1}, x^*) \leq \ell_c = \|c'(0)\|_{c(0)}$, where ℓ_c is the length of the geodesic path. Using that f is geodesically γ -strongly convex on the segment c , owing to (Boumal, 2023, Theorem 11.21), we have

$$f(x^*) \geq f(x_{k+1}) + \langle \text{grad}f(x_{k+1}), c'(0) \rangle_{x_{k+1}} + \frac{\gamma}{2} L(c)^2.$$

Using then $f(x^*) \leq f(x_{k+1})$ gives

$$\begin{aligned} \frac{\gamma}{2} L(c)^2 &\leq \langle \text{grad}f(x_{k+1}), -c'(0) \rangle_{x_{k+1}} \\ &\leq \|\text{grad}f(x_{k+1})\|_{x_{k+1}} \|c'(0)\|_{x_{k+1}} \\ &= \|\text{grad}f(x_{k+1})\|_{x_{k+1}} L(c). \end{aligned}$$

Therefore, we have $L(c) \leq \frac{2}{\gamma} \|\text{grad}f(x_{k+1})\|_{x_{k+1}}$. To conclude, recall that we have $\|s_k\| \leq \nu_R$ by assumption, thus Proposition 3.11 applies., and we obtain

$$\text{dist}(x_{k+1}, x^*) \leq L(c) \leq \frac{2}{\gamma} \|\text{grad}f(x_{k+1})\|_{x_{k+1}} < \frac{2}{\gamma} \|\text{grad}f(x_k)\|_{x_k} < \frac{2}{\gamma} \frac{\delta \gamma}{2} = \delta,$$

proving the desired result. \square

We can now characterize local quadratic convergence properties of Algorithm 1. Our main result is the following proposition.

Proposition 3.13. *Suppose that A1, A2 and A3 hold. Let $x_k \in \mathcal{R}_3$ be an iterate produced by Algorithm 1 such that s_k is a Newton step, and $\|s_k\| \leq \nu_R$ where ν_R is the constant from Lemma 3.6. Suppose further that the k th iteration is successful. Then,*

$$\frac{\kappa_R \hat{L}_H}{2\gamma^2} \|\text{grad}f(x_{k+1})\|_{x_{k+1}} \leq \left(\frac{\kappa_R \hat{L}_H}{2\gamma^2} \|\text{grad}f(x_k)\|_{x_k} \right)^2.$$

Proof. Using Lemma 3.7 and Lemma 3.9 gives

$$\|\text{grad}f(x_{k+1})\|_{x_{k+1}} \leq \frac{\kappa_R \hat{L}_H}{2} \|s_k\|_{x_k}^2 \leq \frac{\kappa_R \hat{L}_H}{2\gamma^2} \|\text{grad}f(x_k)\|_{x_k}^2,$$

and multiplying both sides by $\frac{\kappa_R \hat{L}_H}{2\gamma^2}$ yields the desired conclusion. \square

The result of Proposition 3.13 guarantees quadratic convergence provided the Newton steps are accepted. Our final result summarizes that of this section, and shows that local quadratic convergence is triggered once the algorithm generates a point in \mathcal{R}_3 with a small enough gradient.

Finally, we provide conditions under which we enter a local quadratic convergence regime in Proposition 3.14.

Proposition 3.14 (Local convergence of trust-region). *Suppose that A1, A2 and A3 hold. Let $x_k \in \mathcal{R}_3$ be an iterate produced by Algorithm 1, and let $x^* \in \mathcal{M}$ be a local minimum of (P) such that $\text{dist}(x_k, x^*) \leq \delta$ and f is geodesically γ -strongly convex on $\{y | \text{dist}(y, x^*) \leq 2\delta\}$. Finally, suppose that $\|s_k\| < \nu_R$ and*

$$\|\text{grad}f(x_k)\| < \min\{c_Q \min\{\gamma, \gamma\delta, \gamma^2\}, \gamma\Delta_k\} \quad (3.24)$$

where $c_Q = \min\left[\frac{3(1-\eta_1)}{L_H}, \nu_R, \frac{1}{\kappa_S}, \frac{1}{2}, \frac{2}{\kappa_R \hat{L}_H}\right]$. Then, all subsequent iterations are successful, and correspond to iterates in \mathcal{R}_3 as well as Newton steps. Moreover, the sequence of gradient norms $(\|\text{grad}f(x_k)\|)_k$ converges quadratically to zero.

Proof. First note that (3.24) implies in particular $\|\text{grad}f(x_k)\| < \gamma\Delta_k$. Applying Lemma 3.9, we find that $\|s_k\| \leq \|\text{grad}f(x_k)\|/\gamma < \Delta_k$, and thus s_k must be the Newton step. In addition, the condition (3.24) also implies (3.21), thus the k th iteration is successful per Proposition 3.10. Similarly, it implies (3.22), which together with $\|s_k\| \leq \nu_R$ yields $\|g_{k+1}\| < \|g_k\|$ (Proposition 3.11) and $\|x_{k+1} - x^*\| < \delta$ (Proposition 3.12).

Since $\|g_{k+1}\| < \|g_k\| < \gamma\Delta_k \leq \gamma\Delta_{k+1}$, the same reasoning applies at iteration $k+1$, and at every subsequent iteration by induction, proving the first part of the result. Finally, quadratic convergence follows from (3.24) implying $\|\text{grad}f(x_k)\| < \gamma^2/(c_R L)$ and repeated application of Proposition 3.13. Indeed, for any index $l \geq k$, we have

$$\frac{\hat{L}_H \kappa_R}{2\gamma^2} \|\text{grad}f(x_l)\| \leq \left(\frac{\hat{L}_H \kappa_R}{2\gamma^2} \|\text{grad}f(x_k)\| \right)^{2^{l-k}} \leq \left(\frac{1}{2} \right)^{2^{l-k}},$$

which characterizes quadratic convergence. \square

We again emphasize that the local quadratic convergence property is instrumental to deriving *global* convergence rates in our setting, since the local convergence rate is a direct consequence of the strict saddle property. Deriving global rates of convergence (or, equivalently, complexity results) is the subject of the next section.

3.4 Complexity bounds

In this section, we combine the results from Sections 3.2 and 3.3 to obtain complexity bounds. More precisely, we seek a bound on the number of iterations of Algorithm 1 required to reach an iterate $x_K \in \mathcal{M}$ satisfying

$$\|\text{grad}f(x_K)\| \leq \varepsilon_g \quad \text{and} \quad \langle s, \text{Hess}f(x_K) s \rangle \geq -\varepsilon_H \|s\|^2 \quad \forall s \in \mathbb{T}_{x_K} \mathcal{M}. \quad (3.25)$$

Following Section 3.3, we can bound the number of iterations in a local phase necessary to satisfy (3.25). The result below is a direct corollary of Proposition 3.14.

Theorem 3.15. *Let the assumptions of Proposition 3.14 hold for $x_k \in \mathcal{R}_3$ generated by Algorithm 1. Then, the algorithm returns an iterate satisfying (3.25) in at most*

$$\log_2 \log_2 \left(\frac{2\gamma^2}{\kappa_R \hat{L}_H \varepsilon_g} \right) \quad (3.26)$$

iterations following iteration k .

Proof. As in the proof of Proposition 3.14, we can show that for any $l \geq k$, we have

$$\frac{\hat{L}_H \kappa_R}{2\gamma^2} \|\text{grad}f(x_l)\| \leq \left(\frac{\hat{L}_H \kappa_R}{2\gamma^2} \|\text{grad}f(x_k)\| \right)^{2^{l-k}} \leq \left(\frac{1}{2} \right)^{2^{l-k}}.$$

It follows that if $\|\text{grad}f(x_l)\| \geq \varepsilon_g$, then it must be that

$$l - k \leq \log_2 \log_2 \left(\frac{2\gamma^2}{\hat{L}_H \kappa_R \varepsilon_g} \right).$$

□

We are now in a position to establish our main complexity result, under the following assumption.

A7 (lower bound). *There exists $f^* > -\infty$ such that $f(x) \geq f^*$ for all $x \in \mathcal{M}$.*

The following theorem gives an upper bound on the number of successful steps for Algorithm 1.

Theorem 3.16 (Number of successful iterations of Algorithm 1). *Suppose that A1–A7 hold. Algorithm 1 produces an iterate satisfying (3.25) in at most*

$$\frac{C}{\min \{ \underline{\alpha}^2 \underline{\beta}, \underline{\alpha} \underline{\gamma}^2, \underline{\alpha}^2 \underline{\gamma}, \underline{\beta} \underline{\gamma}^2, \underline{\beta}^2 \underline{\gamma}, \underline{\beta}^3, \underline{\gamma}^3, \underline{\gamma}^2 \underline{\delta} \}} + 1 + \log_2 \log_2 \left(\frac{2\gamma^2}{\kappa_R \hat{L}_H \varepsilon_g} \right)$$

successful iterations, where the constant $C > 0$ depends on $\kappa_H, c_\Delta, \Delta_0, \nu_R, \kappa_R, \hat{L}_H, \eta_1, c_Q$, and for any $\theta \in \{\alpha, \beta, \gamma, \delta\}$, we define $\underline{\theta} = \min(1, \theta)$.

Proof. Let $K \in \mathbb{N}$ such that Algorithm 1 has not produced an iterate satisfying (3.25) by iteration K . Let $\mathcal{S} = \{k \leq K : \rho_k \geq \eta_1\}$ denote the set of indices corresponding to successful (and very successful) iterations. For convenience of exposition, we partition the set of iterations as follows:

$$\begin{aligned} \mathcal{S}_1 &= \{k \in \mathcal{S} : x_k \in \mathcal{R}_1\} \\ \mathcal{S}_2 &= \{k \in \mathcal{S} : x_k \in \mathcal{R}_2 \setminus \mathcal{R}_1\} \\ \mathcal{S}_3 &= \{k \in \mathcal{S} : x_k \in \mathcal{R}_3 \setminus \mathcal{R}_1\}. \end{aligned}$$

We now bound the decrease in function value for all three sets.

Consider first an index $k \in \mathcal{S}_1$. Using Lemma 3.1 and Lemma 3.4, we obtain

$$f(x_k) - f(x_{k+1}) \geq \eta_1 (m_k(0) - m_k(s_k)) \geq \frac{\eta_1}{2} \min \left(\frac{\alpha}{\kappa_H}, \Delta_k \right) \alpha \geq \frac{\eta_1}{2} \min \left(\frac{\alpha}{\kappa_H}, \Delta_{\min} \right) \alpha. \quad (3.27)$$

Consider now $k \in \mathcal{S}_2$. Combining Lemma 3.2 with Lemma 3.4, we arrive at

$$f(x_k) - f(x_{k+1}) \geq \eta_1 (m_k(0) - m_k(s_k)) \geq \frac{\eta_1}{2} \Delta_k^2 \beta \geq \frac{\eta_1}{2} \Delta_{\min}^2 \beta. \quad (3.28)$$

Finally, consider $k \in \mathcal{S}_3$, and partition further this index into $\mathcal{S}_3^l \cup \mathcal{S}_3^{s,b} \cup \mathcal{S}_3^{s,i}$, where

$$\begin{aligned} \mathcal{S}_3^l &= \{k \in \mathcal{S}_3 : \|s_k\| > \nu_R\} \\ \mathcal{S}_3^{s,b} &= \{k \in \mathcal{S}_3 : \|s_k\| \leq \nu_R, \|s_k\| = \Delta_k\} \\ \mathcal{S}_3^{s,i} &= \{k \in \mathcal{S}_3 : \|s_k\| \leq \nu_R, \|s_k\| < \Delta_k\} \end{aligned}$$

ans ν_R is the quantity defined in Lemma 3.7.

If $k \in \mathcal{S}_3^l$, Lemma 3.3 yields

$$f(x_{k+1}) - f(x_k) \geq \frac{\eta_1}{2} \|s_k\|^2 \gamma \geq \frac{\eta_1}{2} \nu_R^2 \gamma. \quad (3.29)$$

If $k \in \mathcal{S}_3^{s,b}$, we use Lemma 3.3 together with Lemma 3.4 to obtain

$$f(x_{k+1}) - f(x_k) \geq \frac{\eta_1}{2} \|s_k\|^2 \gamma = \frac{\eta_1}{2} \Delta_k^2 \gamma \geq \frac{\eta_1}{2} \Delta_{\min}^2 \gamma. \quad (3.30)$$

Finally, if $k \in \mathcal{S}_3^{s,i}$, we partition further $\mathcal{S}_3^{s,i}$ into $\mathcal{S}_3^{s,i,s} \cup \mathcal{S}_3^{s,i,l}$, where

$$\begin{aligned} \mathcal{S}_3^{s,i,l} &= \{k \in \mathcal{S}_3^{s,i} : \|g_{k+1}\| \geq \min\{c_Q \min\{\gamma, \gamma\delta, \gamma^2\}, \gamma\Delta_k\}\} \\ \mathcal{S}_3^{s,i,s} &= \mathcal{S}_3^{s,i} \setminus \mathcal{S}_3^{s,i,l}. \end{aligned}$$

If $k \in \mathcal{S}_3^{s,i,l}$, Proposition 3.8 implies

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1 \gamma \frac{\|g_{k+1}\|}{\hat{L}_H \kappa_R} \geq \frac{\eta_1}{\hat{L}_H \kappa_R} \min\{c_Q \min\{\gamma^2, \gamma^2\delta, \gamma^3\}, \gamma^2 \Delta_k\} \\ &\geq \frac{\eta_1}{\hat{L}_H \kappa_R} \min\{c_Q \min\{\gamma^2, \gamma^2\delta, \gamma^3\}, \gamma^2 \Delta_{\min}\}. \end{aligned} \quad (3.31)$$

Finally, if $k \in \mathcal{S}_3^{s,i,s}$, either $x_{k+1} \in \mathcal{S}_3$ and the local quadratic phase begins according to Proposition 3.14. Otherwise, we must have $x_{k+1} \in \mathcal{S}_1 \cup \mathcal{S}_2$, and as a result we have

$$\left| \mathcal{S}_3^{s,i,s} \right| \leq |\mathcal{S}_1| + |\mathcal{S}_2| + 1 + \log_2 \log_2 \left(\frac{2\gamma^2}{\kappa_R \hat{L}_H \varepsilon_g} \right). \quad (3.32)$$

It thus suffices to bound the cardinality of the other iteration indices to bound that of $\mathcal{S}_3^{s,i,s}$.

Thanks to A7, we have

$$\begin{aligned} f(x_0) - f^* &\geq f(x_0) - f(x_K) \\ &\geq \sum_{\substack{k \in \mathcal{S} \\ k \leq K}} f(x_k) - f(x_{k+1}) \\ &\geq \sum_{\substack{k \in \mathcal{S}_1 \\ k \leq K}} f(x_k) - f(x_{k+1}) + \sum_{\substack{k \in \mathcal{S}_2 \\ k \leq K}} f(x_k) - f(x_{k+1}) + \sum_{\substack{k \in \mathcal{S}_3^l \\ k \leq K}} f(x_k) - f(x_{k+1}) \\ &\quad + \sum_{\substack{k \in \mathcal{S}_3^{s,b} \\ k \leq K}} f(x_k) - f(x_{k+1}) + \sum_{\substack{k \in \mathcal{S}_3^{s,i,l} \\ k \leq K}} f(x_k) - f(x_{k+1}). \end{aligned}$$

Putting (3.27), (3.28), (3.29), (3.30) and (3.31) together, we obtain

$$\begin{aligned} f(x_0) - f^* &\geq |\mathcal{S}_1| \frac{\eta_1}{2} \min\left(\frac{\alpha}{\kappa_H}, \Delta_{\min}\right) \alpha + |\mathcal{S}_2| \frac{\eta_1}{2} \Delta_{\min}^2 \beta + |\mathcal{S}_3^l| \frac{\eta_1}{2} \nu_R^2 \gamma \\ &\quad + |\mathcal{S}_3^{s,b}| \frac{\eta_1}{2} \Delta_{\min}^2 \gamma + |\mathcal{S}_3^{s,i,l}| \frac{\eta_1}{\hat{L}_H \kappa_R} \min\{c_Q \min\{\gamma^2, \gamma^2\delta, \gamma^3\}, \gamma^2 \Delta_{\min}\}. \end{aligned}$$

Since all quantities on the right-hand side are nonnegative, we can bound each cardinality independently as follows:

$$\begin{aligned}
|\mathcal{S}_1| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \max\{\kappa_H \alpha^{-2}, \Delta_{\min}^{-1} \alpha^{-1}\} \\
|\mathcal{S}_2| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \Delta_{\min}^{-2} \beta^{-1} \\
|\mathcal{S}_3^l| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \nu_R^{-2} \gamma^{-1} \\
|\mathcal{S}_3^{s,b}| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \Delta_{\min}^{-2} \gamma^{-1} \\
|\mathcal{S}_3^{s,i,l}| &\leq \frac{\kappa_R \hat{L}_H (f(x_0) - f^*)}{\eta_1} \max\{c_Q^{-1} \max\{\gamma^{-2}, \gamma^{-2} \delta^{-1}, \gamma^{-3}\}, \gamma^{-2} \Delta_{\min}^{-1}\}.
\end{aligned}$$

Using that $\Delta_{\min} = c_\Delta \min\{\Delta_0, \alpha, \beta, \gamma\} \geq c_\Delta \min\{\Delta_0, 1\} \min\{\alpha, \beta, \gamma\}$ and bounding any $\theta \in \{\alpha, \beta, \gamma, \delta\}$ by $\underline{\theta}$ yields the following upper bounds

$$\begin{aligned}
|\mathcal{S}_1| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \max\{\kappa_H, c_\Delta^{-1} \Delta_0^{-1}, c_\Delta^{-1}\} \max\{\underline{\alpha}^{-2}, \underline{\alpha}^{-1} \underline{\beta}^{-1}, \underline{\alpha}^{-1} \underline{\gamma}^{-1}\} \\
|\mathcal{S}_2| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \max\{c_\Delta^{-2} \Delta_0^{-2}, c_\Delta^{-2}\} \max\{\underline{\alpha}^{-2} \underline{\beta}^{-1}, \underline{\beta}^{-3}, \underline{\beta}^{-1} \underline{\gamma}^{-2}\} \\
|\mathcal{S}_3^l| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \nu_R^{-2} \underline{\gamma}^{-1} \\
|\mathcal{S}_3^{s,b}| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \max\{c_\Delta^{-2} \Delta_0^{-2}, c_\Delta^{-2}\} \max\{\underline{\alpha}^{-2} \underline{\gamma}^{-1}, \underline{\beta}^{-2} \underline{\gamma}^{-1}, \underline{\gamma}^{-3}\} \\
|\mathcal{S}_3^{s,i,l}| &\leq \frac{\kappa_R \hat{L}_H (f(x_0) - f^*)}{\eta_1} \max\{c_Q^{-1}, c_\Delta^{-1} \Delta_0^{-1}, c_\Delta^{-1}\} \max\{\underline{\alpha}^{-1} \underline{\gamma}^{-2}, \underline{\beta}^{-1} \underline{\gamma}^{-2}, \underline{\gamma}^{-3}, \underline{\gamma}^{-2} \underline{\delta}^{-1}\}.
\end{aligned}$$

Combining these bounds with (3.32), the total number of successful iterations is bounded as

$$\begin{aligned}
|\mathcal{S}| &= |\mathcal{S}_1| + |\mathcal{S}_2| + |\mathcal{S}_3^l| + |\mathcal{S}_3^{s,b}| + |\mathcal{S}_3^{s,i,l}| + |\mathcal{S}_3^{s,i,s}| \\
&\leq 2|\mathcal{S}_1| + 2|\mathcal{S}_2| + |\mathcal{S}_3^{s,b}| + |\mathcal{S}_3^{s,i,l}| + 1 + \log_2 \log_2 \left(\frac{2\gamma^2}{\kappa_R \hat{L}_H \varepsilon_g} \right) \\
&\leq C \min\{\underline{\alpha}^2 \underline{\beta}, \underline{\alpha} \underline{\gamma}^2, \underline{\alpha}^2 \underline{\gamma}, \underline{\beta} \underline{\gamma}^2, \underline{\beta}^2 \underline{\gamma}, \underline{\beta}^3, \underline{\gamma}^3, \underline{\gamma}^2 \underline{\delta}\}^{-1} + 1 + \log_2 \log_2 \left(\frac{2\gamma^2}{\kappa_R \hat{L}_H \varepsilon_g} \right),
\end{aligned}$$

where

$$C := 2 \max\{\kappa_H, c_\Delta^{-1} \Delta_0^{-1}, c_\Delta^{-1}\} + 3 \max\{c_\Delta^{-2} \Delta_0^{-2}, c_\Delta^{-2}\} + \nu_R^{-2} + \frac{\kappa_R \hat{L}_H \max\{c_Q^{-1}, c_\Delta^{-1} \Delta_0^{-1}, c_\Delta^{-1}\}}{2}. \quad (3.33)$$

Since the bound on $|\mathcal{S}|$ holds for any K such that no iterate satisfying (3.25) was computed prior to iteration K , the desired result holds. \square

Theorem 3.16 gives a complexity result related to the number of successful iterations (which also corresponds to the number of derivative evaluations). To account for the total number of iterations (or, equivalently, function evaluations) that are required to satisfy (3.25), we follow a common strategy and show that this number is at most a constant multiple of the number of successful iterations.

Lemma 3.17. *Under the assumptions of Lemma 3.4, let $K \in \mathbb{N}$ and let \mathcal{S}_K denote the set of successful steps of index $k \geq K$. Then,*

$$|\mathcal{S}_K| \geq \frac{\log_{\tau_2}(1/\tau_1)}{1 + \log_{\tau_2}(1/\tau_1)}(K + 1) - \frac{1}{1 + \log_{\tau_2}(1/\tau_1)} \max \left(0, \log_{\tau_2} \left(\frac{1}{c_\Delta} \right), \log_{\tau_2} \left(\frac{\Delta_0}{c_\Delta \alpha} \right), \log_{\tau_2} \left(\frac{\Delta_0}{c_\Delta \beta} \right), \log_{\tau_2} \left(\frac{\Delta_0}{c_3 \gamma} \right) \right).$$

Proof. The proof follows verbatim (Boumal, 2023, Lemma 6.23) with (3.13) replacing (Boumal, 2023, (6.26)) and τ_1, τ_2 replacing $\frac{1}{4}$ and 2, respectively. \square

Combining Theorem 3.16 with Lemma 3.17 results in the following complexity result.

Theorem 3.18 (Iteration complexity of Algorithm 1). *Under the assumptions of Theorem 3.16, Algorithm 1 produces a point that satisfies (3.25) in at most*

$$\begin{aligned} & \frac{1 + \log_{\tau_2}(1/\tau_1)}{\log_{\tau_2}(1/\tau_1)} \left[\frac{C}{\min \{ \underline{\alpha}^2 \underline{\beta}, \underline{\alpha} \underline{\gamma}^2, \underline{\alpha}^2 \underline{\gamma}, \underline{\beta} \underline{\gamma}^2, \underline{\beta}^2 \underline{\gamma}, \underline{\beta}^3, \underline{\gamma}^3, \underline{\gamma}^2 \underline{\delta} \}} + 1 + \log_2 \log_2 \left(\frac{2\gamma^2}{\kappa_R \hat{L}_H \varepsilon_g} \right) \right] \\ & + \frac{1}{\log_{\tau_2}(1/\tau_1)} \max \left(0, \log_{\tau_2} \left(\frac{1}{c_\Delta} \right), \log_{\tau_2} \left(\frac{\Delta_0}{c_\Delta \alpha} \right), \log_{\tau_2} \left(\frac{\Delta_0}{c_\Delta \beta} \right), \log_{\tau_2} \left(\frac{\Delta_0}{c_3 \gamma} \right) \right) \end{aligned} \quad (3.34)$$

iterations, where $C, \underline{\alpha}, \underline{\beta}, \underline{\gamma}, \underline{\delta}$ are defined as in Theorem 3.16.

The bound of Theorem 3.18 holds for any values $\varepsilon_g > 0$ and $\varepsilon_H > 0$, but is mostly of interest when $\varepsilon_g < \alpha$ and $\varepsilon_H < \beta$. In that case, the iteration complexity (3.34) is an improvement over the $\mathcal{O}(\max(\varepsilon_g^{-2} \varepsilon_H^{-1}, \varepsilon_H^{-3}))$ bound of Riemannian trust-region methods for generic nonconvex functions Boumal et al. (2019). In addition, our bound guarantees that Algorithm 1 reaches $x_k \in \mathcal{R}_3$ such that

$$\|\text{grad}f(x_k)\| \leq \varepsilon_g \quad \text{and} \quad \lambda_{\min}(\text{Hess}f(x_k)) \geq \gamma, \quad (3.35)$$

and moreover this point is at distance at most δ of a local minimum. In that sense, the strict saddle property allows for obtaining stronger guarantees with improved complexity bounds. This ensures that the algorithm finds an approximate minimizer. On the other hand, if $\varepsilon_g \geq \alpha$ and $\varepsilon_H \geq \beta$, it is enough to find a point that is not in $\mathcal{R}_1 \cup \mathcal{R}_2$ to satisfy the termination criterion (1.2). Finally, if $\varepsilon_g \geq \alpha$ or $\varepsilon_H \geq \beta$, an $(\varepsilon_g, \varepsilon_H)$ -critical point (1.1) might not belong to \mathcal{R}_3 . The following table indicates the possible regions in which Algorithm 1 can terminate depending on the values of ε_g and ε_H .

Convergence of Algorithm 1	$\alpha > \varepsilon_g$	$\alpha \leq \varepsilon_g$
$\beta > \varepsilon_H$	\mathcal{R}_3	$\mathcal{R}_1 \cup \mathcal{R}_3$
$\beta \leq \varepsilon_H$	$\mathcal{R}_2 \cup \mathcal{R}_3$	$\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$

In principle, it is always possible to lower ε_H and ε_g to ensure convergence to a minimizer. This does not have a meaningful impact on the worst-case iteration complexity (3.34).

To end this section, we translate our complexity results on our examples of strict saddle functions from Section 2.

Example 3.1. *Suppose that $\mathcal{M} = \mathbb{R}^n$, and that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a γ -strongly convex function with global minimum x^* . Choose $\alpha = 1$ so that the function is $(1, 1, \gamma, \frac{2}{\gamma})$ -strict saddle, and let $\varepsilon_g \in (0, 1)$. Then, by Theorem 3.18, Algorithm 1 computes an iterate such that $\|\nabla f(x_k)\| \leq \varepsilon_g$*

in at most $\mathcal{O}(\gamma^{-3}) + \log \log(\gamma^2 \varepsilon_g^{-1})$ iterations. In comparison, a standard analysis of Newton's method with Armijo backtracking linesearch requires at most $\mathcal{O}(\gamma^{-5}) + \log \log(\gamma^3 \varepsilon_g^{-1})$ iterations to find such point (Boyd and Vandenberghe, 2004). Although our bound appears better in terms of dependencies on γ , we believe it to be an artifact of the line-search analysis, that could possibly be improved by changing the line-search condition.

Example 3.2. Suppose that \mathcal{M} is the unit sphere in \mathbb{R}^n , denoted by \mathbb{S}^{n-1} and let $f : \mathbb{S}^{n-1} \rightarrow \mathbb{R}$ be defined by $f(x) = x^\top A x$, where $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix with eigenvalues $\lambda_1 > \lambda_2 \geq \dots \geq \lambda_{n-1} > \lambda_n$. Then, Theorem 3.18, Algorithm 1 computes an iterate satisfying (3.25) in at most

$$\mathcal{O}\left(\max\{1, (\lambda_1 - \lambda_n)^{-3}, \lambda_1^2 (\lambda_1 - \lambda_n)^{-3}\}\right) + \log \log\left((\lambda_1 - \lambda_n)^{-2} \varepsilon_g^{-1}\right).$$

4 A strict saddle Riemannian trust-region with inexact subproblem solutions

In this section, we design a Riemannian trust-region with inexact minimization of the subproblems which exploits the strict saddle structure. Near critical points, the smallest eigenvalue of the Hessian is bounded away from zero. These subproblems reduce to one of two much simpler tasks: computing a direction of negative curvature or solving a strongly convex subproblem. Away from critical points, the gradient norm is large and it is cheap to approximately minimize the subproblem by computing a direction that resembles a gradient descent step.

The solution of the subproblems is based on the well-known truncated CG framework (Toint, 1981; Steihaug, 1983). The strict-saddle property (Definition 2.4) with knowledge of the strict saddle parameters α, β, γ allows to refine this procedure and select steps that are appropriate for the local landscape, which leads to improved complexity bounds. In order to achieve local quadratic convergence towards minimizers, we use a stopping criterion for tCG with a squared norm of the gradient:

$$\|r_{j+1}\| \leq \zeta \min\{\|g_k\|^2, \kappa \|g_k\|, \gamma \|y_{j+1}\|\} \quad (4.1)$$

for some $\zeta \in (0, 1)$, where $r_j = \nabla m_k(y_j)$ is the residual of the CG algorithm after j iterations.

If all the directions generated by CG are γ -strongly convex, the sufficient decrease condition (4.1) is satisfied in at most k_{\max} iterations. If k_{\max} iterations of CG are performed and that (4.1) is not met, it means that the condition $H_k \succeq \gamma \text{Id}$ does not hold and $x_k \notin \mathcal{R}_3$. Similarly if CG detects a direction of curvature smaller than γ , this is a certificate that $x_k \notin \mathcal{R}_3$. For those iterations, if $\|g_k\| \geq \alpha$, the direction that has been computed by CG will do at least as well as the Cauchy point, which ensures a decrease of (3.6). However when $\|g_k\| < \alpha$, it must be that $x_k \in \mathcal{R}_2$ and we wish to compute a direction of negative curvature at least $-\beta/2$. It is possible to compute a full eigenvalue decomposition of the Hessian using n Hessian-vector products. We want to avoid this and used a randomized iterative procedure to estimate the smallest eigenvalue of the Hessian. We use the Lanczos method which finds the smallest eigenvalue of a symmetric matrix in a Krylov subspace according to an initial random vector. If the initial vector is random, the dimension of the Krylov subspace increases by one at each iteration with high probability (Royer et al., 2020, Appendix B).

We run Krylov for sufficiently many iterations such that the probability of failure is very small. For a chosen probability of failure $p \in (0, 1)$, the number of iterations requires by Lanczos to find a direction of curvature $-\beta/2$ with probability at least $1 - p$ is given in Lemma 4.1. If Lanczos fails to compute the negative curvature, we choose to use the direction previously computed by CG, which at least guarantees a decrease in the model. When Lanczos fails, one could restart Lanczos with a different initial random vector until it finds the curvature $-\beta/2$;

Algorithm 2 Strict saddle RTR with inexact subproblem minimization

- 1: **Given:** Tolerance $\varepsilon_g < \alpha$, ε_H , Constants α, β, γ of the strict saddle function, $x_0 \in \mathbb{R}^n$, trust-region radius $\Delta_0 > 0$, $\bar{\Delta} > 0$, constants $0 < \eta_1 < \eta_2 < 1$ and $0 < \tau_1 < 1 < \tau_2$.
- 2: **for** $k = 1, 2, \dots$ **do**
- 3: Use truncated CG (Algorithm 3) to

$$s_k \leftarrow \underset{s \in T_{x_k} \mathcal{M}}{\text{minimize}} m_k(s) \text{ such that } \|s\| \leq \Delta_k. \quad (4.2)$$

- 4: **if** $\|g_k\| < \varepsilon_g$ or (`outputCG` \in `{max_iter, not_strongly_convex}` and $\|g_k\| < \alpha$) **then**
- 5: $s_k \leftarrow \text{MEO}(H_k, s_k)$ ▷ Negative curvature Algorithm 4
- 6: **end if**
- 7: Compute

$$\rho = \frac{f(x_k) - f(R_{x_k}(s_k))}{m_k(0) - m_k(s_k)}$$

- 8: Set

$$x_{k+1} = \begin{cases} x_k & \text{if } \rho < \eta_1 \\ x_k + s_k & \text{if } \eta_1 \leq \rho \end{cases} \quad (4.3)$$

- 9: Update

$$\Delta_{k+1} = \begin{cases} \tau_1 \Delta_k & \text{if } \rho < \eta_1 & \text{[unsuccessful]} \\ \Delta_k & \text{if } \eta_1 \leq \rho \leq \eta_2 & \text{[successful]} \\ \tau_2 \Delta_k & \text{if } \rho > \eta_2 & \text{[very successful]} \end{cases} \quad (4.4)$$

- 10: $k \leftarrow k + 1$

- 11: **end for**
-

or compute a full eigenvalue decomposition directly. We do neither of those things. Instead, our complexity guarantees for convergence to a second-order critical point hold under the high probability event that Lanczos does not fail.

When the minimum eigenvalue oracle is called and the step is unsuccessful, it is possible to store the vectors generated by the Lanczos algorithm to avoid recomputing them until a successful step is found, see (Curtis et al., 2021).

Algorithm 3 truncated CG for strict saddle function (case 2 and 3)

Input: Nonzero $g_k \in \mathbb{T}_{x_k} \mathcal{M}$ and H_k , accuracy parameters $\kappa \in (0, 1), \zeta \in (0, 1)$
Output: trial step s and flag `outputCG` indicating termination type

Set $g_k = \text{grad}f(x_k)$ and $H_k = \text{Hess}f(x_k)$
Set $k_{max} \leftarrow$ Equation (4.22)
 $y_0 \leftarrow 0, r_0 \leftarrow g_k, p_0 \leftarrow -g_k, j \leftarrow 0$
while $j < k_{max}$ **do**
 if $\langle y_j, H_k y_j \rangle < \gamma \|y_j\|^2$ **then**
 Set $d = \Delta_k y_j / \|y_j\|$ and terminate with `outputCG = not_strongly_convex`
 end if
 if $\langle p_j, H_k p_j \rangle < \gamma \|p_j\|^2$ **then**
 Set $d = \Delta_k p_j / \|p_j\|$ and terminate with `outputCG = not_strongly_convex`
 end if
 $\alpha_j \leftarrow \|r_j\|^2 / \langle p_j, H_k p_j \rangle$ ▷ Begin standard tCG procedure
 $y_{j+1} = y_j + \alpha_j p_j$
 if $\|y_{j+1}\| \geq \Delta_k$ **then**
 Compute $\bar{\alpha}_j \geq 0$ such that $\|y_j + \bar{\alpha}_j p_j\| = \Delta_k$
 return $s \leftarrow y_j + \bar{\alpha}_j p_j$ and `outputCG = boundary_step`
 end if
 $r_{j+1} \leftarrow r_j + \alpha_j H_k p_j$
 if $\|r_{j+1}\| \leq \zeta \min\{\|g_k\|^2, \kappa \|g_k\|, \gamma \|y_{j+1}\|\}$ **then**
 return $s \leftarrow y_{j+1}$ and `outputCG = small_residual`
 end if
 $\beta_{j+1} \leftarrow \|r_{j+1}\|^2 / \|r_j\|^2$
 $p_{j+1} \leftarrow -r_{j+1} + \beta_{j+1} p_j$ ▷ end standard tCG procedure
 $j \leftarrow j + 1$
end while
return $s \leftarrow y_{k_{max}}$ and `outputCG = max_iter`

A minimum eigenvalue oracle

Lemma 4.1 (Lemma 2 from Royer et al. (2020)). *Suppose that the Lanczos method is used to estimate the smallest eigenvalue of H_k starting with a random vector uniformly generated on the unit sphere, where $\|H_k\| \leq \kappa_H$. For any $\delta \in [0, 1)$, this approach finds the smallest eigenvalue of H_k to an absolute precision of $\beta/2$, together with a corresponding direction v , in at most*

$$N_{\text{meo}} := \min \left\{ n, 1 + \left\lceil \frac{1}{2} \ln(2.75n/p^2) \sqrt{\frac{\kappa_H}{\beta}} \right\rceil \right\} \quad (4.5)$$

iterations, with probability at least $1 - p$, where $\dim(\mathcal{M}) = n$.

Algorithm 4 A minimum eigenvalue oracle

Input: Nonzero $g_k \in \mathbb{T}_{x_k} \mathcal{M}$ and H_k , trust-region radius Δ_k , failure probability tolerance $p \in (0, 1)$ and $M \in [\|H\|, \infty)$, direction computed by tCG s_k .

Run the Lanczos algorithm on the matrix H_k with random initial vector to find a direction of curvature at most $-\beta/2$. Run at most (4.5) iterations so that the probability of failure is at most p .

if Lanczos find vector v such that $\|v\| = 1$ and $\langle v, H_k v \rangle \leq -\beta/2$ **then**

return $w = \pm \Delta_k v$ satisfying

$$\langle g_k, w \rangle \leq 0, \quad \langle w, H_k w \rangle \leq -\frac{1}{2}\beta \|w\|^2, \quad \text{and} \quad \|w\| = \Delta_k,$$

else

return s_k

 ▷ MEO failed to find $-\beta/2$ curvature

end if

4.1 Complexity analysis for inexact subproblems

We analyse the decrease in the model for each possible scenario that triggers the termination of the truncated CG algorithm 3.

Proposition 4.2 (tCG small residual). *Let iteration k be successful with $\|s_k\| \leq \nu_R$ and Algorithm 3 terminate with `outputCG=small_residual`, then*

$$m_k(0) - m_k(s_k) \geq \frac{\gamma}{4} \|s_k\|^2. \quad (4.6)$$

If additionally $\|s_k\| \leq \nu_R$, then

$$m_k(0) - m_k(s_k) \geq \frac{\gamma}{4} \|s_k\|^2 \geq \frac{1}{2(\kappa_R^2 + 2L\kappa_R)} \min(\|\text{grad}f(x_{k+1})\|^2 \gamma^{-1}, \gamma^3). \quad (4.7)$$

Proof. The final iterate of CG—written s_k or y_{j+1} —satisfies the small residual condition (4.1). We have

$$\begin{aligned} \langle s_k, H_k s_k \rangle &= \langle y_j + \alpha_j p_j, H_k(y_j + \alpha_j p_j) \rangle \\ &= \langle y_j, H_k y_j \rangle + \alpha_j \langle y_j, H_k p_j \rangle + \alpha_j \langle p_j, H_k y_j \rangle + \alpha_j^2 \langle p_j, H_k p_j \rangle \\ &= \langle y_j, H_k y_j \rangle + \alpha_j^2 \langle p_j, H_k p_j \rangle, \end{aligned} \quad (4.8)$$

where the last line follows from $y_j = \sum_{i=0}^{j-1} \alpha_i p_i$ and $\langle p_j, H_k p_i \rangle = 0$ for $i \neq j$, which implies $\langle p_j, H_k y_j \rangle = 0$. Since Algorithm 3 certifies $\langle p_j, H_k p_j \rangle \geq \gamma \|p_j\|^2$ and $\langle y_j, H_k y_j \rangle \geq \gamma \|y_j\|^2$, this gives

$$\begin{aligned} \langle s_k, H_k s_k \rangle &\geq \gamma \|y_j\|^2 + \gamma \|\alpha_j p_j\|^2 \\ &\geq \frac{\gamma}{2} \|s_k\|^2, \end{aligned} \quad (4.9)$$

where we use $\|u\|^2 + \|v\|^2 \geq \frac{1}{2} \|u + v\|^2$ for all vectors $u, v \in \mathbb{T}_{x_k} \mathcal{M}$. This allows to deduce a

model decrease

$$\begin{aligned}
m_k(0) - m_k(s_k) &= -\langle s_k, g_k \rangle - \frac{1}{2} \langle s_k, H_k s_k \rangle \\
&= -\langle s_k, -H_k s_k + r_{j+1} \rangle - \frac{1}{2} \langle s_k, H_k s_k \rangle \\
&= \frac{1}{2} \langle s_k, H_k s_k \rangle - \langle r_{j+1}, s_k \rangle \\
&\geq \frac{\gamma}{4} \|s_k\|^2, \tag{4.10}
\end{aligned}$$

where we used that $r_{j+1} = H_k y_{j+1} + g_k$, $\langle r_{j+1}, y_{j+1} \rangle = \langle r_{j+1}, \sum_{i=0}^j \alpha_i p_i \rangle = 0$ since for all $i < j$, $r_{j+1} \perp \text{span}(p_0, \dots, p_j)$ and finally Equation (4.9). Mimicking the proof of Lemma 3.7 gives

$$\|\text{grad}f(x_{k+1})\| \leq \kappa_R \left\| \nabla \hat{f}_k(s_k) - \text{grad}f(x_k) + \text{grad}f(x_k) \right\| \tag{4.11}$$

$$= \kappa_R \left\| \nabla \hat{f}_k(s_k) - \text{grad}f(x_k) - H_k y_{j+1} + r_{j+1} \right\| \tag{4.12}$$

$$\leq \kappa_R \left\| \nabla \hat{f}_k(s_k) - \text{grad}f(x_k) - H_k s_k \right\| + \kappa_R \|r_{j+1}\| \tag{4.13}$$

$$\leq \frac{\hat{L}_H \kappa_R}{2} \|s_k\|^2 + \kappa_R \gamma \|s_k\|, \tag{4.14}$$

where the last line follows from the Lipschitz-type inequality of Lemma 3.5 and the small residual condition (4.1). We consider the univariate quadratic

$$\frac{L\kappa_R}{2} \|s_k\|^2 + \kappa_R \gamma \|s_k\| - \|g_{k+1}\| \geq 0$$

as a function of $\|s_k\|$. The convex quadratic has two real roots of opposite signs. The only feasible values for $\|s_k\|$ are positive and therefore greater than the positive root; this gives

$$\begin{aligned}
\|s_k\| &\geq \frac{-\kappa_R \gamma + \sqrt{\kappa_R^2 \gamma^2 + 2L\kappa_R \|g_{k+1}\|}}{L\kappa_R} \\
&= \left[\frac{-\kappa_R + \sqrt{\kappa_R^2 + 2L\kappa_R \|g_{k+1}\| \gamma^{-2}}}{L\kappa_R} \right] \gamma \\
&\geq \left(\frac{-c_R + \sqrt{c_R^2 + 2Lc_R}}{Lc_R} \right) \min\{\|g_{k+1}\| \gamma^{-2}, 1\} \gamma \\
&= \left(\frac{2}{c_R + \sqrt{\kappa_R^2 + 2L\kappa_R}} \right) \min\{\|g_{k+1}\| \gamma^{-1}, \gamma\},
\end{aligned}$$

where we used that $-a + \sqrt{a^2 + bt} \geq (-a + \sqrt{a^2 + b}) \min(t, 1)$ with $a = c_R$, $b = 2Lc_R$ and $t = \|g_{k+1}\| \gamma^{-2}$. The constant in brackets can be simplified further by using that for any $e > 0$, $2/(1 + \sqrt{1 + e}) \geq 1/\sqrt{1 + e}$. Combining (4.9) and (4.10) we conclude

$$\begin{aligned}
m_k(0) - m_k(s_k) &\geq \frac{1}{4} \gamma \|s_k\|^2 \\
&\geq \frac{1}{2(\kappa_R^2 + 2L\kappa_R)} \min\left(\|g_{k+1}\|^2 \gamma^{-1}, \gamma^3\right). \quad \square
\end{aligned}$$

Proposition 4.3 (tCG Boundary step). *If $\text{outputCG} = \text{boundary_step}$, then*

$$m_k(0) - m_k(s_k) \geq \frac{\gamma}{4} \Delta_k^2 \quad (4.15)$$

Proof. Let $H_k = \text{Hess}f(x_k)$ and note that $\|r_j\|^2 = -\langle r_j, p_j \rangle = -\langle g_k, p_j \rangle$. The direction is $s_k = y_j + \bar{\alpha}_j p_j$ with $\|s_k\| = \Delta_k$ for some

$$0 \leq \bar{\alpha}_j \leq \alpha_j = -\langle g_k, p_j \rangle / \langle p_j, H_k p_j \rangle, \quad (4.16)$$

where α_j is the conjugate gradients stepsize that leaves the trust-region, $\|y_j + \alpha_j p_j\| \geq \Delta_k$. Algorithm 3 ensures that $\langle p_j, H_k p_j \rangle \geq \gamma \|p_j\|^2$. Equation (4.16) gives

$$-\bar{\alpha}_j \langle g_k, p_j \rangle \geq \bar{\alpha}_j^2 \langle p_j, H_k p_j \rangle,$$

from which we deduce

$$m_k(0) - m_k(\bar{\alpha}_j p_j) = -\bar{\alpha}_j \langle g_k, p_j \rangle - \frac{\bar{\alpha}_j^2}{2} \langle p_j, H_k p_j \rangle \quad (4.17)$$

$$\geq \bar{\alpha}_j^2 \langle p_j, H_k p_j \rangle - \frac{\bar{\alpha}_j^2}{2} \langle p_j, H_k p_j \rangle \quad (4.18)$$

$$= \frac{\bar{\alpha}_j^2}{2} \langle p_j, H_k p_j \rangle \quad (4.19)$$

$$\geq \frac{\gamma}{2} \|\bar{\alpha}_j p_j\|^2. \quad (4.20)$$

Equation (4.10) with $\langle y_j, H_k y_j \rangle \geq \gamma \|y_j\|^2$ gives $m_k(0) - m_k(y_j) \geq \frac{\gamma}{2} \|y_j\|^2$. Equation (4.8) implies $m_k(s_k) = m_k(y_j) + m_k(\bar{\alpha}_j p_j)$ and we conclude

$$\begin{aligned} m_k(0) - m_k(s_k) &= m_k(0) - m_k(y_j) + m_k(0) - m_k(\bar{\alpha}_j p_j) \\ &\geq \frac{\gamma}{2} \|y_j\|^2 + \frac{\gamma}{2} \|\bar{\alpha}_j p_j\|^2 \\ &\geq \frac{\gamma}{2} \left(\|y_j\|^2 + \|\bar{\alpha}_j p_j\|^2 \right) \\ &\geq \frac{\gamma}{4} \|s_k\|^2, \end{aligned}$$

where we use $\|u\|^2 + \|v\|^2 \geq \frac{1}{2} \|u + v\|^2$ for all vectors $u, v \in \mathbb{T}_{x_k} \mathcal{M}$. □

Lemma 4.4 (Lemma 11 in [Royer and Wright \(2018\)](#)). *Let $x_k \in \mathcal{R}_3$ with $\|g_k\| > \varepsilon_g$. Assume A6 such that $\gamma \text{Id} \preceq H_k \preceq \kappa_H \text{Id}$. The conjugate gradient algorithm applied to the linear system $H_k s = -g_k$ computes a vector $s_k \in \mathbb{T}_{x_k} \mathcal{M}$ such that*

$$\|H_k s_k + g_k\| \leq \zeta \min \left\{ \|g_k\|^2, \kappa \|g_k\|, \gamma \|s_k\| \right\} \quad (4.21)$$

for some $\zeta \in (0, 1)$ in at most

$$k_{\max} := \min \left\{ n, \frac{1}{2} \sqrt{\frac{\kappa_H}{\gamma}} \ln \left(\frac{2\sqrt{\kappa_H}}{\zeta\sqrt{\gamma}} \max(\varepsilon_g^{-1}, \kappa^{-1}, \kappa_H/\gamma) \right) \right\} \quad (4.22)$$

iterations, where $\varrho = \kappa_H/\gamma$.

Proof. The result follows from ([Royer and Wright, 2018](#), Lemma 11). □

Proposition 4.5 (Not convex tCG). *Assume that Algorithm 3 terminates with flag `not_strongly_convex`, then*

$$m_k(0) - m_k(s_k) \geq \frac{1}{4} \min \left\{ \Delta_k^2 \beta, \min(\Delta_k, \frac{\alpha}{\kappa_H}) \alpha \right\} \quad (4.23)$$

with probability at least $1 - p$.

Proof. If $\|g_k\| \geq \alpha$, the step s_k (given by tCG) decreases the model at least as much as the Cauchy step, $m_k(0) - m_k(s_k) \geq \frac{1}{2} \min(\Delta_k, \alpha/\kappa_H) \alpha$. If $\|g_k\| < \alpha$, Algorithm 4 returns a unit-norm direction $v \in T_{x_k} \mathcal{M}$ such that $\langle v, H_k v \rangle \leq -\frac{1}{2} \beta$. Using Lemma 3.2, we have (4.23) with probability at least p . \square

4.2 Local convergence of inexact RTR

We analyse the local convergence of Algorithm 2 towards minimizers using strong geodesic convexity. If $H_k \succeq \gamma \text{Id}$, the tCG step s_k is smaller in norm than any global minimizer of the trust-region subproblem. Using (3.20) gives

$$\|s_k^{\text{tCG}}\| \leq \|s_k^{\text{exact}}\| \leq \|g_k\| / \gamma. \quad (4.24)$$

Proposition 4.6. *Let $x_k \in \mathcal{R}_3$ with $\|s_k\| \leq \nu_R$ and*

$$\|g_k\| \leq \min \{ \xi_2, \gamma \Delta_k \}, \quad (4.25)$$

where ν_R comes from Proposition 3.6 and $\xi_2 = \min(c_Q, 2/(\kappa_R(\hat{L}_H + 2)))$. Algorithm 3 returns a point in \mathcal{R}_3 with gradient norm below ε_g in at most

$$\log_2 \log_2 \left(\frac{2\gamma^2}{\kappa_R(L+2)\varepsilon_g} \right) \quad (4.26)$$

iterations.

Proof. Consider $x^* \in \mathcal{R}_3$, a local minimizer of f such that $\text{dist}(x_k, x^*) \leq \delta$. Since $\|g_k\| \leq \gamma \Delta_k$, the Newton step is inside the trust region. Therefore, the tCG step s_k satisfies (4.1) and is no greater in norm than the exact subproblem minimizer. To show a successful step, the proof of Proposition 3.10 holds mutatis mutandis with the realisation that $\|s_k\| \leq \|g_k\| / \gamma$ and $m(0) - m(s_k) \geq \frac{1}{2} \gamma \|s_k\|^2$ comes from Proposition 4.2. We then proceed from (4.13), using (4.24) and (4.1)

$$\|g_{k+1}\| \leq c_R \left\| \nabla \hat{f}_k(s_k) - g_k - H_k s_k \right\| + c_R \|r_{j+1}\| \quad (4.27)$$

$$\leq \kappa_R \frac{L}{2} \|s_k\|^2 + \kappa_R \|g_k\|^2 \quad (4.28)$$

$$\leq \frac{L\kappa_R}{2\gamma^2} \|g_k\|^2 + \kappa_R \|g_k\|^2 \quad (4.29)$$

$$\leq \kappa_R \frac{L+2}{2\gamma^2} \|g_k\|^2. \quad (4.30)$$

Using (4.25) shows that $\|g_{k+1}\| \leq \frac{1}{2} \|g_k\|$. Using that fact, the proof of Proposition 3.12 with the appropriate minor adjustments ensures that $\text{dist}(x_{k+1}, x^*) < \delta$. Furthermore, $\|g_{k+1}\| <$

$\|g_k\| < \gamma\Delta_k \leq \gamma\Delta_{k+1}$; and, by induction, all subsequent iterates will be at distance at most δ of x^* , and each iteration will be a successful step with tCG terminating with a small residual. To show quadratic convergence, we have

$$\kappa_R \frac{(L+2)}{2\gamma^2} \|g_{k+1}\| \leq \left(\kappa_R \frac{(L+2)}{2\gamma^2} \|g_k\| \right)^2. \quad (4.31)$$

Following the proof of Theorem 3.15, we reach a point $x_\ell \in \mathcal{R}_3$ with $\|\text{grad}f(x_\ell)\|$ in a number of iterations upper bounded by (4.26). \square

4.3 Complexity of inexact RTR

Lemma 4.7 (Lower bound on TR radius). *As long as Algorithm 2 as not returned, the trust region radius satisfies $\Delta_k \geq \mathcal{O}(\Delta_0, \alpha, \beta, \gamma)$.*

Proof. The proof from Proposition 3.4 hold mutatis mutandis. If $x_k \in \mathcal{R}_1$, then the Cauchy decrease holds (3.6). If $x_k \in \mathcal{R}_2 \cup \mathcal{R}_3$, then the decrease is either (4.6) (small residual), (4.15) (boundary step) or (4.23). \square

Theorem 4.8 (Successful steps of inexact RTR). *For any realization of a run of Algorithm 2, the number of successful iterations performed before termination is at most*

$$K_S := \mathcal{O} \left(\max(1/\underline{\alpha}^2, 1/\underline{\beta}^3, 1/\underline{\gamma}^3) + \log \log(\underline{\gamma}\varepsilon_g^{-1}) \right). \quad (4.32)$$

with probability $(1-p)^{K_S}$.

Proof. If $x_k \in \mathcal{R}_1$, for any value of `outputCG`, the step s_k does at least as well as the Cauchy step (3.6). If `outputCG = boundary_step`, s_k satisfies (4.15). Let `outputCG = not_strongly_convex` and $x_k \in \mathcal{R}_2$. The MEO is called and s_k satisfies (4.23) with probability at least p . By (Curtis et al., 2021, Theorem 4.6), the MEO avoids failure for K iterations with probability at least $(1-p)^K$.

If `outputCG = small_residual`, we consider large steps and small steps separately according to Lemma 3.7. If $\|s_k\| > \nu_R$, then $m_k(0) - m_k(s_k) \geq \gamma\nu_R^2/4$ by (4.6). If $\|s_k\| \leq \nu_R$, then (4.7) gives

$$m_k(0) - m_k(s_k) \geq \mathcal{O} \left(\min(\|g_{k+1}\| \gamma^{-1}, \gamma^3) \right). \quad (4.33)$$

We proceed similarly to the proof of Theorem 3.16. If $\|g_{k+1}\| \geq \min(\xi_2, \gamma\Delta_k)$, then $m_k(0) - m_k(s_k) \geq \mathcal{O} \left(\min(\xi_2\gamma^{-1}, \Delta_k, \gamma^3) \right)$. If $\|g_{k+1}\| < \min(\xi_2, \gamma\Delta_k)$ and $x_{k+1} \in \mathcal{R}_3$, then local convergence starts at x_{k+1} towards a minimizer by Proposition 4.6. The number of iterations such that $\|g_{k+1}\| < \min(\xi_2, \gamma\Delta_k)$ and $x_{k+1} \in \mathcal{R}_1$ is bounded by the number of iterations in \mathcal{R}_1 , labelled $|\mathcal{S}_1|$. If, on the other hand, $\|g_{k+1}\| < \min(\xi_2, \gamma\Delta_k)$ and $x_{k+1} \in \mathcal{R}_2$, then we cannot have a sufficient guarantee of decrease until the MEO is called or another termination triggers the end of CG. There can be at most (4.26) iterations in a row that remain in \mathcal{R}_2 and where tCG terminates with small residual, before another type of iteration occurs (producing a guaranteed decrease) or the gradient norm drops below ε_g , which triggers the call to a MEO, which will produce a decrease or ensure that the iterate is a $(\varepsilon_g, \varepsilon_H)$ -SOCP (1.1). We put every type of

decrease together until the start of the local phase. For any $\bar{K} \in \mathbb{N}$:

$$\begin{aligned}
f(x_0) - f^* &\geq f(x_0) - f(x_K) = \sum_{k=0}^{\bar{K}-1} f(x_k) - f(x_{k+1}) \\
&\geq \sum_{k \in B} f(x_k) - f(x_{k+1}) + \sum_{k \in \text{MEO}} f(x_k) - f(x_{k+1}) + \sum_{k \in \text{small residual}} f(x_k) - f(x_{k+1}) \\
&\geq |K| \cdot \frac{1}{2} \eta_1 \min \left\{ \frac{\alpha^2}{\kappa_H}, \Delta_{\min} \alpha, \beta \Delta_{\min}^2, \gamma \nu_R^2, \xi_2 \gamma^{-1}, \Delta_{\min}, \gamma^3 \right\} \\
&= |K| \cdot \frac{1}{2} \eta_1 \min \{ \kappa_H^{-1}, c_Q, \Delta_0, \Delta_0^2, \nu_R^2 \} \min \{ \underline{\alpha}^2, \underline{\beta} \underline{\alpha}^2, \underline{\beta}^3, \underline{\delta}, \underline{\beta} \underline{\gamma}^2, \underline{\gamma}^3 \}.
\end{aligned}$$

We account for iterations that may terminate with a small residual of CG when a call to the MEO is required to show a decrease. We find that the number of successful steps before the local phase begins is at most

$$K_S = \frac{2(f(x_0) - f^*)}{\eta_1 \min \{ \kappa_H^{-1}, c_Q, \Delta_0, \Delta_0^2, \nu_R^2 \}} \max \{ \underline{\alpha}^{-2}, \underline{\beta}^{-1} \underline{\alpha}^{-2}, \underline{\beta}^{-3}, \underline{\delta}^{-1}, \underline{\beta}^{-1} \underline{\gamma}^{-2}, \underline{\gamma}^{-3} \} \quad (4.34)$$

successful iterations. \square

Lemma 4.9 (Number of unsuccessful steps for inexact RTR). *Let x_0, \dots, x_n be n iterates generated by Algorithm 2. Define the set of successful steps as*

$$\mathcal{S}_n = \{j \in \{0, \dots, n\} : \rho_j \geq \eta_1\} \quad (4.35)$$

and let \mathcal{U}_n designate the unsuccessful steps, so that \mathcal{S}_n and \mathcal{U}_n form a partition of $\{0, \dots, n\}$. If none of them satisfy the termination conditions, it holds that

$$|\mathcal{S}_n| \geq (1 - \tau_2)(n + 1) - \tau_2 \max \left(0, \log_{\tau_2} \left(\frac{\Delta_0}{c_1 \alpha} \right), \log_{\tau_2} \left(\frac{\Delta_0}{c_2 \beta} \right), \log_{\tau_2} \left(\frac{\Delta_0}{c_3 \gamma} \right) \right) \quad (4.36)$$

Proof. The proof follows (Boumal, 2023, Lemma 6.23) and rests on the lower-bound for Δ_k from Lemma 4.7. \square

Theorem 4.10. *Algorithm 2 returns a $(\varepsilon_g, \varepsilon_H)$ -critical point (1.1) with probability at least $(1 - p)^{K_S}$ in at most*

$$K_S (1 - \tau_2)^{-1} + \tau_2 \max \left(0, \log_{\tau_2} \left(\frac{\Delta_0}{c_1 \alpha} \right), \log_{\tau_2} \left(\frac{\Delta_0}{c_2 \beta} \right), \log_{\tau_2} \left(\frac{\Delta_0}{c_3 \gamma} \right) \right) (1 - \tau_2)^{-1} \quad (4.37)$$

iterations, where K_S is defined in Theorem 4.8.

Proof. See (Curtis et al., 2021, Theorem 4.6). \square

Since tCG performs at most k_{\max} iterations and the MEO performs at most N_{meo} iterations, the number of Hessian-vector products is bounded by $(k_{\max} + N_{\text{meo}}) |K| =$ where the total number of iterations $|K|$ is bounded by Theorem 4.10, k_{\max} is defined in (4.22) and N_{meo} in (4.5). Thus, we have the following result.

Theorem 4.11 (Complexity of Algorithm 2). *For any realization of Algorithm 2, the total number of Hessian-vector products performed before producing a $(\varepsilon_g, \varepsilon_H)$ -critical point (1.1) is at most*

$$\tilde{O} \left(\min \left(n, \underline{\beta}^{-1/2}, \underline{\gamma}^{-1/2} \right) \max \{ \underline{\alpha}^{-2}, \underline{\beta}^{-1} \underline{\alpha}^{-2}, \underline{\beta}^{-3}, \underline{\delta}^{-1}, \underline{\beta}^{-1} \underline{\gamma}^{-2}, \underline{\gamma}^{-3} \} + \log \log (\varepsilon_g^{-1} \gamma) \right). \quad (4.38)$$

5 Discussion

We have shown that worst-case complexity guarantees of Riemannian trust-region algorithms on nonconvex functions improve significantly when the function satisfies a strict saddle property. In particular, an algorithm with exact subproblem minimization does not require any modification from its standard version in order to benefit from these improved guarantees. Our analysis crucially relies on the fast quadratic local convergence of Newton’s method, and can be adapted to inexact subproblem solves by incorporating knowledge of the strict saddle constants in the problem. Although those parameters are known for a variety of problems, adaptive schemes have been proposed to estimate them as the algorithm unfolds (O’Neill and Wright, 2023). Investigating the numerical performance of these algorithms, along with their multiple possibilities for implementation, will be the subject of future work.

Acknowledgments The authors are grateful to Coralia Cartis for useful discussions regarding local convergence of trust-region methods, and for sharing reference Shek (2015). Funding for this research was partially provided by Agence Nationale de la Recherche through program ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

References

- Absil, P.-A., Baker, C., and Gallivan, K. (2007). Trust-Region Methods on Riemannian Manifolds. *Foundations of Computational Mathematics*, 7(3):303–330.
- Absil, P.-A., Mahony, R., and Sepulchre, R. (2008). *Optimization Algorithms on Matrix Manifolds*. Princeton University Press.
- Agarwal, N., Boumal, N., Bullins, B., and Cartis, C. (2021). Adaptive regularization with cubics on manifolds. *Mathematical Programming*, 188(1):85–134.
- Boumal, N. (2023). *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press.
- Boumal, N., Absil, P.-A., and Cartis, C. (2019). Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, 39(1):1–33.
- Boumal, N., Voroninski, V., and Bandeira, A. S. (2020). Deterministic Guarantees for Burer-Monteiro Factorizations of Smooth Semidefinite Programs. *Communications on Pure and Applied Mathematics*, 73(3):581–608.
- Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. (2018). Accelerated Methods for Non-Convex Optimization. *SIAM Journal on Optimization*, 28(2):1751–1772.
- Cartis, C., Gould, N. I. M., and Toint, P. L. (2012). Complexity bounds for second-order optimality in unconstrained optimization. *Journal of Complexity*, 28(1):93–108.
- Criscitiello, C. and Boumal, N. (2019). Efficiently escaping saddle points on manifolds. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Curtis, F. E. and Robinson, D. P. (2021). Regional complexity analysis of algorithms for nonconvex smooth optimization. *Mathematical Programming*, 187(1):579–615.

- Curtis, F. E., Robinson, D. P., Royer, C. W., and Wright, S. J. (2021). Trust-Region Newton-CG with Strong Second-Order Complexity Guarantees for Nonconvex Optimization. *SIAM Journal on Optimization*, 31(1):518–544.
- Ge, R., Lee, J. D., and Ma, T. (2016). Matrix Completion has No Spurious Local Minimum. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 2973–2981. Curran Associates, Inc.
- Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. (2017). How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732. PMLR.
- Lee, J. D., Panageas, I., Piliouras, G., Simchowitz, M., Jordan, M. I., and Recht, B. (2019). First-order methods almost always avoid strict saddle points. *Mathematical programming*, 176(1):311–337.
- Li, Q., Zhu, Z., and Tang, G. (2019). The non-convex geometry of low-rank matrix optimization. *Information and Inference: A Journal of the IMA*, 8(1):51–96.
- Liu, Y. and Roosta, F. (2023). A Newton-MR algorithm with complexity guarantees for non-convex smooth unconstrained optimization.
- Luo, Y. and Trillos, N. G. (2022). Nonconvex Matrix Factorization is Geodesically Convex: Global Landscape Analysis for Fixed-rank Matrix Optimization From a Riemannian Perspective.
- Moré, J. J. and Sorensen, D. C. (1983). Computing a trust region step. *SIAM Journal on scientific and statistical computing*, 4(3):553–572.
- O’Neill, M. and Wright, S. J. (2023). A Line-Search Descent Algorithm for Strict Saddle Functions with Complexity Guarantees. *Journal of Machine Learning Research*, 24(10):1–34.
- Paternain, S., Mokhtari, A., and Ribeiro, A. (2019). A Newton-Based Method for Nonconvex Optimization with Fast Evasion of Saddle Points. *SIAM Journal on Optimization*, 29(1):343–368.
- Qu, Q., Zhai, Y., Li, X., Zhang, Y., and Zhu, Z. (2019). Analysis of the optimization landscapes for overcomplete representation learning. *arXiv preprint arXiv:1912.02427*.
- Royer, C. W., O’Neill, M., and Wright, S. J. (2020). A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization. *Mathematical Programming*, 180(1):451–488.
- Royer, C. W. and Wright, S. J. (2018). Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1448–1477.
- Shek, P. (2015). Local convergence of cubic regularisation methods for unconstrained non-convex optimisation problems. Part C Master dissertation (supervisor: C. Cartis), Mathematical Institute, University of Oxford.
- Steihaug, T. (1983). The Conjugate Gradient Method and Trust Regions in Large Scale Optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637.
- Sun, J., Qu, Q., and Wright, J. (2015). When are nonconvex problems not scary? *arXiv preprint arXiv:1510.06096*.

- Sun, J., Qu, Q., and Wright, J. (2017a). Complete Dictionary Recovery Over the Sphere I: Overview and the Geometric Picture. *IEEE Transactions on Information Theory*, 63(2):853–884.
- Sun, J., Qu, Q., and Wright, J. (2017b). Complete Dictionary Recovery Over the Sphere II: Recovery by Riemannian Trust-Region Method. *IEEE Transactions on Information Theory*, 63(2):885–914.
- Sun, J., Qu, Q., and Wright, J. (2018). A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198.
- Sun, Y., Flammarion, N., and Fazel, M. (2019). Escaping from saddle points on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Toint, P. (1981). Towards an Efficient Sparsity Exploiting Newton Method for Minimization. In Duff, I. S., editor, *Sparse Matrices and Their Uses*, pages 57–88. Academic press, London.
- Wright, J. and Ma, Y. (2022). *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press.
- Wright, S. J. and Recht, B. (2022). *Optimization for Data Analysis*. Cambridge University Press.
- Zhang, H. and Sra, S. (2016). First-order Methods for Geodesically Convex Optimization. In *Conference on Learning Theory*, pages 1617–1638.