Mathematical Programming
https://doi.org/10.1007/s10107-024-02156-2

FULL LENGTH PAPER

Series A

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

Florentin Goyens[1] · **Clément W. Royer**[1]

Received: 12 February 2024 / Accepted: 7 October 2024

Abstract

The difficulty of minimizing a nonconvex function is in part explained by the presence of saddle points. This slows down optimization algorithms and impacts worst-case complexity guarantees. However, many nonconvex problems of interest possess a favorable structure for optimization, in the sense that saddle points can be escaped efficiently by appropriate algorithms. This *strict saddle* property has been extensively used in data science to derive good properties for first-order algorithms, such as convergence to second-order critical points. However, the analysis and the design of second-order algorithms in the strict saddle setting have received significantly less attention. In this paper, we consider second-order trust-region methods for a class of strict saddle functions defined on Riemannian manifolds. These functions exhibit (geodesic) strong convexity around minimizers and negative curvature at saddle points. We first show that the standard trust-region method with exact subproblem minimization finds an approximate local minimizer in a number of iterations that depends logarithmically on the accuracy parameter, which significantly improves known results for general nonconvex optimization. We then propose a new inexact variant of the algorithm that explicitly leverages the strict saddle property to compute the most appropriate step at every iteration. Our bounds for the inexact variant also improve over the general nonconvex case, and illustrate the benefit of using strict saddle properties within optimization algorithms.

Keywords Riemannian optimization · Strict saddle function · Second-order method · Complexity guarantees

Mathematics Subject Classification 49M05 · 49M15 · 65K05 · 90C60

✉ Florentin Goyens
goyensflorentin@gmail.com

Clément W. Royer
clement.royer@lamsade.dauphine.fr

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

Published online: 08 November 2024

1 Introduction

We consider the optimization problem

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

where $\mathcal{M}$ is an n-dimensional Riemannian manifold, and $f : \mathcal{M} \to \mathbb{R}$ is twice continuously differentiable and nonconvex. A popular way to solve Problem (P) is to use Riemannian optimization techniques, that use differential geometry to generalize unconstrained optimization methods to the Riemannian setting [2, 4]. Theoretical guarantees for such methods have historically focused on the behavior close to minimizers (local convergence). These results usually rely on the objective function being convex (or strongly convex) around minimizers, thereby enabling the derivation of local convergence rates [2, 27].

Meanwhile, the past decade has seen a growing interest in global convergence results for nonconvex optimization, where one quantifies the rate of convergence towards a stationary point independently of the starting point [11]. These rates can be stated in the form of complexity results, which bound the number of iterations necessary to satisfy approximate first- or second-order necessary conditions for optimality. Second-order stationary points for Problem (P) have a zero Riemannian gradient and positive semidefinite Riemannian Hessian:

$$\operatorname{grad} f(x) = 0 \qquad \text{and} \qquad \lambda_{\min}(\operatorname{Hess} f(x)) \geq 0, \tag{1.1}$$

where $\lambda_{\min}(\cdot)$ is the smallest eigenvalue of a symmetric operator. Given positive tolerances $(\varepsilon_g, \varepsilon_H)$, complexity results bound the cost of satisfying an approximate version of (1.1), given by

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

In this paper, the structure of the problem allows to target approximate minimizers that satisfy

$$\|\operatorname{grad} f(x_k)\| \leq \varepsilon_g \quad \text{and} \quad \lambda_{\min}(\operatorname{Hess} f(x_k)) \geq \gamma, \tag{1.3}$$

for some problem parameter $\gamma > 0$.

In the unconstrained or Euclidean setting (i.e., when $\mathcal{M} = \mathbb{R}^n$), it is well established that classical second-order trust-region methods [12] reach an iterate satisfying (1.2) in at most $\mathcal{O}(\max(\varepsilon_g^{-2}\varepsilon_H^{-1}, \varepsilon_H^{-3}))$ iterations [9]. Although this complexity can be improved to $\mathcal{O}(\max(\varepsilon_g^{-2}, \varepsilon_H^{-3}))$ without changing the essence of the algorithm [14, 21], the resulting bound remains suboptimal among a large class of second-order methods [10]. Indeed, techniques such as cubic regularization enjoy a $\mathcal{O}(\max(\varepsilon_g^{-3/2}, \varepsilon_H^{-3}))$ complexity bound, that strictly improves over standard trust-region methods and is optimal among the class of second-order algorithms. Similar bounds were obtained for the Riemannian counterparts of trust-region methods [5] and cubic regularization [3]. Modifications of the trust-region scheme have been proposed to achieve the optimal complexity of cubic regularization [16, 17].

These worst-case results are pessimistic in nature and do not reflect the good behaviour of second-order methods on many practical problems. In an effort to reconcile theoretical guarantees with practical performances, it becomes necessary to leverage additional structure from the function f. Numerous problems of the form (P) have the property that the nonconvexity is *benign*, meaning that second-order critical points—Equation (1.1)—are global minimizers [35, 41]. Data analysis tasks with this property include eigenvalue problems, Burer-Monteiro factorizations of semidefinite programs [6, 25], phase retrieval [38], matrix completion and factorization [20, 23], dictionary learning [30, 36] and others. In this work, we consider a subset of problems with benign nonconvexity (see Definition 2.4), which focuses on problems with isolated minimizers.

Benign nonconvexity implies that the Hessian possesses a negative eigenvalue at every saddle point. This *strict saddle* property allows first- and second-order methods to provably avoid saddle points and converge towards minimizers. First-order methods escape strict saddle points almost surely [22], and complexity bounds can even be derived for randomized first-order techniques, in both the Euclidean and Riemannian setting [13, 39]. In addition, second-order methods, that leverage directions of negative curvature of the Hessian, escape strict saddle points by design, and are thus particularly suitable for strict saddle problems [41, Chapter 9].

Adaptations of complexity analysis to strict saddle problems have recently begun to appear in the literature. On one hand, complexity results were established for specific instances satisfying a strict saddle property, such as phase retrieval [38] or dictionary learning [37]. More recently, O'Neill and Wright [28] considered low-rank matrix optimization problems under a strict saddle property, and designed a line-search method that made explicit use of the strict saddle structure. In these works, the analysis is tailored to specific problems, and its generalization to a broader strict saddle setting is not straightforward.

On the other hand, general analyses based on dividing the feasible set into regions of interest yielded complexity bounds that improved over the general nonconvex setting, in the sense that the dependencies with respect to ε_g and ε_H were only logarithmic rather than polynomial [15, 29]. Carmon et al. [8] showed that an accelerated gradient technique tailored to nonconvex problems would enjoy improved complexity when applied to a function satisfying the strict saddle property. These general results apply to unconstrained strict saddle problems, and do not cover optimization problems on manifolds, a popular source of strict saddle problems [41].

Contributions and outline

In this work, we analyze a trust-region framework for minimizing strict saddle functions over Riemannian manifolds. The strict saddle problems we consider are strongly convex near minimizers, which leads to connections with Riemannian optimization of geodesically strongly convex functions. In particular, we leverage local convergence results for Newton's method in order to derive complexity results for our framework. We show that the standard trust-region method [1] with exact subproblem minimization applied to a strict saddle function benefits from improved complexity guarantees

compared to the general nonconvex setting. Indeed, our complexity bound possesses a $\mathcal{O}\left(\log\log(\varepsilon^{-1})\right)$ dependency in the optimality tolerances, thanks to the local quadratic convergence of the method, which improves over polynomial dependencies from the general case. We also derive similar results for an inexact version of our algorithm based on inexact solutions of the trust-region subproblem, that makes explicit use of the strict saddle structure. Our analysis builds on recent advances in the complexity of (Euclidean) trust-region methods by relying on iterative linear algebra routines. This yields complexity bounds in terms of iterations as well as Hessian-vector products.

To the best of our knowledge, we provide the first strict saddle analysis of a generic second-order trust-region method, and the first strict saddle analysis that applies to a generic manifold $\mathcal{M}$. All our results apply naturally to the unconstrained case $\mathcal{M} = \mathbb{R}^n$. Overall, our results advocate for further use of the strict saddle structure in the design and analysis of nonconvex optimization methods.

The rest of the paper is organized as follows. In Sect. 2, we describe the class of strict saddle functions on Riemannian manifolds that we investigate throughout the paper. This is prefaced by background material on Riemannian optimization and geodesic convexity. In Sect. 3, we analyze the global complexity of the Riemannian trust-region with exact subproblem minimization. This is a well-known algorithm designed for general nonconvex optimization, for which we show an improved complexity when applied to strict saddle functions. In Sect. 4, we propose a new Riemannian trust-region method with inexact subproblem minimization that uses landscape parameters to compute directions which are appropriate for the local landscape. The guarantees for the inexact algorithm account for the cost of solving the subproblem.

2 Strict saddle functions on Riemannian manifolds

In this section, we define a class of strict saddle functions on Riemannian manifolds. We first present background material on Riemannian optimization in Sect. 2.1, with a focus on retractions. We then discuss the notion of geodesic strong convexity in Sect. 2.2, which plays a role in our definition of strict saddle functions. This definition is provided along with several examples in Sect. 2.3.

2.1 Retractions and derivatives on Riemannian manifolds

Recall that problem (P) considers the minimization of a smooth function f over a Riemannian manifold $\mathcal{M}$. We cover the basic ideas that allow to build feasible algorithms for (P).

At every $x \in \mathcal{M}$, the linear approximation of the manifold $\mathcal{M}$ is called the *tangent space*, written $\mathrm{T}_x\mathcal{M}$. Each tangent space is equipped with an inner product $\langle \cdot, \cdot \rangle_x$, which defines the norm of a tangent vector as $\|v\|_x := \sqrt{\langle v, v \rangle_x}$ for $v \in \mathrm{T}_x\mathcal{M}$. (We often write $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ when the reference point is clear from context.) For smooth functions, the metric defines a *Riemannian gradient* and *Riemannian Hessian* of f at $x \in \mathcal{M}$, which we denote by $\operatorname{grad} f(x) \in \mathrm{T}_x\mathcal{M}$ and $\operatorname{Hess} f(x) \colon \mathrm{T}_x\mathcal{M} \to \mathrm{T}_x\mathcal{M}$,

respectively. By contrast, we use the symbols ∇ and ∇^2 for the gradient and Hessian of a function defined over a Euclidean space.

Riemannian optimization algorithms use tangent vectors to generate search directions. Following a tangent direction in a straight line may lead outside the manifold, which is undesirable. Therefore, we need a tool to travel on the manifold in a direction prescribed by a tangent vector. This can be done by following the geodesic associated with a tangent vector. On manifolds, geodesics are curves with zero acceleration that generalize the notion of straight line in Euclidean spaces. Formally, a geodesic is a smooth curve $c\colon I \to \mathcal{M}$ 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 [4, Chapter 5]. The exponential map travels along the manifold by following geodesics, but optimization algorithms commonly use first-order approximations of the exponential map, called retractions [2, §4.1]. A *retraction* at x is a map from the tangent space to the manifold, denoted by $R_x\colon \mathrm{T}_x\mathcal{M} \to \mathcal{M}$. For many manifolds of interest, practical and popular retractions are defined globally [2, Chapter 4]. However, the retraction at $x \in \mathcal{M}$ may only be defined locally, in a ball of radius $\varrho(x) > 0$ centered around 0_x in $\mathrm{T}_x\mathcal{M}$. In that case the size of the step at $x \in \mathcal{M}$ must be limited to $\varrho(x)$. We discuss this further in Sect. 3.1.

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 $\mathrm{T}_x\mathcal{M}$ is the function $\hat{f}_x\colon \mathrm{T}_x\mathcal{M} \to \mathbb{R}$ defined by

$$\hat{f}_x(s) := f \circ R_x(s) \text{ for all } s \in \mathrm{T}_x\mathcal{M}.$$

In particular, given $x \in \mathcal{M}$ and $s \in \mathrm{T}_x\mathcal{M}$, we consider the gradient of the pullback function $\nabla \hat{f}_x(s) \in \mathrm{T}_x\mathcal{M}$ as well as its Hessian $\nabla^2 \hat{f}_x(s) : \mathrm{T}_x\mathcal{M} \to \mathrm{T}_x\mathcal{M}$. Note the distinction between these derivatives and the Riemannian derivatives of f at $R_x(s)$, denoted by $\operatorname{grad} f(R_x(s))$ and $\operatorname{Hess} f(R_x(s))$. The identities $\hat{f}_x(0) = f(x)$ and $\nabla \hat{f}_x(0) = \operatorname{grad} f(x)$ hold by definition [4, Proposition 3.59] and additional assumptions on the retraction allow to relate the second-order derivatives. To benefit fully from second-order methods, we require that the retraction be a second-order approximation of geodesics.

A1 The retraction mapping is a *second-order retraction*: for any $x \in \mathcal{M}$ and $s \in \mathrm{T}_x\mathcal{M}$, the curve $c : t \in [0, 1] \to R_x(ts)$ has zero acceleration at $t = 0$, that is, $c''(0) = 0$.

If R_x is a second-order retraction, it holds that

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

i.e., the Hessian of the pullback function is the Riemannian Hessian of f [4, Proposition 5.45].

Remark 2.1 In this paper, we choose to use a general retraction over the more restrictive exponential map. This requires certain smoothness assumptions on the pullback function (see A3), but has the advantage of resembling the Euclidean setting. Using the exponential map typically leads to a different analysis that relies on parallel transport along geodesics, where the curvature of the manifold appears explicitly [13, 39, Section 4].

2.2 Geodesic convexity

We now provide the key definitions behind geodesic convexity, a concept that generalizes convexity in Euclidean spaces to Riemannian manifolds. Geodesically convex sets and functions are 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 $c\colon [0, 1] \to \mathcal{M}$ such that $c(0) = x,\ c(1) = y$ and $c(t)$ is in S for all $t \in [0, 1]$.

A function is geodesically convex on $S \subset \mathcal{M}$ if it is convex in the usual sense along all geodesics on S.

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

For smooth functions, geodesic strong convexity is determined by the eigenvalues of the Riemannian Hessian.

Proposition 2.1 (Theorem 11.23 in [4]) *A function $f\colon \mathcal{M} \to \mathbb{R}$ is geodesically γ-strongly convex on the set $S \subset \mathcal{M}$ if S is a geodesically convex set and $\lambda_{\min}(\mathrm{Hess} f(x)) \geq \gamma$ for every $x \in S$.*

Since we are interested in nonconvex problems, we consider functions that are geodesically strongly convex over a subset of the manifold (near minimizers). Functions with geodesic convexity over the entire manifold have also been studied, with most interesting applications arising on Hadamard manifolds [42].

2.3 Strict saddle property

We are now ready to define our problem class of interest, robust *strict saddle* functions on $\mathcal{M}$. The definition is based on [19, 35].

Definition 2.4 Let $f\colon \mathcal{M} \to \mathbb{R}$ be twice differentiable and let $\alpha, \beta, \gamma, \delta$ be positive constants. 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

$$
\begin{aligned}
\mathcal{R}_1 &= \{x \in \mathcal{M} : \|\mathrm{grad} f(x)\| \geq \alpha\} \\
\mathcal{R}_2 &= \{x \in \mathcal{M} : \lambda_{\min}(\mathrm{Hess} f(x)) \leq -\beta\} \\
\mathcal{R}_3 &= \big\{x \in \mathcal{M} :\ \text{there exists } x^* \in \mathcal{M}, \text{a local minimizer of } f \text{ such that } \mathrm{dist}(x, x^*) \leq \delta \text{ and} \\
&\qquad f \text{ is geodesically } \gamma\text{-strongly convex over the set } \{y \in \mathcal{M} : \mathrm{dist}(x^*, y) < 2\delta\}\big\}.
\end{aligned}
$$

Definition 2.4 has the following interpretation. If f is a strict saddle function on $\mathcal{M}$, then, at any $x \in \mathcal{M}$, either the norm of the Riemannian gradient is sufficiently large, the Riemannian Hessian has a sufficiently negative 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 exist in the literature, and the main differences appear in the definition of the region $\mathcal{R}_3$, where strong convexity is not always required [24, 28]. Our definition excludes non-isolated minimizers, where strong convexity cannot hold. Nevertheless, non-isolated minimizers can arise due to rotational symmetries in the problem, such as for phase retrieval and matrix factorization [41]. We note that recent work has focused on reformulating such problems on a quotient set induced by the symmetry, leading to problems where minimizers are isolated [25].

We conclude this section with two simple examples of strict saddle functions in the sense of Definition 2.4. Our first example is a strongly convex function over $\mathbb{R}^n$.

Example 2.1 Let $f : \mathbb{R}^n \to \mathbb{R}$ be a (geodesically) γ-strongly convex function with global minimizer x^*. For any $\alpha > 0$, f is $(\alpha, 1, \gamma, \frac{2\alpha}{\gamma})$-strict saddle. The region $\mathcal{R}_2$ is empty, and for any $x \notin \mathcal{R}_1$ (i.e., $\|\nabla f(x)\| < \alpha$), we show that $x \in \mathcal{R}_3$. Strong convexity gives

$$\tfrac{\gamma}{2}\|x-x^*\|^2 \leq f(x)-f(x^*) \leq -\nabla f(x)^{\mathrm{T}}(x-x^*) \leq \|\nabla f(x)\|\|x-x^*\| \leq \alpha\|x-x^*\|,$$

hence $\|x-x^*\| \leq \frac{2\alpha}{\gamma} =: \delta$. Clearly, f is γ-strongly convex on $\{x : \|x-x^*\| < 2\delta\} \subset \mathbb{R}^n$.

Our second example, previously introduced in [35], illustrates the interest of the region $\mathcal{R}_2$ in the presence of nonconvexity.

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

To end this section, we state our key assumption about Problem (P).

A2 There exist positive constants $(\alpha, \beta, \gamma, \delta)$ such that the function f is $(\alpha, \beta, \gamma, \delta)$-strict saddle on the manifold $\mathcal{M}$ and $\mathcal{R}_3$ is a compact subset of $\mathcal{M}$.

The compactness assumption on $\mathcal{R}_3$ merely prevents the function from having infinitely many minimizers on $\mathcal{M}$, and is made to simplify the presentation. It is possible to extend our analysis to an unbounded region $\mathcal{R}_3$, but this lengthens the argument considerably. Note that Assumption A2 holds for both examples above. The boundedness assumption also allows to control the distance between iterates of our algorithms.

Lemma 2.2 (Lemma 6.32 in [4]) *Under A2, there exists positive constants ν_S, κ_S such that for all $x \in \mathcal{R}_3$ and $s \in \mathrm{T}_x\mathcal{M}$, if $\|s\|_x \leq \nu_S$, then* $\mathrm{dist}(x, R_x(s)) \leq \kappa_S \|s\|_x$, *where* $\mathrm{dist}(\cdot, \cdot)$ *is the Riemannian distance on $\mathcal{M}$.*

3 Riemannian trust-region method with exact subproblem minimization

In this section we analyze the classical Riemannian trust-region algorithm (RTR) with exact subproblem minimization. Our goal is to leverage the strict saddle property to obtain better complexity bounds than those existing for general nonconvex functions [5]. The algorithm is perfectly standard, and does not require any knowledge of the strict saddle parameters of the function. Our improved complexity bounds rely on a good understanding of the local convergence of the algorithm. In particular, our analysis borrows from recent results on Newton-type methods for the Euclidean setting [16]. Note that using the exponential map instead of a generic retraction would simplify the analysis, as noticed for Riemannian cubic regularization [3].

Section 3.1 describes the exact trust-region algorithm, along with key assumptions. Standard decrease lemmas are provided in Sect. 3.2. A local convergence analysis of the algorithm in the region of geodesic strong convexity is provided in Sect. 3.3. This analysis allows to derive our global convergence result, that is established in Sect. 3.4.

3.1 Algorithm and assumptions

The Riemannian trust-region method with exact subproblem minimization is described in Algorithm 1. At every iteration, a step s_k is computed by minimizing a quadratic model of (the pullback of) the function over the tangent space corresponding to the current iterate x_k. In this section, we assume that the subproblem (3.1) is solved exactly using standard approaches [1, 26] (the inexact case is addressed in Sect. 4). The algorithm computes the step s_k, then evaluates f at the point $R_{x_k}(s_k) \in \mathcal{M}$ to measure the change in function value. If the function decrease is at least a fraction of the model decrease, the iteration is successful, and the candidate point $R_{x_k}(s_k)$ becomes the new iterate. The trust-region radius is either unchanged (successful iteration) or can be increased (very successful iteration). If the iteration is unsuccessful, the algorithm remains at the current iterate and the trust-region radius is decreased.

The model is a second-order Taylor expansion of the pullback function $\hat{f}_{x_k}$, namely

$$m_k \colon \mathrm{T}_x\mathcal{M} \to \mathbb{R} \colon s \mapsto m_k(s) = f(x_k) + \langle s, g_k \rangle + \frac{1}{2} \langle s, H_k s \rangle \,, \tag{3.2}$$

where $g_k = \nabla \hat{f}_{x_k}(0) = \mathrm{grad} f(x_k)$ and $H_k = \nabla^2 \hat{f}_{x_k}(0)$, which gives a second-order accurate model. Although we do not consider it here, we belive that our analysis can be extended to approximate second-order accurate models, under the condition that H_k is a suitable approximation of $\nabla^2 \hat{f}_k(0)$ [2, Eq. (7.36)]. When the retraction is second-order (A1), we have $H_k = \mathrm{Hess} f(x_k)$ by (2.1).

Algorithm 1 RTR with exact subproblem minimization

Inputs: 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 $0 < \tau_1 < 1 < \tau_2$.
for $k = 1, 2, \ldots$ **do**
  Compute s_k as a solution to the trust-region subproblem

$$s_k \in \operatorname*{arg\,min}_{s \in \mathrm{T}_{x_k}\mathcal{M}} m_k(s) \text{ subject to } \|s\| \leq \Delta_k, \tag{3.1}$$

  where m_k is the model defined by (3.2).

  Compute $\rho_k = \dfrac{f(x_k) - f\left(R_{x_k}(s_k)\right)}{m_k(0) - m_k(s_k)}$ and set $x_{k+1} = \begin{cases} R_k(s_k) & \text{if } \rho_k \geq \eta_1 \\ x_k & \text{otherwise.} \end{cases}$

  Set $\Delta_{k+1} = \begin{cases} \min\left(\tau_2 \Delta_k, \bar{\Delta}\right) & \text{if } \rho_k > \eta_2 \quad \text{[very successful]} \\ \Delta_k & \text{if } \eta_2 \geq \rho_k \geq \eta_1 \quad \text{[successful]} \\ \tau_1 \Delta_k & \text{otherwise.} \quad \text{[unsuccessful]} \end{cases}$
end for

In order to derive complexity results for Algorithm 1, we make a standard Lipschitz-type assumption on the Hessian of the pullback [5]. Recall that if the retraction at x_k is defined locally, we write $\varrho(x_k) > 0$ for the radius of the ball centered around 0_{x_k} in $\mathrm{T}_{x_k}\mathcal{M}$ in which it is defined.

A3 There exists $L_H > 0$ such that for all iterates x_k generated by Algorithm 1, the pullback $\hat{f}_k = f \circ R_{x_k}$ satisfies

$$f(R_{x_k}(s)) \leq f(x_k) + \langle s, \operatorname{grad} f(x_k) \rangle + \frac{1}{2}\left\langle s, \nabla^2 \hat{f}_k(0)[s] \right\rangle + \frac{L_H}{6}\|s\|^3. \tag{3.3}$$

for all $s \in \mathrm{T}_{x_k}\mathcal{M}$ such that $\|s\| \leq \varrho(x_k)$. We further assume $\Delta_k \leq \varrho(x_k)$, so that the property (3.3) holds in the entire trust region produced by Algorithm 1.

A simple strategy to ensure $\varrho(x_k) \geq \Delta_k$ in the assumption above is to set $\bar{\Delta}$ below $\inf_{x \in \mathcal{M}:\, f(x) \leq f(x_0)} \varrho(x)$, which is positive if the injectivity radius of the manifold is positive [5, Remark 2.2]. Throughout we work implicitly under the assumption that this is satisfied. We additionally make the following assumption on the Hessian operators considered throughout the algorithm.

A4 There exists $\kappa_H > 0$ such that for all iterates x_k generated by Algorithm 1, we have

$$\|H_k\| := \sup_{\substack{s \in \mathrm{T}_{x_k}\mathcal{M} \\ \|s\| \leq 1}} |\langle s, H_k(s) \rangle| \leq \kappa_H. \tag{3.4}$$

3.2 Preliminary lemmas

In this section, using standard arguments from the theory of trust-region methods, we bound the model decrease in the regions $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ defined by the strict saddle property. We also provide a lower bound on the trust-region radius.

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

Lemma 3.1 *Under A2 and A4, consider the kth iterate of Algorithm 1 and suppose that $x_k \in \mathcal{R}_1$. Then,*

$$m_k(0) - m_k(s_k) \geq \frac{1}{2} \min\left(\Delta_k, \frac{\alpha}{\kappa_H}\right) \alpha. \tag{3.5}$$

Proof Define s_k^C as the Cauchy point associated with the trust-region subproblem (3.1), i.e. $s_k^C = -t g_k$ with $t = \arg\min_{t \geq 0, \|t\, g_k\| \leq \Delta_k} m_k(-t\, g_k)$. A straightforward application of Boumal [4, Lemma 6.15] gives

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

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

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

Lemma 3.2 *Under A1 and A2, consider the kth iterate of Algorithm 1 and suppose that $x_k \in \mathcal{R}_2$. Then,*

$$m_k(0) - m_k(s_k) \geq \frac{1}{2} \beta \Delta_k^2. \tag{3.6}$$

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

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

The vector s_k^E—called an eigenstep—exists because $x_k \in \mathcal{R}_2$. By Boumal [4, Lemma 6.16], it satisfies

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

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

Our last decrease lemma is based on the strong convexity constant, and proceeds similarly to the previous two results.

Lemma 3.3 *Under A1 and A2, consider the kth iterate of Algorithm 1 and suppose that $x_k \in \mathcal{R}_3$. Then, the step s_k is uniquely defined, and satisfies*

$$m_k(0) - m_k(s_k) \geq \frac{1}{2} \gamma \ \|s_k\|^2 . \tag{3.7}$$

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 [2, Proposition 7.3.1]:

$$(H_k + \lambda_k \,\mathrm{Id})\, s_k = -g_k \tag{3.8}$$

$$\langle s, (H_k + \lambda_k \,\mathrm{Id})\,[s]\rangle \geq 0 \quad \forall s \in \mathrm{T}_{x_k}\mathcal{M} \tag{3.9}$$

$$\|s_k\| \leq \Delta_k \tag{3.10}$$

$$\lambda_k(\Delta_k - \|s_k\|) = 0. \tag{3.11}$$

Moreover, if the inequality in (3.9) is strict for nonzero s, then the solution is unique. To establish a decrease guarantee for s_k, we combine (3.8) and $\lambda_k \geq 0$ to obtain

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

The last line follows from $H_k = \mathrm{Hess}\, f(x_k)$ and $x_k \in \mathcal{R}_3$, i.e., $\langle s, H_k s\rangle \geq \gamma \|s\|^2$ for any $s \in \mathrm{T}_{x_k}\mathcal{M}$. This also implies that $H_k + \lambda_k \,\mathrm{Id}$ is positive definite, hence s_k is uniquely defined. □

The three lemmas above, together with the Lipschitz-type assumptions on the pullback, yield a lower bound on the trust-region radius.

Lemma 3.4 *Let A1, A2, A3 and A4 hold. Then, for any iteration of index k, the trust-region radius Δ_k in Algorithm 1 satisfies*

$$\Delta_k \geq \Delta_{\min} := c_\Delta \min\left(\Delta_0, \alpha^{1/2}, \alpha^{2/3}, \beta, \gamma\right), \tag{3.12}$$

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

Proof We begin by showing that if the trust-region radius drops below a certain threshold, then the iteration must be successful. We consider the quantity

$$1 - \rho_k = 1 - \frac{f(x_k) - f(R_{x_k}(s_k))}{m_k(0) - m_k(s_k)} = \frac{f(R_{x_k}(s_k)) - m_k(s_k)}{m_k(0) - m_k(s_k)} \tag{3.13}$$

for the three regions defined by the strict saddle property. First note that for any $x_k \in \mathcal{M}$, a second-order retraction (A1) and Lipschitz continuity of the Hessian (A3)

give the following bound on the numerator of (3.13):

$$f(R_{x_k}(s_k)) - m_k(s_k) = f(R_{x_k}(s_k)) - f(x_k) - \langle g_k, s_k \rangle - \frac{1}{2}\langle s_k, H_k s_k \rangle$$
$$\leq \frac{L_H}{6}\|s_k\|^3 \leq \frac{L_H}{6}\Delta_k^3. \tag{3.14}$$

For $x_k \in \mathcal{R}_1$, the denominator of (3.13) satisfies (3.5). It follows that

$$1 - \rho_k \leq \frac{L_H \Delta_k^3}{3\min\left(\Delta_k \alpha, \alpha^2/\kappa_H\right)} \leq \frac{L_H}{3}\max\left(\frac{\Delta_k^2}{\alpha}, \frac{\kappa_H \Delta_k^3}{\alpha^2}\right).$$

As a result, if $x_k \in \mathcal{R}_1$ and

$$\Delta_k \leq \min\left(\sqrt{\frac{3(1-\eta_1)}{L_H}}\alpha^{1/2}, \sqrt[3]{\frac{3(1-\eta_1)}{\kappa_H L_H}}\alpha^{2/3}\right),$$

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

If $x_k \in \mathcal{R}_2$, the denominator of (3.13) satisfies (3.6), which we combine with (3.14) to give

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

Thus, if $x_k \in \mathcal{R}_2$ and $\Delta_k \leq 3(1-\eta_1)\beta/L_H$, we have $\rho_k \geq \eta_1$ and iteration k is successful.

Finally, for $x_k \in \mathcal{R}_3$, the upper bound (3.14) together with the model decrease (3.7) gives

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

As a result, if $x_k \in \mathcal{R}_3$ and $\Delta_k \leq 3(1-\eta_1)\gamma/L_H$, then $\rho_k \geq \eta_1$ and iteration k is successful.

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

$$\Delta_k \leq \min\left(\sqrt{\frac{3(1-\eta_1)}{L_H}}\alpha^{1/2}, \sqrt[3]{\frac{3(1-\eta_1)}{\kappa_H L_H}}\alpha^{2/3}, \frac{3(1-\eta_1)}{L_H}\beta, \frac{3(1-\eta_1)}{L_H}\gamma\right),$$

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 $k \geq 0$:

$$\Delta_k \geq \min\left(\Delta_0, \tau_1\sqrt{\frac{3(1-\eta_1)}{L_H}}\alpha^{1/2}, \tau_1\sqrt[3]{\frac{3(1-\eta_1)}{\kappa_H L_H}}\alpha^{2/3}, \tau_1\frac{3(1-\eta_1)}{L_H}\beta, \tau_1\frac{3(1-\eta_1)}{L_H}\gamma\right)$$
$$\geq c_\Delta \min\left(\Delta_0, \alpha^{1/2}, \alpha^{2/3}, \beta, \gamma\right). \qquad \square$$

For $x_k \in \mathcal{R}_1 \cup \mathcal{R}_2$, it is straightforward to combine Lemma 3.4 with the results of Lemmas 3.1 and 3.2 to guarantee a model decrease that is independent of k. For $x_k \in \mathcal{R}_3$, deriving a uniform lower bound on the decrease based on Lemma 3.3 is 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 Algorithm 1 in the region of strong convexity $\mathcal{R}_3$. The global subproblem minimizer is a regularized Newton step—Equation (3.8)—and thus our approach mimics the study of Newton's method applied to strongly convex functions [7]. For sufficiently large gradients, we provide a lower bound on the decrease achieved by the step; and for small gradients, we show that a local convergence phase begins, during which the iterates converge quadratically towards a local minimizer of (P). To establish local convergence, we quantify how small the gradient norm needs to be so that the following occurs: the full Newton step is the solution of the subproblem, it is successful, it produces a new iterate that is also in $\mathcal{R}_3$, and the sequence of gradient norms converges quadratically towards zero.

Before stating those results, we establish several consequences of the boundedness assumption on $\mathcal{R}_3$ (A2) that are helpful in analyzing (regularized) Newton steps, starting with a Lipschitz-type inequality on the gradient of the pullback.

Lemma 3.5 *Under A2, there exists $\hat{L}_H > 0$ such that for all iterates $x_k \in \mathcal{R}_3$ produced by Algorithm 1, we have*

$$\left\| \nabla \hat{f}_k(s_k) - \nabla \hat{f}_k(0) - \nabla^2 \hat{f}_k(0)[s_k] \right\| \leq \frac{\hat{L}_H}{2} \|s_k\|^2 . \tag{3.15}$$

Proof The constant $\hat{L}_H$ exists by continuity of the derivatives over $\mathcal{R}_3$ (a compact set) and boundedness of the steps [4, Lemma 10.57]. The compactness of $\mathcal{R}_3$ comes from A2, while the boundedness of the steps s_k follows from $\|s_k\| \leq \Delta_k \leq \bar{\Delta}$. □

The next result describes a non-singularity condition for the differential of the retraction on small steps. This allows to relate the gradient of the pullback and the gradient at the next iterate. This property was introduced in [1], and further analyzed in the context of cubic regularization of Newton's method [3].

Lemma 3.6 *Let $\kappa_R > 1$, under A2 there exists $\nu_R > 0$ such that for any $x_k \in \mathcal{R}_3$ and $s \in \mathrm{T}_{x_k}\mathcal{M}$ with $\|s\| \leq \nu_R$, we have*

$$\|\mathrm{grad}\, f(R_x(s))\|_{x_{k+1}} \leq \kappa_R \left\| \nabla \hat{f}(s) \right\|_{x_k} . \tag{3.16}$$

Proof We apply [3, Theorem 7] to $\mathcal{R}_3$ as a non-empty compact subset of $\mathcal{M}$. This ensures that for any $\kappa_R > 1$, there exists a constant $\nu_R > 0$ such that, at each $x_k \in \mathcal{R}_3$,

$$\|s\| \leq \nu_R \quad \Leftrightarrow \quad \sigma_{\min}(\mathrm{D}R_{x_k}(s)) \geq \frac{1}{\kappa_R},$$

where $\mathrm{D}R_{x_k}$ denotes the differential of R_{x_k}. The desired conclusion follows by combining this result with [3, Equation 22], which states that

$$\left\|\nabla \hat{f}_{x_k}(s)\right\| \geq \sigma_{\min}\left(\mathrm{D}R_{x_k}(s)\right)\left\|\operatorname{grad} f(R_{x_k}(s))\right\|. \qquad \square$$

Combining the previous two lemmas, we bound the change in gradient norm for Newton steps.

Lemma 3.7 *Under A1 and A2, let $x_k \in \mathcal{R}_3$ be an iterate produced by Algorithm 1 such that s_k is the 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*

$$\|\operatorname{grad} f(x_{k+1})\|_{x_{k+1}} \leq \kappa_R \frac{\hat{L}_H}{2} \|s_k\|_{x_k}^2 \tag{3.17}$$

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

Proof By assumption, $x_{k+1} = R_{x_k}(s_k)$. Using successively Lemma 3.6, A1 and Lemma 3.5, we obtain

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

Remark 3.1 Using the exponential map rather than a general retraction significantly simplifies the analysis above. Indeed, with the exponential map, Lemma 3.5 can be replaced by

$$\left\|P_s^{-1} \operatorname{grad} f(\operatorname{Exp}_x(s)) - \operatorname{grad} f(x) - \operatorname{Hess} f(x)[s]\right\| \leq \frac{L_H}{2}\|s\|^2 \tag{3.18}$$

where P_s^{-1} is the parallel transport from $\mathrm{T}_{\operatorname{Exp}_x(s)}\mathcal{M}$ to $\mathrm{T}_x\mathcal{M}$, and L_H is the Lipschitz constant of $\operatorname{Hess} f$ from (3.3), see [4, Corollary 10.56]. As a result, the proof of Lemma 3.7 is also simplified, and no longer requires the decomposition in short and long steps induced by Lemma 3.6.

We are equipped to prove a decrease guarantee for Newton steps.

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

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

Proof By combining (3.17) with (3.7), we obtain

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

□

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. We begin with a bound on the norm of the subproblem minimizer in $\mathcal{R}_3$.

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

$$\|s_k\| \leq \frac{\|g_k\|}{\gamma}. \tag{3.19}$$

Proof The result holds trivially if $\|s_k\| = 0$. Otherwise, using the definition of $\mathcal{R}_3$ together with $\lambda_k \geq 0$ and (3.8), we get

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

and division by $\|s_k\|$ gives (3.19). □

Lemma 3.9 is an elementary identity that we use throughout. We use it in the next proposition to show that iterations in $\mathcal{R}_3$ with a small enough gradient are successful.

Proposition 3.10 *Under A1 and A3, suppose that Algorithm 1 generates $x_k \in \mathcal{R}_3$ such that*

$$\|\operatorname{grad} f(x_k)\| < \frac{3(1-\eta_1)\gamma^2}{L_H}. \tag{3.20}$$

Then, the kth iteration is successful.

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

$$f(R_k(s_k)) - m_k(s_k) + (1-\eta_1)(m_k(s_k) - m_k(0)) \leq 0. \tag{3.21}$$

The proof then consists in finding an upper bound for the left-hand side that is negative. From Lemma 3.3, we have that

$$(1-\eta_1)\,(m_k(s_k) - m_k(0)) \leq (1-\eta_1)(-\frac{\gamma}{2} \|s_k\|^2).$$

Combining this property with (3.14) and (3.19) gives

$$\begin{aligned} f(R_k(s_k)) - m_k(s_k) + (1-\eta_1)(m_k(s_k) - m_k(0)) &\leq \frac{L_H}{6}\|s_k\|^3 - \frac{\gamma}{2}(1-\eta_1)\|s_k\|^2 \\ &= \|s_k\|^2\left(\frac{L_H}{6}\|s_k\| - \frac{\gamma}{2}(1-\eta_1)\right) \\ &\leq \|s_k\|^2\left(\frac{L_H}{6}\frac{\|g_k\|}{\gamma} - \frac{\gamma}{2}(1-\eta_1)\right). \end{aligned}$$

The right-hand side is negative by (3.20), from which we conclude that (3.21) holds, and the iteration is successful. □

We now show that if the gradient norm is small enough, the Newton step decreases the gradient norm.

Proposition 3.11 *Under A1 and A2, let $x_k \in \mathcal{R}_3$ be an iterate produced by Algorithm 1 such that s_k is the Newton step, i.e., $H_k s_k = -g_k$, with $\|s_k\| \leq \nu_R$ where ν_R is the constant from Lemma 3.6. If*

$$\|\operatorname{grad} f(x_k)\| < \frac{2\gamma^2}{\kappa_R \hat{L}_H}, \tag{3.22}$$

and the iteration is successful, then $\|\operatorname{grad} f(x_{k+1})\|_{x_{k+1}} < \|\operatorname{grad} f(x_k)\|_{x_k}$.

Proof Combining (3.17), (3.19) and (3.22) gives

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

In order to derive a local convergence result, we show that, if $x_k \in \mathcal{R}_3$ and the gradient norm is small enough, the Newton step remains in the neighborhood of the same minimizer.

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

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

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

Proof We first show that $\operatorname{dist}(x_{k+1}, x^*) < 2\delta$. Using (3.19) and (3.23), we have that

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

It follows from Lemma 2.2 that

$$\operatorname{dist}(x_{k+1}, x_k) \leq \kappa_S \|s_k\| \leq \kappa_S \frac{\|g_k\|}{\gamma} \leq \frac{\kappa_S}{\gamma} \frac{\gamma\delta}{2\kappa_S} = \frac{\delta}{2},$$

where the last inequality is due to (3.23). As a result,

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

By definition of x^*, we know that f is geodesically γ-strongly convex over $S \subseteq \mathcal{M}$, a subset of $\mathcal{M}$ that includes x^* and x_{k+1}. Consider a geodesic $c\colon [0, 1] \to S$ contained in S with $c(0) = x_{k+1}$ and $c(1) = x^*$, such that $\operatorname{dist}(x_{k+1}, x^*) \leq L(c)$ where $L(c) = \left\|c'(0)\right\|_{c(0)}$ is the length of the geodesic path. From [4, Theorem 11.21], we have

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

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

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

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

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

□

We now characterize the local convergence of Algorithm 1. The quadratic convergence rate stems from the following proposition.

Proposition 3.13 *Under A1 and A2, 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 kth iteration is successful. Then,*

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

Proof Using Lemma 3.7 and Lemma 3.9 gives

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

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

Proposition 3.13 guarantees that the gradient norm decreases quadratically along Newton steps, provided that the gradient norm is small enough. In Proposition 3.14, we show that the local phase of Algorithm 1 begins once the trust-region method generates a point in $\mathcal{R}_3$ with a small enough gradient.

Proposition 3.14 (Local convergence of Algorithm 1) *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 $\operatorname{dist}(x_k, x^*) \leq \delta$ and f is geodesically γ-strongly convex on $\{y \in \mathcal{M} : \operatorname{dist}(y, x^*) < 2\delta\}$. Finally, suppose that*

$$\|\operatorname{grad} f(x_k)\| < \min\left(c_Q \min\left(\gamma, \gamma^2, \gamma\delta\right), \gamma\Delta_k\right) \tag{3.24}$$

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

Proof Lemma 3.9 and (3.24) give $\|s_k\| \leq \|\operatorname{grad} f(x_k)\|/\gamma < \Delta_k$, and thus s_k is the Newton step. In addition, the condition (3.24) also implies (3.20), thus the kth iteration is successful by Proposition 3.10. Similarly, (3.24) implies (3.22), (3.23), and $\|s_k\| \leq \nu_R$, which yields $\|g_{k+1}\| < \|g_k\|$ (Proposition 3.11) and $\operatorname{dist}(x_{k+1}, x^*) < \delta$ (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 repeated application of Proposition 3.13, as (3.24) implies $\|\operatorname{grad} f(x_k)\| < \gamma^2/(\kappa_R \hat{L}_H)$. Indeed, for any index $l \geq k$, we have

$$\frac{\hat{L}_H \kappa_R}{2\gamma^2}\|\operatorname{grad} f(x_l)\| \leq \left(\frac{\hat{L}_H \kappa_R}{2\gamma^2}\|\operatorname{grad} f(x_k)\|\right)^{2^{l-k}} \leq \left(\frac{1}{2}\right)^{2^{l-k}}, \tag{3.25}$$

which characterizes quadratic convergence. □

We emphasize that the local convergence property is an integral part of our *global* convergence analysis. Deriving global rates of convergence (i.e., complexity results) is the subject of the next section.

3.4 Complexity bounds

In this section, we combine the results from Sects. 3.2 and 3.3 to obtain complexity bounds. More precisely, we seek a bound on the number of iterations Algorithm 1 requires to reach an iterate $x_K \in \mathcal{M}$ that is an $(\varepsilon_g, \varepsilon_H)$-second-order critical point—defined in Equation (1.2).

Following Sect. 3.3, we can bound the number of iterations in the local phase necessary to satisfy (1.2). 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 (1.3) in at most*

$$N^{\text{local}}(\varepsilon_g) := \max\left[0, \log_2 \log_2 \left(\frac{2\gamma^2}{\kappa_R \hat{L}_H \varepsilon_g}\right)\right] \tag{3.26}$$

iterations following iteration k.

Proof If $2\gamma^2/(\kappa_R \hat{L}_H) < \varepsilon_g$, then (3.24) implies that x_k satisfies (1.3), as $\|g_k\| \leq \gamma^2/(\kappa_R \hat{L}_H) < \varepsilon_g$. Otherwise if $2\gamma^2/(\kappa_R \hat{L}_H) \geq \varepsilon_g$, for any $l \geq k$, it follows from Equation (3.25) that $\|\text{grad}\, f(x_l)\| \geq \varepsilon_g$ implies

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

Our main complexity result comes under the assumption that f is lower bounded on $\mathcal{M}$.

A5 There exists $f^* > -\infty$ such that $f(x) \geq f^*$ for all $x \in \mathcal{M}$.

We first give 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–A5 hold. For $\varepsilon_g > 0$, Algorithm 1 produces an iterate satisfying (1.3) in at most*

$$\frac{C}{\min\left(\underline{\alpha}^2, \underline{\alpha}^{4/3}\underline{\beta}, \underline{\alpha}^{4/3}\underline{\gamma}, \underline{\alpha}^{2/3}\underline{\gamma}^2, \underline{\beta}^3, \underline{\beta}^2\underline{\gamma}, \underline{\beta}\underline{\gamma}^2, \underline{\gamma}^3, \underline{\gamma}^2\underline{\delta}\right)} + 1 + N^{\text{local}}(\varepsilon_g)$$

successful iterations, where $N^{\text{local}}(\varepsilon_g)$ is defined in (3.26), 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 (1.3) 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 up to index K. 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.

Let $k \in \mathcal{S}_1$, using Lemma 3.1 and Lemma 3.4, we obtain

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

For $k \in \mathcal{S}_2$, combining Lemma 3.2 with Lemma 3.4 gives

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

We partition $\mathcal{S}_3$ into iterations with long steps, short steps on the boundary of the trust region, and short steps inside the trust region: $\mathcal{S}_3 := \mathcal{S}_3^l \cup \mathcal{S}_3^{s,b} \cup \mathcal{S}_3^{s,i}$, where

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

where $\nu_R > 0$ is defined in Lemma 3.6.

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

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

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

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

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

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

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

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

Finally, if $k \in \mathcal{S}_3^{s,i,s}$, either $x_{k+1} \in \mathcal{S}_3$ and the local convergence phase begins at x_{k+1} according to Proposition 3.14. Thus Theorem 3.15 ensures that the local convergence produces an iterate that satisfies (1.3) in a number of iterations given by (3.26). Otherwise, we must have $x_{k+1} \in \mathcal{S}_1 \cup \mathcal{S}_2$, and as a result we have

$$\left|\mathcal{S}_3^{s,i,s}\right| \leq |\mathcal{S}_1| + |\mathcal{S}_2| + 1 + N^{\text{local}}(\varepsilon_g). \tag{3.32}$$

It thus suffices to bound the cardinality of $\mathcal{S}_1$ and $\mathcal{S}_2$ to bound $\left|\mathcal{S}_3^{s,i,s}\right|$.

Thanks to A5, we have

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

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

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

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

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

Using that $\Delta_{\min} = c_\Delta \min\big(\Delta_0, \alpha^{1/2}, \alpha^{2/3}, \beta, \gamma\big) \geq c_\Delta \min(\Delta_0, 1) \min\Big(\underline{\alpha}^{2/3}, \underline{\beta}, \underline{\gamma}\Big)$ yields the following upper bounds

$$\begin{aligned} |\mathcal{S}_1| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \max\Big(\kappa_H, c_\Delta^{-1} \Delta_0^{-1}, c_\Delta^{-1}\Big) \max\Big(\underline{\alpha}^{-2}, \underline{\alpha}^{-1} \underline{\beta}^{-1}, \underline{\alpha}^{-1} \underline{\gamma}^{-1}\Big) \\ |\mathcal{S}_2| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \max\Big(c_\Delta^{-2} \Delta_0^{-2}, c_\Delta^{-2}\Big) \max\Big(\underline{\alpha}^{-4/3} \underline{\beta}^{-1}, \underline{\beta}^{-3}, \underline{\beta}^{-1} \underline{\gamma}^{-2}\Big) \\ |\mathcal{S}_3^l| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \nu_R^{-2} \underline{\gamma}^{-1} \\ |\mathcal{S}_3^{s,b}| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \max\Big(c_\Delta^{-2} \Delta_0^{-2}, c_\Delta^{-2}\Big) \max\Big(\underline{\alpha}^{-4/3} \underline{\gamma}^{-1}, \underline{\beta}^{-2} \underline{\gamma}^{-1}, \underline{\gamma}^{-3}\Big) \\ |\mathcal{S}_3^{s,i,l}| &\leq \frac{\kappa_R \hat{L}_H (f(x_0) - f^*)}{\eta_1} \max\Big(c_Q^{-1}, c_\Delta^{-1} \Delta_0^{-1}, c_\Delta^{-1}\Big) \end{aligned}$$

$$\max\left(\underline{\alpha}^{-2/3}\underline{\gamma}^{-2}, \underline{\beta}^{-1}\underline{\gamma}^{-2}, \underline{\gamma}^{-3}, \underline{\gamma}^{-2}\delta^{-1}\right).$$

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

$$\begin{aligned}|\mathcal{S}| &= |\mathcal{S}_1| + |\mathcal{S}_2| + |\mathcal{S}_3^l| + |\mathcal{S}_3^{s,b}| + |\mathcal{S}_3^{s,i,l}| + |\mathcal{S}_3^{s,i,s}| \\ &\leq 2|\mathcal{S}_1| + 2|\mathcal{S}_2| + |\mathcal{S}_3^l| + |\mathcal{S}_3^{s,b}| + |\mathcal{S}_3^{s,i,l}| + 1 + N^{\text{local}}(\varepsilon_g) \\ &\leq C\min\left(\underline{\alpha}^2, \underline{\alpha}^{4/3}\underline{\beta}, \underline{\alpha}^{4/3}\underline{\gamma}, \underline{\alpha}^{2/3}\underline{\gamma}^2, \underline{\beta}^3, \underline{\beta}^2\underline{\gamma}, \underline{\beta\gamma}^2, \underline{\gamma}^3, \underline{\gamma}^2\underline{\delta}\right)^{-1} + 1 + N^{\text{local}}(\varepsilon_g),\end{aligned}$$

where

$$C = \frac{(f(x_0) - f^*)}{\eta_1}\Big[4\max\left(\kappa_H, c_\Delta^{-1}\Delta_0^{-1}, c_\Delta^{-1}\right) + 6\max\left(c_\Delta^{-2}\Delta_0^{-2}, c_\Delta^{-2}\right) + 2\nu_R^{-2} + \kappa_R\hat{L}_H\max\left(c_Q^{-1}, c_\Delta^{-1}\Delta_0^{-1}, c_\Delta^{-1}\right)\Big].$$

□

Theorem 3.16 bounds the number of successful iterations required to satisfy (1.3), which corresponds to the number of gradient and Hessian evaluations. To account for the total number of iterations—the number of function evaluations, 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 $\mathcal{S}$ denote the set of successful iterations of index $k \leq K$. Then,*

$$\begin{aligned}|\mathcal{S}| \geq\ & \frac{\log_{\tau_2}(1/\tau_1)}{1 + \log_{\tau_2}(1/\tau_1)}(K+1) \\ & - \frac{1}{1 + \log_{\tau_2}(1/\tau_1)}\max\left[0, \log_{\tau_2}\left(\frac{1}{c_\Delta}\right), \log_{\tau_2}\left(\frac{\Delta_0}{c_\Delta\alpha^{\frac{1}{2}}}\right), \log_{\tau_2}\left(\frac{\Delta_0}{c_\Delta\alpha^{\frac{2}{3}}}\right),\right. \\ & \log_{\tau_2}\left(\frac{\Delta_0}{c_\Delta\beta}\right), \\ & \left.\log_{\tau_2}\left(\frac{\Delta_0}{c_\Delta\gamma}\right)\right].\end{aligned}$$

Proof The proof follows verbatim [4, Lemma 6.23] with (3.12) replacing [4, Eq. (6.36)] and τ_1, τ_2 replacing $\frac{1}{4}$ and 2, respectively. Since $c_\Delta \leq 1$ by definition, the maximum is always a nonnegative quantity. □

Combining Theorem 3.16 with Lemma 3.17 gives the total iteration complexity.

Theorem 3.18 (Iteration complexity of Algorithm 1) *Under the assumptions of Theorem 3.16, for any $\varepsilon_g > 0$, Algorithm 1 produces an approximate minimizer that*

satisfies (1.3) *in at most*

$$\frac{1+\log_{\tau_2}(1/\tau_1)}{\log_{\tau_2}(1/\tau_1)}\left[\frac{C}{\min\left(\underline{\alpha}^2,\underline{\alpha}^{4/3}\underline{\beta},\underline{\alpha}^{4/3}\underline{\gamma},\underline{\alpha}^{2/3}\underline{\gamma}^2,\underline{\beta}^3,\underline{\beta}^2\underline{\gamma},\underline{\beta}\underline{\gamma}^2,\underline{\gamma}^3,\underline{\gamma}^2\underline{\delta}\right)}+1+N^{\text{local}}(\varepsilon_g)\right]$$
$$+\frac{1}{\log_{\tau_2}(1/\tau_1)}\max\left(\log_{\tau_2}\left(\frac{1}{c_\Delta}\right),\log_{\tau_2}\left(\frac{\Delta_0}{c_\Delta\alpha^{\frac{1}{2}}}\right),\log_{\tau_2}\left(\frac{\Delta_0}{c_\Delta\alpha^{\frac{2}{3}}}\right),\log_{\tau_2}\left(\frac{\Delta_0}{c_\Delta\beta}\right),\log_{\tau_2}\left(\frac{\Delta_0}{c_\Delta\gamma}\right)\right) \tag{3.33}$$

iterations, where $N^{\text{local}}(\varepsilon_g)$, C, $\underline{\alpha}$, $\underline{\beta}$, $\underline{\gamma}$, $\underline{\delta}$ *are defined as in Theorem 3.16.*

For a general nonconvex problem, it is not possible to guarantee convergence to a minimizer (1.3). In that sense, the strict saddle property allows to obtain improved complexity bounds but also stronger guarantees of optimality. Classical complexity results are only able to guarantee second-order criticality (1.2), which, a fortiori, Algorithm 1 does in the same number of iterations.

Corollary 3.19 *For any* $\varepsilon_g > 0$, $\varepsilon_H > 0$, *Algorithm 1 returns an* $(\varepsilon_g, \varepsilon_H)-$*second-order critical point* $x_k \in \mathcal{M}$ *that satisfies* (1.2) *in a number of iterations upper bounded by* (3.33).

The bound of Corollary 3.19 holds for any values $\varepsilon_g > 0$ and $\varepsilon_H > 0$, but it is especially relevant when $\varepsilon_g < \alpha$ and $\varepsilon_H < \min(\beta, \gamma, \delta)$. This means that the tolerances for second-order criticality are smaller than the landscape parameters. In that case, the iteration complexity (3.33) is an improvement over the $\mathcal{O}\left(\max(\varepsilon_g^{-2}\varepsilon_H^{-1}, \varepsilon_H^{-3})\right)$ bound of Riemannian trust-region methods for generic nonconvex functions [5]. In our result, the value of ε_g only affect the complexity through the $\mathcal{O}\left(\log\log(\varepsilon_g^{-1})\right)$ factor of $N^{\text{local}}(\varepsilon_g)$ (3.26). For nonconvex Riemannian optimization, cubic regularization has a better complexity than trust-region, as it guarantees (1.2) in at most $\mathcal{O}\left(\max(\varepsilon_g^{-3/2}, \varepsilon_H^{-3})\right)$ iterations [3]. For strict saddle functions, our doubly-logarithmic dependency on ε is also an improvement over Riemannian cubic regularization.

In the unconstrained setting, there exist variants of trust-region methods that have a better complexity than $\mathcal{O}\left(\max(\varepsilon_g^{-2}\varepsilon_H^{-1}, \varepsilon_H^{-3})\right)$. They include [17] with a worst-case complexity of $\mathcal{O}\left(\max(\varepsilon_g^{-3/2}, \varepsilon_H^{-3})\right)$, [16] with a complexity of $\mathcal{O}\left(\max(\varepsilon_g^{-3}\varepsilon_H^{3}, \varepsilon_H^{-3})\right)$, as well as [14, 21] with a complexity of $\mathcal{O}\left(\max(\varepsilon_g^{-2}, \varepsilon_H^{-3})\right)$. In the unconstrained case, our $\mathcal{O}\left(\log\log(\varepsilon_g^{-1})\right)$ result is also an improvement over these variants applied to strict saddle functions. We also improve over the result in [28], which leverages a strict saddle property of low-rank optimization problems with a different regularity condition near minimizers, to ensure a complexity of $\mathcal{O}\left(\log(\max(\varepsilon_g^{-1}, \varepsilon_H^{-1}))\right)$ iterations for a first-order method augmented with negative curvature steps.

Perhaps surprisingly, our bound depends on ε_g but not on ε_H. This is because the strict saddle property implies that $\lambda_{\min}(\mathrm{Hess}\, f(x_k)) < -\varepsilon_H$ is impossible in the region $\mathcal{R}_3$, but $\|\mathrm{grad}\, f(x_k)\| > \epsilon_g$ is.

Remark 3.2 We emphasize that Algorithm 1 is written as an infinite loop, and by Theorem 3.18 it converges towards an isolated minimizer. In general nonconvex optimization, it is common to use a stopping criterion, e.g. to terminate as soon as an $(\varepsilon_g, \varepsilon_H)-$second-order critical point (1.2) is found. If this stopping criterion is added, under the condition that $\varepsilon_g < \alpha$ and $\varepsilon_H < \beta$, the $(\varepsilon_g, \varepsilon_H)-$second-order critical point is guaranteed to be an approximate minimizer satisfying (1.3), as a consequence of the strict saddle structure. For larger values of ε_g and ε_H, the $(\varepsilon_g, \varepsilon_H)-$second-order critical point may be near a point with negative curvature, and the algorithm stopped before detecting the negative curvature.

To end this section, we apply our complexity result to the examples of strict saddle functions from Sect. 2.

Example 3.1 As a continuation of Example 2.1, let $f : \mathbb{R}^n \to \mathbb{R}$ be a γ-strongly convex function, choose $\alpha = 1$ so that f 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 gives at most $\mathcal{O}(\gamma^{-5}) + \log\log(\gamma^3 \varepsilon_g^{-1})$ iterations to find such a point [7]. Although our bound has a better dependency on γ, we believe that this is an artefact of the line-search analysis, which could be improved by changing the line-search condition.

Example 3.2 As a continuation of Example 2.2, let $f : \mathbb{S}^{n-1} \to \mathbb{R}$ be defined by $f(x) = x^{\mathrm{T}} A x$, where $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix with eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_{n-1} > \lambda_n$. Then, by Theorem 3.18, Algorithm 1 computes an iterate satisfying (1.2) in at most

$$\mathcal{O}\left(\max\left(1, \frac{\lambda_1^2}{(\lambda_{n-1}-\lambda_n)^2}, \frac{\lambda_1^{4/3}}{(\lambda_{n-1}-\lambda_n)^{7/3}}, \frac{\lambda_1^{2/3}}{(\lambda_{n-1}-\lambda_n)^{8/3}}, \frac{1}{(\lambda_{n-1}-\lambda_n)^3}, \frac{\lambda_1}{(\lambda_{n-1}-\lambda_n)^3}\right)\right) + \mathcal{O}\left(\log\log\left((\lambda_{n-1}-\lambda_n)^2 \varepsilon_g^{-1}\right)\right)$$

iterations.

4 Riemannian trust-region method with inexact subproblem minimization

In this section, we design an inexact variant of the Riemannian trust-region algorithm, that is tailored to strict saddle functions. In each region of $\mathcal{M}$, some landscape-aware step is appropriate and ensures a good function decrease: gradient-like steps in $\mathcal{R}_1$, negative curvature steps in $\mathcal{R}_2$, and (regularized) Newton steps in $\mathcal{R}_3$. Our goal is to compute these steps approximately, without computing the entire spectrum of the Hessian to determine the region of the current point. The natural choice for this is the

well-known truncated conjugate gradient algorithm [34, 40], and its recent adaptations that yield second-order complexity guarantees [16]. We make minimal adjustements to the standard truncated conjugate gradient (tCG), in order to leverage the strict saddle structure and ensure convergence to second-order critical points. Unlike Algorithm 1, the inexact algorithm (Algorithm 4) explicitly uses the strict saddle parameters α, β, γ. This idea appears in a recent proposal for nonconvex matrix factorization problems, which satisfy a different strict saddle property [28]. Hence, throughout this section, we assume that the parameters α, β, γ are known.

4.1 Inexact algorithm and subroutines

Recall that, at every iteration k, the trust-region subproblem is given by

$$\min_{s \in \mathrm{T}_{x_k}\mathcal{M}} m_k(s) \text{ subject to } \|s\| \leq \Delta_k, \tag{4.1}$$

where m_k is the quadratic model defined in (3.2). For nonzero g_k, we apply a truncated conjugate gradient method to find an approximate solution of the subproblem. Our variant of truncated conjugate gradient, described in Algorithm 2, is a Riemannian adaptation of [16]—a nonconvex trust-region method with complexity guarantees; which we further adapt to strict saddle problems.

The main differences between Algorithm 2 and [16, Algorithm 3.1] lie in the tolerance on the curvature and the stopping criterion. For the former, we use the strict saddle constant γ (the strong convexity constant in $\mathcal{R}_3$) instead of an arbitrary tolerance on the smallest eigenvalue of the Hessian. Throughout the iterations of Algorithm 2, we monitor the curvature of the Hessian along the directions generated by CG. Any curvature less than γ indicates that the current iterate does not belong to $\mathcal{R}_3$, and triggers termination of tCG. We also strengthen the stopping criterion of tCG from [16], and use

$$\left\|r_{j+1}\right\| \leq \zeta \min\left(\|g_k\|^2, \|g_k\|, \gamma \left\|y_{j+1}\right\|\right), \tag{4.2}$$

where $r_j = \nabla m_k(y_j)$ is the residual of the CG algorithm after j iterations, and $\zeta \in (0, 1)$. The term $\|g_k\|^2$ in (4.2) ensures a local quadratic convergence, as we show in Sect. 4.3.

The remaining terms on the right-hand side of (4.2) are used to certify a good decrease when the current iterate belongs to $\mathcal{R}_1$ or $\mathcal{R}_3$. The last term is inspired from cubic regularization methods, which require $\|\nabla m(s_k)\| \leq \mathcal{O}(\|s_k\|^2)$ to obtain a lower bound on the norm of the step. Our condition requires $\|\nabla m(s_k)\| \leq \mathcal{O}(\gamma \|s_k\|)$ and works in a similar way. When H_k is positive definite, the residual condition (4.2) is satisfied in $\min\left(n, \mathcal{O}\left(\gamma^{-1/2}\right)\right)$ iterations. This property is intrinsic to the conjugate gradient algorithm.

Lemma 4.1 *Suppose that we apply the conjugate gradient algorithm to $g_k \in \mathrm{T}_{x_k}\mathcal{M}$ with $\|g_k\| > \varepsilon_g$ and H_k such that $\gamma \,\mathrm{Id} \preceq H_k \preceq \kappa_H \,\mathrm{Id}$. Then, given a tolerance*

Algorithm 2 Strict Saddle version of Truncated Conjugate Gradient (tCG) for subproblem (4.1)

Input: Nonzero gradient g_k, Hessian matrix H_k, trust-region radius Δ_k, accuracy parameter $\zeta \in (0,1)$, bound $\kappa_H \in [\|H_k\|, \infty)$, strict saddle parameter $\gamma > 0$.
Output: trial step s and flag `outCG` indicating termination type

Define $J_{\mathrm{CG}} = \min\left\{n, \frac{1}{2}\sqrt{\frac{\kappa_H}{\gamma}} \ln\left(\frac{2\sqrt{\kappa_H}}{\zeta\sqrt{\gamma}} \max\left(\varepsilon_g^{-1}, \kappa_H/\gamma\right)\right)\right\}$.
Set $y_0 = 0$, $r_0 = g_k$, $p_0 = -g_k$, $j = 0$.
while $j < J_{\mathrm{CG}}$ **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 `outCG = 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 `outCG = not_strongly_convex`
  end if
  $\sigma_j \leftarrow \|r_j\|^2 / \langle p_j, H_k p_j\rangle$ ▷ Begin standard tCG procedure
  $y_{j+1} = y_j + \sigma_j p_j$
  if $\|y_{j+1}\| \geq \Delta_k$ **then**
    Compute $\bar{\sigma}_j \geq 0$ such that $\|y_j + \bar{\sigma}_j p_j\| = \Delta_k$
    return $s \leftarrow y_j + \bar{\sigma}_j p_j$ and `outCG = boundary_step`
  end if
  $r_{j+1} \leftarrow r_j + \sigma_j H_k p_j$
  if condition (4.2) holds **then**
    return $s \leftarrow y_{j+1}$ and `outCG = small_residual`
  end if
  $\tau_{j+1} \leftarrow \|r_{j+1}\|^2 / \|r_j\|^2$
  $p_{j+1} \leftarrow -r_{j+1} + \tau_{j+1} p_j$ ▷ end standard tCG procedure
  $j \leftarrow j + 1$
end while
return $s \leftarrow y_{k_{max}}$ and `outCG = max_iter`

$\zeta \in (0,1)$, *CG finds a vector* $y \in \mathrm{T}_{x_k}\mathcal{M}$ *such that*

$$\|H_k y + g_k\| \leq \zeta \min\left(\|g_k\|^2, \|g_k\|, \gamma \|y\|\right) \tag{4.3}$$

in at most

$$J_{\mathrm{CG}} := \min\left(n, \frac{1}{2}\sqrt{\frac{\kappa_H}{\gamma}} \ln\left(\frac{2\sqrt{\kappa_H}}{\zeta\sqrt{\gamma}} \max\left(\varepsilon_g^{-1}, \kappa_H/\gamma\right)\right)\right) \tag{4.4}$$

iterations or, equivalently, Hessian-vector products.

Proof The result follows from [32, Lemma 11], and in particular the inequality

$$\|H_k y_j + g_k\| \leq 2\sqrt{\frac{\kappa_H}{\gamma}} \left(\frac{\sqrt{\kappa_H/\gamma} - 1}{\sqrt{\kappa_H/\gamma} + 1}\right)^j \|g_k\|,$$

which holds for any $j \leq n$. As long as the stopping criterion is not satisfied, we have

$$\zeta \min\left(\|g_k\|^2, \|g_k\|, \gamma \|y_j\|\right) \leq 2\sqrt{\frac{\kappa_H}{\gamma}} \left(\frac{\sqrt{\kappa_H/\gamma} - 1}{\sqrt{\kappa_H/\gamma} + 1}\right)^j \|g_k\|.$$

As in [32], orthogonality of the residuals gives $\|y_j\| \geq \|g_k\| / \kappa_H$. Since $\gamma \leq \kappa_H$, this gives

$$\left(\frac{\sqrt{\kappa_H/\gamma} - 1}{\sqrt{\kappa_H/\gamma} + 1}\right)^j \geq \frac{\zeta}{2} \frac{\sqrt{\gamma}}{\sqrt{\kappa_H}} \min(\|g_k\|, 1, \gamma/\kappa_H) \tag{4.5}$$

$$\geq \frac{\zeta}{2} \frac{\sqrt{\gamma}}{\sqrt{\kappa_H}} \min(\varepsilon_g, \gamma/\kappa_H). \tag{4.6}$$

Taking the logarithm on both sides gives

$$j \leq \frac{\ln\left(\frac{2\sqrt{\kappa_H}}{\zeta\sqrt{\gamma}} \max(\varepsilon_g^{-1}, \kappa_H/\gamma)\right)}{\ln\left(\frac{\sqrt{\kappa_H/\gamma}-1}{\sqrt{\kappa_H/\gamma}+1}\right)}, \tag{4.7}$$

and we conclude by bounding the denominator in the manner of the aforementionned proof. The model is also minimized in at most n steps, $\|H_k y_n + g_k\| = 0$. □

Lemma 4.1 implies that one can use a cap on the number of iterations, along with checks on the curvature of H_k, to monitor the convergence of tCG. In particular, if $\lambda_{\min}(H_k) \geq \gamma$, then Algorithm 2 satisfies (4.2) in J_{CG} iterations, which may be smaller than n in large dimensions. When this is not the case, and that $x_k \notin \mathcal{R}_1$, the Hessian is guaranteed to have negative curvature, and we compute (an approximation of) the smallest eigenvalue.

To this end, we rely on a minimum eigenvalue oracle (MEO), that either computes a direction of sufficient negative curvature, or certifies that such direction does not exist [16, 31]. One possible implementation of this procedure consists in constructing a full eigenvalue decomposition of the Hessian, which deterministically guarantees the desired outcome but requires access to the entire Hessian matrix (or, equivalently, n Hessian-vector products). Cheaper variants rely on Krylov subspace methods, such as Lanczos' method, that provide the desired guarantee with high probability using potentially less than n Hessian-vector products [31, Appendix B]. For the sake of generality, we describe the MEO as a probabilistic method in Algorithm 3. Similarly to Algorithm 2, a key difference with previous minimum eigenvalue oracles is that the strict saddle constant β (associated with the region of negative curvature $\mathcal{R}_2$) replaces an a priori optimality tolerance on the minimum eigenvalue of the Hessian.

Algorithm 3 Minimum eigenvalue oracle (MEO)

Inputs: Matrix H_k, trust-region radius Δ_k, failure probability tolerance $p \in (0, 1)$, bound $\kappa_H \in [\|H\|, \infty)$, strict saddle parameter β.

Outputs: Either a certificate that $\lambda_{\min}(H_k) \geq -\beta$ valid with probability at least $1 - p$, or a vector $s \in \mathrm{T}_{x_k}\mathcal{M}$ such that

$$\langle g_k, s\rangle \leq 0, \quad \langle s, H_k s\rangle \leq -\frac{1}{2}\beta \|s\|^2, \quad \text{and} \quad \|s\| = \Delta_k. \tag{4.8}$$

Our inexact trust-region method (Algorithm 4) combines tCG (Algorithm 2) and the MEO (Algorithm 3). We first attempt to use tCG to solve subproblem (4.1) approximately. If the current iterate has a large enough gradient ($x_k \in \mathcal{R}_1$) or tCG hits the boundary of the trust region, we use the step given by tCG. Otherwise, we call the MEO to estimate the minimum eigenvalue of the Hessian. If the MEO finds a direction of sufficient negative curvature, we use the best step between the MEO and tCG as our next step. Otherwise, we proceed with the tCG step.

Algorithm 4 Strict saddle RTR with inexact subproblem minimization

Inputs: 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 $0 < \tau_1 < 1 < \tau_2$, failure probability tolerance $p \in [0, 1)$, strict saddle parameters α, β, γ.
for $k = 1, 2, \ldots$ **do**
  if $\|g_k\| > 0$ **then**
    Call Algorithm 2 on the subproblem (4.1) to obtain s_k^{CG} and `outCG`.
  end if
  if $\|g_k\| \geq \alpha$ or `outCG=boundary_step` **then**
    Set $s_k = s_k^{CG}$.
  else if $\|g_k\| = 0$ or (`outCG` $\in$ `{max_iter,not_strongly_convex,small_residual}` and $\|g_k\| < \alpha$) **then**.
    Call Algorithm 3 on the Hessian H_k.
    if Algorithm 3 certifies that $\lambda_{\min}(H_k) > -\beta$ **then**
      Set $s_k = s_k^{CG}$ if $\|g_k\| > 0$, otherwise terminate and return x_k.
    else
      Set $s_k = \underset{s \in \{s_k^{MEO}, s_k^{CG}\}}{\arg\min} m_k(s)$, where s_k^{MEO} is the output of Algorithm 3.
    end if
  end if
  Compute $\rho_k = \dfrac{f(x_k) - f\left(R_{x_k}(s_k)\right)}{m_k(0) - m_k(s_k)}$ and set $x_{k+1} = \begin{cases} R_k(x_k) & \text{if } \rho_k \geq \eta_1 \\ x_k & \text{otherwise.} \end{cases}$

  Set $\Delta_{k+1} = \begin{cases} \min\left(\tau_2 \Delta_k, \bar{\Delta}\right) & \text{if } \rho_k > \eta_2 \quad \text{[very successful]} \\ \Delta_k & \text{if } \eta_2 \geq \rho_k \geq \eta_1 \quad \text{[successful]} \\ \tau_1 \Delta_k & \text{otherwise.} \quad \text{[unsuccessful]} \end{cases}$
end for

Note that when $g_k = 0$, the trust-region method calls Algorithm 3 directly, and terminates if it certifies that $\lambda_{\min}(H_k) \geq -\beta$, since this implies $x_k \in \mathcal{R}_3$ is a local minimum. This corner case is not central to our complexity analysis. Note also that we could have started by calling the MEO instead of tCG, and this would have lead to the same complexity guarantees. However, we believe that starting with tCG is closer to practical implementation of inexact trust-region algorithms. It also follows previous observations that methods that give priority to tCG perform very few calls to the MEO, which is desirable in practice [16].

For the rest of Sect. 4, we consider the following assumption on the MEO (Algorithm 4).

A6 For any iteration of Algorithm 4, if Algorithm 3 is called during this iteration, it outputs the correct answer, i.e., a certificate that $\lambda_{\min}(H_k) > -\beta$ or a step of curvature

less than $-\frac{\beta}{2}$, in at most

$$J_{\text{MEO}} := \min\left(n, 1 + \left\lceil \frac{1}{2}\ln(2.75n/p^2)\sqrt{\frac{\kappa_H}{\beta}} \right\rceil\right) \tag{4.9}$$

iterations, with probability at least $1 - p$.

Computing a full eigenvalue decomposition for Algorithm 3 ensures that A6 holds for any $p \geq 0$, with $J_{\text{MEO}} = n$. The bound (4.9) also applies to Krylov subspace methods, such as Lanczos' method with an initial vector uniformaly distributed on the unit sphere [31].

4.2 Properties of inexact steps

In this section, we provide several decrease lemmas for the steps produced by Algorithms 2 and 3. Our proof technique follows earlier works on Euclidean trust-region methods for general nonconvex functions [16].

We first consider iterations of Algorithm 2 where $x_k \in \mathcal{R}_1$ (large gradient norm).

Lemma 4.2 *Under A2 and A4, consider the kth iteration of Algorithm 4. Suppose that $\|g_k\| \geq \alpha$, so that Algorithm 2 is called. Suppose that it outputs s_k with `outCG` $\neq$ `boundary_step`. Then, we have*

$$m_k(0) - m_k(s_k) \geq \frac{1}{2}\min\left(\Delta_k, \frac{\alpha}{\kappa_H}\right)\alpha. \tag{4.10}$$

Proof By assumption, the step generated by tCG is taken, and it guarantees at least as much model decrease as the Cauchy step s_k^C, defined in the proof of Lemma 3.1. Indeed, the Cauchy step corresponds to the first iterate of tCG [4, 12]. Applying Lemma 3.1 gives

$$m_k(0) - m_k(s_k) \geq m_k(0) - m_k(s_k^C) \geq \frac{1}{2}\min\left(\Delta_k, \frac{\alpha}{\kappa_H}\right)\alpha. \qquad \square$$

The following lemma gives a model decrease when $x_k \in \mathcal{R}_2$ and the MEO is called.

Lemma 4.3 *Under A1 and A2, consider the kth iteration of Algorithm 4. Suppose that Algorithm 3 is called with $x_k \in \mathcal{R}_2$. Then,the method outputs a vector $s_k \in \mathrm{T}_{x_k}\mathcal{M}$ satisfying*

$$m_k(0) - m_k(s_k) \geq \frac{1}{4}\beta\Delta_k^2. \tag{4.11}$$

with probability at least $1 - p$.

Proof Since $x_k \in \mathcal{R}_2$, Algorithm 3 is called with H_k such that $\lambda_{\min}(H_k) \leq -\beta$. Algorithm 4 then outputs a step satisfying (4.8) with probability $1 - p$. Reasoning as in the proof of Lemma 3.2, we bound the model decrease as

$$m_k(0) - m_k(s_k) \geq m_k(0) - m_k(s_k^{MEO}) \geq -\frac{1}{2}\langle s_k, H_k s_k\rangle \geq \frac{\beta}{4}\|s_k\|^2 = \frac{\beta}{4}\Delta_k^2. \qquad \square$$

Note that Lemma 4.3 is a probabilistic result, because it depends on the minimum eigenvalue oracle, but this oracle is only called for small gradients.

We now consider an iteration at which tCG terminates with a small residual, which satisfies (4.2).

Lemma 4.4 *Under A1 and A2, consider the kth iteration of Algorithm 4. Suppose that Algorithm 2 is called and outputs s_k with* `outCG=small_residual`. *Then,*

$$m_k(0) - m_k(s_k) \geq \frac{1}{4}\gamma \|s_k\|^2 . \tag{4.12}$$

In addition, if $\|s_k\| \leq v_R$, we have

$$m_k(0) - m_k(s_k) \geq \frac{1}{2(\kappa_R^2 + 2\hat{L}_H \kappa_R)} \min\left(\left\| \operatorname{grad} f\left(R_{x_k}(s_k)\right)\right\|^2 \gamma^{-1}, \gamma^3 \right). \tag{4.13}$$

Proof By assumption, we have $s_k = y_{j+1}$, where y_{j+1} is the first iterate of tCG that satisfies the residual condition (4.2). As a result,

$$\begin{aligned} \langle s_k, H_k s_k \rangle &= \langle y_j + \sigma_j p_j, H_k(y_j + \sigma_j p_j) \rangle \\ &= \langle y_j, H_k y_j \rangle + \sigma_j \langle y_j, H_k p_j \rangle + \sigma_j \langle p_j, H_k y_j \rangle + \sigma_j^2 \langle p_j, H_k p_j \rangle \\ &= \langle y_j, H_k y_j \rangle + \sigma_j^2 \langle p_j, H_k p_j \rangle , \end{aligned} \tag{4.14}$$

where the last line follows from the standard properties of CG iterates $y_j = \sum_{i=0}^{j-1} \sigma_i p_i$ and $\langle p_j, H_k p_i \rangle = 0$ for $i \neq j$, which implies $\langle p_j, H_k y_j \rangle = 0$ (see, e.g., [27, Chapter 5] or [31, Appendix A]). In addition, Algorithm 2 ensures that $\langle p_j, H_k p_j \rangle \geq \gamma \|p_j\|^2$ and $\langle y_j, H_k y_j \rangle \geq \gamma \|y_j\|^2$, hence

$$\langle s_k, H_k s_k \rangle \geq \gamma \|y_j\|^2 + \gamma \|\sigma_j p_j\|^2 \geq \frac{\gamma}{2} \|s_k\|^2 , \tag{4.15}$$

where the last inequality follows from $\|u\|^2 + \|v\|^2 \geq \frac{1}{2}\|u+v\|^2$ for all vectors $u, v \in \mathrm{T}_{x_k}\mathcal{M}$. Meanwhile, the model decrease satisfies

$$\begin{aligned} m_k(0) - m_k(s_k) = -\langle s_k, g_k \rangle - \frac{1}{2}\langle s_k, H_k s_k \rangle &= -\langle s_k, -H_k s_k + r_{j+1} \rangle - \frac{1}{2}\langle s_k, H_k s_k \rangle \\ &= \frac{1}{2}\langle s_k, H_k s_k \rangle - \langle r_{j+1}, s_k \rangle \\ &= \frac{1}{2}\langle s_k, H_k s_k \rangle , \end{aligned}$$

using that $r_{j+1} = H_k y_{j+1} + g_k = H_k s_k + g_k$ and $\langle r_{j+1}, y_{j+1} \rangle = 0$ by the orthogonality property of the CG residual. Combining the last equality with (4.15), we obtain

$$m_k(0) - m_k(s_k) \geq \frac{1}{4}\gamma \|s_k\|^2 , \tag{4.16}$$

which proves the first part of the proposition.

Now assume $\|s_k\| \leq \nu_R$ and let $g_k^+ = \operatorname{grad} f\left(R_{x_k}(s_k)\right)$. Using the proof of Lemma 3.7 gives

$$\begin{aligned}\left\|g_k^+\right\| &\leq \kappa_R \left\|\nabla \hat{f}_k(s_k) - \operatorname{grad} f(x_k) + \operatorname{grad} f(x_k)\right\| \\ &= \kappa_R \left\|\nabla \hat{f}_k(s_k) - \operatorname{grad} f(x_k) - H_k s_k + r_{j+1}\right\| \\ &\leq \kappa_R \left\|\nabla \hat{f}_k(s_k) - \operatorname{grad} f(x_k) - H_k s_k\right\| + \kappa_R \left\|r_{j+1}\right\| \\ &\leq \frac{\hat{L}_H \kappa_R}{2} \|s_k\|^2 + \kappa_R \gamma \|s_k\|,\end{aligned} \tag{4.17}$$

where the last line follows from the small residual condition (4.2). The inequality

$$\frac{\hat{L}_H \kappa_R}{2} \|s_k\|^2 + \kappa_R \gamma \|s_k\| - \left\|g_k^+\right\| \geq 0 \tag{4.18}$$

involves a univariate quadratic function of $\|s_k\| \geq 0$, and thus (4.18) holds as long as

$$\begin{aligned}\|s_k\| \geq \frac{-\kappa_R \gamma + \sqrt{\kappa_R^2 \gamma^2 + 2\hat{L}_H \kappa_R \left\|g_k^+\right\|}}{\hat{L}_H \kappa_R} &= \left(\frac{-\kappa_R + \sqrt{\kappa_R^2 + 2\hat{L}_H \kappa_R \left\|g_k^+\right\| \gamma^{-2}}}{\hat{L}_H \kappa_R}\right)\gamma \\ &\geq \left(\frac{-\kappa_R + \sqrt{\kappa_R^2 + 2\hat{L}_H \kappa_R}}{\hat{L}_H \kappa_R}\right) \min\left(\left\|g_k^+\right\| \gamma^{-2}, 1\right) \gamma \\ &= \left(\frac{2}{\kappa_R + \sqrt{\kappa_R^2 + 2\hat{L}_H \kappa_R}}\right) \min\left(\left\|g_k^+\right\| \gamma^{-1}, \gamma\right) \\ &\geq \frac{1}{\sqrt{\kappa_R^2 + 2\hat{L}_H \kappa_R}} \min\left(\left\|g_k^+\right\| \gamma^{-1}, \gamma\right),\end{aligned}$$

where we used that $-a + \sqrt{a^2 + bt} \geq (-a + \sqrt{a^2 + b}) \min(t, 1)$ with $a = \kappa_R$, $b = 2\hat{L}_H \kappa_R$ and $t = \|g_{k+1}\| \gamma^{-2}$ and that $\dfrac{2}{1 + \sqrt{1 + c}} \geq \dfrac{1}{\sqrt{1 + c}}$ for any $c > 0$. Combining the above with (4.16), we get

$$m_k(0) - m_k(s_k) \geq \frac{1}{4}\gamma \|s_k\|^2 \geq \frac{1}{2(\kappa_R^2 + 2\hat{L}_H \kappa_R)} \min\left(\left\|\operatorname{grad} f\left(R_{x_k}(s_k)\right)\right\|^2 \gamma^{-1}, \gamma^3\right),$$

proving (4.13). □

The next lemma considers steps that lie on the boundary of the trust region. Note that our proof differs from the general nonconvex setting [16, Lemma 4.3], because we do not add an artificial regularizer in the subproblem.

Lemma 4.5 *Under A1 and A2, consider the kth iteration of Algorithm 4. Suppose that Algorithm 2 is called and outputs s_k together with* `outCG=boundary_step`*. Then,*

$$m_k(0) - m_k(s_k) \geq \frac{1}{4}\gamma \Delta_k^2. \tag{4.19}$$

Proof Since `outCG=boundary_step`, the step has the form $s_k = y_j + \bar{\sigma}_j p_j$ with $\|s_k\| = \Delta_k$ and

$$0 \leq \bar{\sigma}_j \leq \sigma_j = \frac{\|r_j\|^2}{\langle p_j, H_k p_j \rangle} = -\frac{\langle g_k, p_j \rangle}{\langle p_j, H_k p_j \rangle}, \tag{4.20}$$

where the last equality follows from orthogonality of the residuals and item 6 of [31, Lemma 7], as

$$-\langle g_k, p_j \rangle = \left\langle r_0, \sum_{k=0}^{j} (\|r_j\|^2 / \|r_k\|^2) r_k \right\rangle = \left\langle r_0, (\|r_j\|^2 / \|r_0\|^2) r_0 \right\rangle = \|r_j\|^2. \tag{4.21}$$

Since $\langle p_j, H_k p_j \rangle \geq \gamma \|p_j\|^2 > 0$, Equation (4.20) implies

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

from which we obtain

$$\begin{aligned} m_k(0) - m_k(\bar{\sigma}_j p_j) &= -\bar{\sigma}_j \langle g_k, p_j \rangle - \frac{\bar{\sigma}_j^2}{2} \langle p_j, H_k p_j \rangle \\ &\geq \bar{\sigma}_j^2 \langle p_j, H_k p_j \rangle - \frac{\bar{\sigma}_j^2}{2} \langle p_j, H_k p_j \rangle \\ &= \frac{\bar{\sigma}_j^2}{2} \langle p_j, H_k p_j \rangle \\ &\geq \frac{\gamma}{2} \|\bar{\sigma}_j p_j\|^2. \end{aligned} \tag{4.22}$$

The reasoning used to prove (4.12) can be applied to $m_k(0) - m_k(y_j)$ (recall that $\langle y_j, H_k y_j \rangle \geq \gamma \|y_j\|^2$), which gives

$$m_k(0) - m_k(y_j) \geq \frac{\gamma}{2} \|y_j\|^2. \tag{4.23}$$

Finally, using $m_k(s_k) - m_k(0) = m_k(y_j) - m_k(0) + m_k(\bar{\sigma}_j p_j) - m_k(0)$ (see (4.14)), we combine (4.22) and (4.23) to conclude as follows:

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

where the last line holds because $\|u\|^2 + \|v\|^2 \geq \frac{1}{2} \|u+v\|^2$ for all $u, v \in \mathrm{T}_{x_k}\mathcal{M}$. □

We end this section with a lower bound on the trust-region radius based on the decrease lemmas above, akin to the exact setting.

Lemma 4.6 *Let A1, A2, A3 and A4 hold. For any index $k \geq 0$, assume that all calls to the MEO (Algorithm 3) up to iteration k with an iterate in $\mathcal{R}_2$ succeed in finding a direction of sufficient negative curvature. Then, the trust-region radius Δ_k in Algorithm 4 satisfies*

$$\Delta_k \geq \widetilde{\Delta}_{\min} := \tilde{c}_\Delta \min\left(\Delta_0, \alpha^{1/2}, \alpha^{2/3}, \beta, \gamma\right), \tag{4.24}$$

where $\tilde{c}_\Delta = \min\left(1, \tau_1 \frac{3(1-\eta_1)}{2L_H}, \tau_1 \sqrt{\frac{3(1-\eta_1)}{2L_H}}, \tau_1 \sqrt[3]{\frac{3(1-\eta_1)}{2\kappa_H L_H}}\right)$.

Proof The proof follows the lines of the exact case (Lemma 3.4) by considering the quantity $1 - \rho_k$. However, rather than partitioning the iterates according to the strict saddle regions, we consider the various steps that can be produced by the trust-region algorithm.

Consider first that tCG (Algorithm 2) is called and outputs s_k with flag `outCG=small_residual`. Per Lemma 4.4, we know that the model decrease satisfies (4.12). Using A3, we combine this with the bound (3.14) on $f(R_{x_k}(s_k)) - m_k(s_k)$ to give

$$1 - \rho_k = \frac{f(R_{x_k}(s_k)) - m_k(s_k)}{m_k(0) - m_k(s_k)} \leq \frac{(L_H/6)\left\| s_k \right\|^3}{(\gamma/4)\left\| s_k \right\|^2} = \frac{2\,L_H}{3\gamma} \left\| s_k \right\| \leq \frac{2\,L_H}{3\gamma} \Delta_k.$$

Therefore, if `outCG=small_residual` and $\Delta_k \leq \frac{3\gamma(1-\eta_1)}{2\,L_H}$, then $\rho_k \geq \eta_1$ and the iteration is successful.

Suppose now that tCG outputs a boundary step, i.e., `outCG=boundary_step`. Lemma 4.5 combined with (3.14) gives

$$1 - \rho_k \leq \frac{(L_H/6)\Delta_k^3}{(\gamma/4)\Delta_k^2} \leq \frac{2\,L_H}{3\gamma} \Delta_k,$$

hence the same conclusion than in the previous case holds.

If the MEO is called with $x_k \in \mathcal{R}_2$, then by assumption it succeeds in finding a direction of curvature at most $-\beta/2$. Lemma 4.3 combined with (3.14) gives

$$1 - \rho_k \leq \frac{(L_H/6)\Delta_k^3}{(\beta/4)\Delta_k^2} \leq \frac{2L_H}{3\beta}\Delta_k.$$

If $\|g_k\| \geq \alpha$ and s_k is not a boundary step, Lemma 4.2 combined with (3.14) gives

$$1 - \rho_k \leq \frac{(L_H/6)\Delta_k^3}{\frac{1}{2}\min\left(\Delta_k, \frac{\alpha}{\kappa_H}\right)\alpha} \leq \frac{L_H}{3}\max\left(\frac{\Delta_k^2}{\alpha}, \frac{\kappa_H \Delta_k^3}{\alpha^2}\right).$$

We have thus established that the kth iteration is successful as long as

$$\Delta_k \leq \min\left(\frac{3(1-\eta_1)}{2L_H}\gamma, \frac{3(1-\eta_1)}{2L_H}\beta, \sqrt{\frac{3(1-\eta_1)}{L_H}}\alpha^{1/2}, \sqrt[3]{\frac{3(1-\eta_1)}{\kappa_H L_H}}\alpha^{2/3}\right).$$

holds, in which case $\Delta_{k+1} \geq \Delta_k$. Applying the updating rule on Δ_k, we find that the trust-region radius is lower bounded for any $k \geq 0$ by

$$\begin{aligned}\Delta_k &\geq \min\left(\Delta_0, \tau_1\frac{3(1-\eta_1)}{2L_H}\gamma, \tau_1\frac{3(1-\eta_1)}{2L_H}\beta, \tau_1\sqrt{\frac{3(1-\eta_1)}{2L_H}}\alpha^{1/2}, \tau_1\sqrt[3]{\frac{3(1-\eta_1)}{2\kappa_H L_H}}\alpha^{2/3}\right) \\ &\geq \tilde{c}_\Delta \min\left(\Delta_0, \alpha^{1/2}, \alpha^{2/3}\beta, \gamma\right).\end{aligned}$$

□

The result of Lemma 4.6 relies on the MEO not failing to detect sufficient negative curvature if present. In the upcoming analysis, we bound the probability of such failure while deriving our main complexity result.

4.3 Complexity bounds

We now derive complexity bounds for our inexact trust-region method (Algorithm 4), and show that they depend logarithmically on the optimality tolerances, similarly to those established in Sect. 3.4 for the exact algorithm.

We begin by describing the local convergence of Algorithm 4. The reasoning follows the exact setting detailed in Sect. 3.3, and uses several results from that section.

Theorem 4.7 *Suppose that A1 and A2 hold. Let $x_k \in \mathcal{R}_3$ be an iterate produced by Algorithm 4, and let $x^* \in \mathcal{M}$ be a local minimum of (P) such that $\operatorname{dist}(x_k, x^*) \leq \delta$ and f is geodesically γ-strongly convex on $\{y \in \mathcal{M} : \operatorname{dist}(y, x^*) < 2\delta\}$. Finally, suppose that either $\operatorname{grad} f(x_k) = 0$ or that*

$$\|\operatorname{grad} f(x_k)\| < \min\left(\tilde{c}_Q \min\left(1, \gamma, \gamma^2, \gamma\delta\right), \gamma\Delta_k\right), \tag{4.25}$$

where $\tilde{c}_Q = \min\left(\frac{3(1-\eta_1)}{2L_H}, \nu_R, \frac{1}{\kappa_S}, \frac{1}{2}, \frac{1}{\kappa_R(2+\hat{L}_H)}\right)$. *Then, either Algorithm 4 terminates or computes an iterate satisfying* (1.3) *in at most*

$$\tilde{N}^{\text{local}}(\varepsilon_g) := \max\left[0, \log_2\log_2\left(\frac{2\min(1,\gamma^2)}{\kappa_R(\hat{L}_H+2)\varepsilon_g}\right)\right] \tag{4.26}$$

iterations.

Proof Since $x_k \in \mathcal{R}_3$, we know that $\lambda_{\min}(H_k) \geq \gamma > 0$. If $\operatorname{grad} f(x_k)$ is zero, then Algorithm 4 calls Algorithm 3, which *deterministically* outputs a certificate that $\lambda_{\min}(H_k) > -\beta$ since no negative curvature direction exists, hence the algorithm terminates.

Consider now $\|\operatorname{grad} f(x_k)\| > 0$. Since $x_k \in \mathcal{R}_3$, Lemma 3.9 and (4.25) imply that the global minimizer of the trust-region subproblem s_k^* satisfies $\|s_k^*\| \leq \|g_k\|/\gamma < \Delta_k$. As a result, s_k^* is the minimizer of the quadratic model, and lies inside the trust region. The iterates of CG have a norm not greater than the model minimizer [12, Thm 7.5.1], thus we have

$$\|s_k\| \leq \|s_k^*\| < \Delta_k. \tag{4.27}$$

We show that this step leads to a successful iteration. As in Proposition 3.10, we obtain from (4.25) along with the decrease (4.12) and (3.14) that

$$\begin{aligned} f(R_{x_k}(s_k)) - m_k(s_k) + (1-\eta_1)(m_k(s_k) - f(x_k)) &\leq \frac{L_H}{6}\|s_k\|^3 - (1-\eta_1)\frac{\gamma}{4}\|s_k\|^2 \\ &\leq \|s_k\|^2\left(\frac{L_H}{6}\frac{\|g_k\|}{\gamma} - \frac{\gamma}{4}(1-\eta_1)\right) < 0, \end{aligned}$$

showing that $\rho_k \geq \eta_1$ and thus the iteration is successful.

Because $H_k \succeq \gamma\,\mathrm{Id}$ and the model minimizer is inside the trust-region, tCG terminates with a vector s_k that satisfies the small residual condition (4.2). We combine Lemma 3.5, Lemma 3.6, Lemma 3.9 and (4.25) to obtain

$$\begin{aligned} \|g_{k+1}\|_{x_{k+1}} \leq \kappa_R\left\|\nabla\hat{f}_k(s_k)\right\|_{x_k} &\leq \kappa_R\left\|\nabla\hat{f}_k(s_k) - g_k - H_k s_k\right\|_{x_k} + \kappa_R\left\|r_{j+1}\right\|_{x_k} \\ &\leq \kappa_R\frac{\hat{L}_H}{2}\|s_k\|_{x_k}^2 + \kappa_R\|g_k\|_{x_k}^2 \\ &\leq \kappa_R\left(\frac{1}{\gamma^2} + \frac{\hat{L}_H}{2}\right)\|g_k\|_{x_k}^2 \\ &\leq \kappa_R\frac{2+\hat{L}_H}{2\min\left(1,\gamma^2\right)}\|g_k\|_{x_k}^2 < \frac{1}{2}\|g_k\|_{x_k}. \end{aligned} \tag{4.28}$$

As a result, we have $\|g_{k+1}\| < \|g_k\| < \gamma\Delta_k \leq \gamma\Delta_{k+1}$. Applying Proposition 3.12 ensures that $\operatorname{dist}(x_{k+1}, x^*) \leq \delta$ (note that $\tilde{c}_Q < c_Q$ and thus (3.23) holds). We can apply the above reasoning to x_{k+1}, which guarantees that either the method terminates or all subsequent iterates correspond to successful iterations with truncated CG steps

that satisfy (4.2). Finally, to bound the number of iterations before reaching some x_ℓ such that $\|\operatorname{grad} f(x_\ell)\| < \varepsilon_g$, we rewrite (4.28) as

$$\kappa_R \frac{(\hat{L}_H + 2)}{2\min\left(1, \gamma^2\right)} \|g_{k+1}\| \leq \left(\kappa_R \frac{(\hat{L}_H + 2)}{2\min\left(1, \gamma^2\right)} \|g_k\|\right)^2.$$

This shows the iteration bound (4.26) for the local phase, using the proof of Theorem 3.15. □

Having characterized the local convergence behavior of our method, we can now derive global convergence results. As in the exact setting, we first bound the number of successful iterations.

Theorem 4.8 (Number of successful iterations for Algorithm 4) *Suppose that A1–A6 hold. For any $\varepsilon_g > 0$, Algorithm 4 either terminates or produces an iterate satisfying (1.3) in at most*

$$K_{\tilde{\mathcal{S}}} := \frac{\tilde{C}}{\min\left(\underline{\alpha}^2, \underline{\alpha}^{4/3}\underline{\beta}, \underline{\alpha}^{4/3}\underline{\gamma}, \underline{\beta}^3, \underline{\beta}^2\underline{\gamma}, \underline{\beta}\,\underline{\gamma}^2, \underline{\gamma}^3, \underline{\gamma}\underline{\delta}^2\right)} + 1 + \tilde{N}^{\text{local}}(\varepsilon_g)$$

successful iterations with probability $(1-p)^{K_{\tilde{\mathcal{S}}}}$, where $\tilde{N}^{\text{local}}(\varepsilon_g)$ is defined in Equation (4.26), the constant $\tilde{C} > 0$ depends on κ_H, $\tilde{c}_\Delta$, Δ_0, ν_R, κ_R, $\hat{L}_H$, η_1, $\tilde{c}_Q$ and $\underline{\alpha}$, $\underline{\beta}$, $\underline{\gamma}$, $\underline{\delta}$ are defined in Theorem 3.16.

Proof Let $K \in \mathbb{N}$ such that Algorithm 4 has not produced an iterate satisfying (1.3) by iteration K. Then the method cannot terminate before iteration K, and tCG is called at every iteration. We partition the set $\tilde{\mathcal{S}} = \{k \leq K : \rho_k \geq \eta_1\}$ of successful iterations based on properties of the steps, i.e., we define

$$\begin{aligned}
\tilde{\mathcal{S}}_0 &= \left\{k \in \tilde{\mathcal{S}} : \texttt{outCG=boundary_step}\right\}, \\
\tilde{\mathcal{S}}_1 &= \left\{k \in \tilde{\mathcal{S}} \setminus \tilde{\mathcal{S}}_0 : x_k \in \mathcal{R}_1\right\}, \\
\tilde{\mathcal{S}}_2 &= \left\{k \in \tilde{\mathcal{S}} \setminus \tilde{\mathcal{S}}_0 : x_k \in \mathcal{R}_2 \setminus \mathcal{R}_1\right\}, \\
\tilde{\mathcal{S}}_3^l &= \left\{k \in \tilde{\mathcal{S}} \setminus \tilde{\mathcal{S}}_0 : x_k \in \mathcal{R}_3 \setminus \mathcal{R}_1, \texttt{outCG=small_residual}, \|s_k\| > \nu_R\right\}, \\
\tilde{\mathcal{S}}_3^s &= \left\{k \in \tilde{\mathcal{S}} \setminus \tilde{\mathcal{S}}_0 : x_k \in \mathcal{R}_3 \setminus \mathcal{R}_1, \texttt{outCG=small_residual}, \|s_k\| \leq \nu_R\right\}.
\end{aligned}$$

First consider $k \in \tilde{\mathcal{S}}_0$. We obtain through Lemma 4.5 that

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{4}\gamma\Delta_k^2 \geq \frac{\eta_1}{4}\gamma\tilde{\Delta}_{\min}^2. \tag{4.29}$$

For $k \in \tilde{\mathcal{S}}_1$, we have $\|g_k\| \geq \alpha$ and Lemma 4.2 gives

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{2} \min\left(\Delta_k, \frac{\alpha}{\kappa_H}\right)\alpha \geq \frac{\eta_1}{2}\min\left(\frac{\alpha^2}{\kappa_H}, \alpha\widetilde{\Delta}_{\min}\right). \tag{4.30}$$

Consider now $k \in \tilde{\mathcal{S}}_2$, Lemma 4.3 gives

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{4}\beta\Delta_k^2 \geq \frac{\eta_1}{4}\beta\widetilde{\Delta}_{\min}^2, \tag{4.31}$$

with probability $1 - p$.
For any $k \in \tilde{\mathcal{S}}_3^l$, Lemma 4.4 guarantees that

$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{4}\gamma\|s_k\|^2 \geq \frac{\eta_1 \nu_R^2}{4}\gamma. \tag{4.32}$$

For any $k \in \tilde{\mathcal{S}}_3^s$, Lemma 4.4 and (4.13) apply. We further partition this set into $\tilde{\mathcal{S}}_3^{s,s} \cup \tilde{\mathcal{S}}_3^{s,l}$ where

$$\begin{aligned}
\tilde{\mathcal{S}}_3^{s,l} &= \left\{k \in \tilde{\mathcal{S}}_3^s : \|g_{k+1}\| \geq \min\left(\tilde{c}_Q \min(1, \gamma, \gamma^2, \gamma\delta), \gamma\Delta_k\right)\right\}, \\
\tilde{\mathcal{S}}_3^{s,s} &= \tilde{\mathcal{S}}_3^s \setminus \tilde{\mathcal{S}}_3^{s,l}.
\end{aligned}$$

If $k \in \tilde{\mathcal{S}}_3^{s,l}$, Equation (4.13) implies that

$$\begin{aligned}
f(x_k) - f(x_{k+1}) &\geq \frac{\eta_1}{2(\kappa_R^2 + 2\hat{L}_H\kappa_R)} \min\left(\|g_{k+1}\|^2\gamma^{-1}, \gamma^3\right) \\
&\geq \frac{\eta_1}{2(\kappa_R^2 + 2\hat{L}_H\kappa_R)} \min\left(\tilde{c}_Q^2 \min(\gamma^{-1}, \gamma, \gamma^3, \gamma\delta^2), \gamma\Delta_k^2, \gamma^3\right) \\
&\geq \frac{\eta_1}{2(\kappa_R^2 + 2\hat{L}_H\kappa_R)} \min\left(\tilde{c}_Q^2 \min(\gamma^{-1}, \gamma, \gamma^3, \gamma\delta^2), \gamma\widetilde{\Delta}_{\min}^2, \gamma^3\right).
\end{aligned} \tag{4.33}$$

Finally, if $k \in \tilde{\mathcal{S}}_3^{s,s}$, we discuss based on the region of x_{k+1}. If $x_{k+1} \in \mathcal{R}_3$, the local phase begins at x_{k+1} per Theorem 4.7. If $x_{k+1} \in \mathcal{R}_1 \backslash \mathcal{R}_3$, we have $\|g_{k+1}\| \geq \alpha$ and $k+1 \in \tilde{\mathcal{S}}_0 \cup \tilde{\mathcal{S}}_1$. Finally, if $x_{k+1} \in \mathcal{R}_2$, it must be that $k+1 \in \tilde{\mathcal{S}}_0 \cup \tilde{\mathcal{S}}_2$. Thus,

$$\left|\tilde{\mathcal{S}}_3^{s,s}\right| \leq \left|\tilde{\mathcal{S}}_0\right| + \left|\tilde{\mathcal{S}}_1\right| + \left|\tilde{\mathcal{S}}_2\right| + 1 + \tilde{N}^{\text{local}}(\varepsilon_g) \tag{4.34}$$

and it suffices to bound $|\tilde{\mathcal{S}}_0|$, $|\tilde{\mathcal{S}}_1|$, $|\tilde{\mathcal{S}}_2|$, $|\tilde{\mathcal{S}}_3^l|$, and $|\tilde{\mathcal{S}}_3^{s,l}|$ to bound $|\tilde{\mathcal{S}}|$. The rest of the proof is similar to the proof of Theorem 3.16 for the exact case. The following inequality holds by A5:

$$
\begin{aligned}
f(x_0) - f^* \geq & \sum_{k \in \tilde{\mathcal{S}}_0} f(x_k) - f(x_{k+1}) + \sum_{k \in \tilde{\mathcal{S}}_1} f(x_k) - f(x_{k+1}) \\
& + \sum_{k \in \tilde{\mathcal{S}}_2} f(x_k) - f(x_{k+1}) \\
& + \sum_{k \in \tilde{\mathcal{S}}_3^l} f(x_k) - f(x_{k+1}) + \sum_{k \in \tilde{\mathcal{S}}_3^{s,l}} f(x_k) - f(x_{k+1}).
\end{aligned}
$$

Combining this inequality with (4.29), (4.30), (4.31), (4.32) and (4.33), and considering each sum individually, we obtain

$$
\begin{aligned}
\left|\tilde{\mathcal{S}}_0\right| &\leq \frac{4(f(x_0) - f^*)}{\eta_1} \gamma^{-1} \widetilde{\Delta}_{\min}^{-2}, \\
\left|\tilde{\mathcal{S}}_1\right| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \max\left(\kappa_H \alpha^{-2}, \alpha^{-1} \widetilde{\Delta}_{\min}^{-1}\right), \\
\left|\tilde{\mathcal{S}}_2\right| &\leq \frac{4(f(x_0) - f^*)}{\eta_1} \beta^{-1} \widetilde{\Delta}_{\min}^{-2}, \\
\left|\tilde{\mathcal{S}}_3^l\right| &\leq \frac{4(f(x_0) - f^*)}{\eta_1} \nu_R^{-2} \gamma^{-1}, \\
\left|\tilde{\mathcal{S}}_3^{s,l}\right| &\leq \frac{2(\kappa_R^2 + 2\hat{L}_H \kappa_R)(f(x_0) - f^*)}{\eta_1} \\
&\quad \max\left((\tilde{c}_Q)^{-2} \min\left(\gamma^{-1}, \gamma, \gamma^3, \gamma\delta^2\right)^{-1}, \gamma^{-1} \widetilde{\Delta}_{\min}^{-2}, \gamma^{-3}\right).
\end{aligned}
$$

Using the definition of $\widetilde{\Delta}_{\min}$ (4.24) as well as $\underline{\alpha}$, $\underline{\beta}$,$\underline{\gamma}$,$\underline{\delta}$, we obtain

$$
\begin{aligned}
\left|\tilde{\mathcal{S}}_0\right| &\leq \frac{4(f(x_0) - f^*)}{\eta_1} \max\left(\Delta_0^{-2}, 1\right) \min\left(\underline{\alpha}^{4/3}\underline{\gamma}, \underline{\beta}^2\underline{\gamma}, \underline{\gamma}^3\right)^{-1}, \\
\left|\tilde{\mathcal{S}}_1\right| &\leq \frac{2(f(x_0) - f^*)}{\eta_1} \max\left(\kappa_H, (\tilde{c}_\Delta \Delta_0)^{-1}, (\tilde{c}_\Delta)^{-1}\right) \\
&\quad \times \min\left(\underline{\alpha}^2, \underline{\alpha}\underline{\beta}, \underline{\alpha}\underline{\gamma}\right)^{-1}, \\
\left|\tilde{\mathcal{S}}_2\right| &\leq \frac{4(f(x_0) - f^*)}{\eta_1} \max\left(\Delta_0^{-2}, 1\right) \min\left(\underline{\alpha}^{4/3}\underline{\beta}, \underline{\beta}^3, \underline{\gamma}^2\underline{\beta}\right)^{-1}, \\
\left|\tilde{\mathcal{S}}_3^l\right| &\leq \frac{4(f(x_0) - f^*)}{\eta_1} \nu_R^{-2} \underline{\gamma}^{-1}, \\
\left|\tilde{\mathcal{S}}_3^{s,l}\right| &\leq \frac{2(\kappa_R^2 + 2\hat{L}_H \kappa_R)(f(x_0) - f^*)}{\eta_1} \max\left((\tilde{c}_\Delta \Delta_0)^{-2}, (\tilde{c}_Q)^{-2}\right) \\
&\quad \times \min\left(\underline{\alpha}^{4/3}\underline{\gamma}, \underline{\beta}^2\underline{\gamma}, \underline{\gamma}^3, \underline{\gamma}\underline{\delta}^2\right)^{-1}.
\end{aligned}
$$

Combining these bounds with (4.34) gives

$$\begin{aligned}\left|\tilde{\mathcal{S}}\right| &= \left|\tilde{\mathcal{S}}_0\right| + \left|\tilde{\mathcal{S}}_1\right| + \left|\tilde{\mathcal{S}}_2\right| + \left|\tilde{\mathcal{S}}_3^l\right| + \left|\tilde{\mathcal{S}}_3^{s,l}\right| + \left|\tilde{\mathcal{S}}_3^{s,s}\right| \\ &\leq 2\left|\tilde{\mathcal{S}}_0\right| + 2\left|\tilde{\mathcal{S}}_1\right| + 2\left|\tilde{\mathcal{S}}_2\right| + \left|\tilde{\mathcal{S}}_3^l\right| + \left|\tilde{\mathcal{S}}_3^{s,l}\right| + 1 + \tilde{N}^{\text{local}}(\varepsilon_g) \\ &\leq \tilde{C}\min\left(\underline{\alpha}^2, \underline{\alpha}^{4/3}\underline{\beta}, \underline{\alpha}^{4/3}\underline{\gamma}, \underline{\beta}^3, \underline{\beta}\,\underline{\gamma}^2, \underline{\beta}^2\underline{\gamma}, \underline{\gamma}^3, \underline{\gamma}\underline{\delta}^2\right)^{-1} + 1 + \tilde{N}^{\text{local}}(\varepsilon_g)\end{aligned}$$

with

$$\begin{aligned}\tilde{C} := \tfrac{4(f(x_0)-f^*)}{\eta_1}\Big[&\max\left(\Delta_0^{-2}, 1\right) + \max\left(\kappa_H, (\tilde{c}_\Delta\Delta_0)^{-1}, (\tilde{c}_\Delta)^{-1}\right) + \nu_R^{-2} \\ &+ \tfrac{(\kappa_R^2+2L\kappa_R)}{2}\max\left((\tilde{c}_\Delta\Delta_0)^{-2}, (\tilde{c}_Q)^{-2}\right)\Big].\end{aligned}$$

Thus we have $K \leq K_{\tilde{\mathcal{S}}}$ if none of the calls to the MEO with $x_k \in \mathcal{R}_2$ fail to find sufficient negative curvature. The probability of all calls succeeding is bounded below by $(1-p)^{|\tilde{\mathcal{S}}_2|} \geq (1-p)^K \geq (1-p)^{K_{\tilde{\mathcal{S}}}}$, hence the conclusion. □

As in the exact setting, we can express the number of total iterations as a function of the number of successful iterations. This leads to the following result.

Theorem 4.9 *Under the assumptions of Theorem 4.8, for any $\varepsilon_g > 0$, Algorithm 4 produces a point that satisfies (1.3) in at most*

$$\begin{aligned}\tilde{K} = &\frac{1+\log_{\tau_2}(1/\tau_1)}{\log_{\tau_2}(1/\tau_1)}K_{\tilde{\mathcal{S}}} + \frac{1}{\log_{\tau_2}(1/\tau_1)} \\ &\max\left(0, \log_{\tau_2}\left(\frac{1}{\tilde{c}_\Delta}\right), \log_{\tau_2}\left(\frac{\Delta_0}{\tilde{c}_\Delta\underline{\alpha}^{\frac{2}{3}}}\right), \log_{\tau_2}\left(\frac{\Delta_0}{\tilde{c}_\Delta\underline{\beta}}\right), \log_{\tau_2}\left(\frac{\Delta_0}{\tilde{c}_\Delta\underline{\gamma}}\right)\right)\end{aligned}$$

iterations with probability at least $(1-p)^{\tilde{K}}$.

Proof Let $K \in \mathbb{N}$ be an iteration index satisfying the same assumptions than in the proof of Theorem 4.8, and let $\tilde{\mathcal{S}}$ be defined according to K as in that proof. By the same argument as in Lemma 3.17, we have that

$$\begin{aligned}|\tilde{\mathcal{S}}| \geq &\frac{\log_{\tau_2}(1/\tau_1)}{1+\log_{\tau_2}(1/\tau_1)}(K+1) \\ &-\frac{1}{1+\log_{\tau_2}(1/\tau_1)}\max\Bigg[0, \log_{\tau_2}\left(\frac{1}{\tilde{c}_\Delta}\right), \log_{\tau_2}\left(\frac{\Delta_0}{\tilde{c}_\Delta\alpha^{\frac{1}{2}}}\right), \log_{\tau_2}\left(\frac{\Delta_0}{\tilde{c}_\Delta\alpha^{\frac{2}{3}}}\right), \\ &\log_{\tau_2}\left(\frac{\Delta_0}{\tilde{c}_\Delta\beta}\right), \\ &\log_{\tau_2}\left(\frac{\Delta_0}{\tilde{c}_\Delta\gamma}\right)\Bigg].\end{aligned}$$

Combining this result with Theorem 4.9 gives the desired bound, that holds with probability at least $(1-p)^{\tilde{K}}$ if Algorithm 3 can be called at every iteration, and $(1-p)^{K_{\tilde{S}}}$ if it is called only on successful iterations. □

Remark 4.1 While implementing Algorithm 4, one may avoid repeated calls to the MEO if a negative curvature direction was found at iteration k and the iteration is unsuccessful [16, Implementation strategy 4.1]. Such an approach can improve the probability bounds in Theorems 4.8 and 4.9 to $(1-p)^{K_{\tilde{\mathcal{S}}_2}}$, where $K_{\tilde{\mathcal{S}}_2}$ is a bound on the number of iterations in $\tilde{\mathcal{S}}_2$ as defined in the proof of the theorem. In this paper, we focus on the improvement brought by the strict saddle properties of f, and therefore use a simpler, albeit suboptimal bound on the probability.

Since Algorithm 4 relies on an iterative procedure to solve the subproblems, we also provide a bound on the number of Hessian-vector products required to reach an approximate stationary point. The bound is obtained by accounting for both the inner iterations of tCG and the cost of the MEO. The former is bounded by J_{CG} in the algorithm, while the latter follows directly from Assumption A6. Combining these properties with the result of Theorem 4.9, we obtain the following result.

Corollary 4.10 *Under the assumptions of Theorem 4.9, for any $\varepsilon_g > 0$, the total number of Hessian-vector products performed by Algorithm 4 and its subroutines before producing an iterate satisfying (1.3) is at most*

$$\begin{aligned}\max(J_{\mathrm{CG}}, J_{\mathrm{MEO}})\tilde{K} = \mathcal{O}\Big(&\max\Big(n, \ln\Big(\max\Big(\varepsilon_g^{-1}\underline{\gamma}^{-1/2}, \underline{\gamma}^{-3/2}\Big)\Big)\underline{\gamma}^{-1/2}, \ln(n)\underline{\beta}^{-1/2}\Big)\\ &\times\Big[\min\Big(\underline{\alpha}^2, \underline{\alpha}^{4/3}\underline{\beta}, \underline{\alpha}^{4/3}\underline{\gamma}, \underline{\beta}^3, \underline{\beta}\,\underline{\gamma}^2, \underline{\beta}^2\underline{\gamma}, \underline{\gamma}^3, \underline{\gamma}\underline{\delta}^2\Big)^{-1} + \tilde{N}^{\mathrm{local}}(\varepsilon_g)\Big]\Big)\end{aligned} \tag{4.35}$$

with probability $(1-p)^{\tilde{K}}$, where $\tilde{K}$ is defined in Theorem 4.9 and $\tilde{N}^{\mathrm{local}}(\varepsilon_g)$ is defined in (4.26).

A fortiori, for any $\varepsilon_g > 0$, $\varepsilon_H > 0$, Algorithm 4 produces an $(\varepsilon_g, \varepsilon_H)-$second-order critical point (1.2) in a number of Hessian-vector products bounded by (4.35). Note that we stated Algorithm 4 as an infinite loop which converges to a minimizer, and Remark 3.2 about the possibility of adding a stopping criterion also applies here. As in the exact case, we observe that using the strict saddle structure yields bounds that depend logarithmically on ε_g, which is an improvement on the state of the art for Riemannian nonconvex optimization [5].

As a final comment, we note that the operation complexity bound (4.35) exhibits an additional logarithmic factor in ε_g compared to the iteration complexity bound of Theorem 4.9, but no additional dependency on ε_H, unlike in the case of general nonconvex functions. Overall, the complexity guarantees of the inexact variant that we propose also gives an improvement over existing results in the Euclidean case, for which the dependency on ε_g is polynomial [16, 18, 31].

5 Conclusions

We show that worst-case complexity guarantees of Riemannian trust-region algorithms on nonconvex functions improve significantly when the function satisfies a strict saddle property. The guarantees we provide only depend logarithmically on the prescribed optimality tolerances, and, as such, are a better reflection of how problem-dependent quantities affect the performance. 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 relies on the local quadratic convergence of Newton's method, and can be adapted to inexact subproblem minimizations 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 [28]. Investigating the numerical performance of these algorithms, along with their multiple possibilities for implementation, will be the subject of future work. Extending our results to a broader class of strict saddle functions which includes non-isolated minimizers would also be valuable.

Acknowledgements The authors are grateful to Coralia Cartis for useful discussions regarding local convergence of trust-region methods and for sharing reference [33]. This research was partially funded by the Agence Nationale de la Recherche through program ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

References

1. Absil, P.-A., Baker, C., Gallivan, K.: Trust-region methods on Riemannian manifolds. Found. Comput. Math. **7**(3), 303–330 (2007)
2. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2008)
3. Agarwal, N., Boumal, N., Bullins, B., Cartis, C.: Adaptive regularization with cubics on manifolds. Math. Program. **188**(1), 85–134 (2021)
4. Boumal, N.: An Introduction to Optimization on Smooth Manifolds. Cambridge University Press, Cambridge (2023)
5. Boumal, N., Absil, P.-A., Cartis, C.: Global rates of convergence for nonconvex optimization on manifolds. IMA J. Numer. Anal. **39**(1), 1–33 (2019)
6. Boumal, N., Voroninski, V., Bandeira, A.S.: Deterministic guarantees for Burer–Monteiro factorizations of smooth semidefinite programs. Commun. Pure Appl. Math. **73**(3), 581–608 (2020)
7. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
8. Carmon, Y., Duchi, J.C., Hinder, O., Sidford, A.: Accelerated methods for nonconvex optimization. SIAM J. Optim. **28**(2), 1751–1772 (2018)
9. Cartis, C., Gould, N.I.M., Toint, P.L.: Complexity bounds for second-order optimality in unconstrained optimization. J. Complex. **28**, 93–108 (2012)
10. Cartis, C., Gould, N.I.M., Toint, P.L.: Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. In: Proceedings of the international congress of mathematicians (ICM 2018), vol. 3, pp. 3697–3738 (2019)
11. Cartis, C., Gould, N.I.M., Toint, P.L.: Evaluation complexity of algorithms for nonconvex optimization. MOS-SIAM Series on Optimization (2022)
12. Conn, A.R., Gould, N.I., Toint, P.L.: Trust Region Methods, vol. 1. SIAM, Philadelphia (2000)
13. Criscitiello, C., Boumal, N.: Efficiently escaping saddle points on manifolds. In: Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc (2019)
14. Curtis, F.E., Lubberts, Z., Robinson, D.P.: Concise complexity analyses for trust region methods. Opt. Lett. **12**, 1713–1724 (2018)

15. Curtis, F.E., Robinson, D.P.: Regional complexity analysis of algorithms for nonconvex smooth optimization. Math. Program. **187**(1), 579–615 (2021)
16. Curtis, F.E., Robinson, D.P., Royer, C.W., Wright, S.J.: Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization. SIAM J. Optim. **31**(1), 518–544 (2021)
17. Curtis, F.E., Robinson, D.P., Samadi, M.: A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. Math. Program. **162**, 1–32 (2017)
18. Curtis, F.E., Wang, Q.: Worst-case complexity of trace with inexact subproblem solutions for nonconvex smooth optimization. SIAM J. Optim. **33**, 2191–2221 (2023)
19. Ge, R., Huang, F., Jin, C., Yuan, Y.: Escaping from saddle points — online stochastic gradient for tensor decomposition. In: Grünwald, P., Hazan, E., Kale, S., editors, Proceedings of the 28th conference on learning theory, vol. 40 of Proceedings of machine learning research, pages 797–842, Paris, France. PMLR (2015)
20. Ge, R., Lee, J.D., Ma, T.: Matrix Completion has no spurious local minimum. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol 29, pp. 2973–2981. Curran Associates Inc (2016)
21. Gratton, S., Royer, C.W., Vicente, L.N.: A decoupled first/second-order steps technique for nonconvex nonlinear unconstrained optimization with improved complexity bounds. Math. Program. **179**, 195–222 (2020)
22. Lee, J.D., Panageas, I., Piliouras, G., Simchowitz, M., Jordan, M.I., Recht, B.: First-order methods almost always avoid strict saddle points. Math. Program. **176**(1), 311–337 (2019)
23. Li, Q., Zhu, Z., Tang, G.: The non-convex geometry of low-rank matrix optimization. Inf. Inference: J. IMA **8**(1), 51–96 (2019)
24. Liu, Y., Roosta, F.: A Newton-MR algorithm with complexity guarantees for nonconvex smooth unconstrained optimization. (arXiv:2208.07095) (2023)
25. Luo, Y., Trillos, N.G.: Nonconvex matrix factorization is geodesically convex: global landscape analysis for fixed-rank matrix optimization from a Riemannian perspective (2022)
26. Moré, J.J., Sorensen, D.C.: Computing a trust region step. SIAM J. Sci. Stat. Comput. **4**(3), 553–572 (1983)
27. Nocedal, J., Wright, S.: Numerical Optimization. Springer Science & Business Media, Berlin (2006)
28. O'Neill, M., Wright, S.J.: A line-search descent algorithm for strict saddle functions with complexity guarantees. J. Mach. Learn. Res. **24**(10), 1–34 (2023)
29. Paternain, S., Mokhtari, A., Ribeiro, A.: A Newton-based method for nonconvex optimization with fast evasion of saddle points. SIAM J. Optim. **29**(1), 343–368 (2019)
30. Qu, Q., Zhai, Y., Li, X., Zhang, Y., Zhu, Z.: Analysis of the optimization landscapes for overcomplete representation learning. arXiv preprint arXiv:1912.02427 (2019)
31. Royer, C.W., O'Neill, M., Wright, S.J.: A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization. Math. Program. **180**(1), 451–488 (2020)
32. Royer, C.W., Wright, S.J.: Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. SIAM J. Optim. **28**(2), 1448–1477 (2018)
33. Shek, P.: Local convergence of cubic regularisation methods for unconstrained non-convex optimisation problems. Part C Master dissertation (supervisor: C. Cartis), Mathematical Institute, University of Oxford (2015)
34. Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. SIAM J. Numer. Anal. **20**(3), 626–637 (1983)
35. Sun, J., Qu, Q., Wright, J.: When are nonconvex problems not scary? arXiv preprint arXiv:1510.06096 (2015)
36. Sun, J., Qu, Q., Wright, J.: Complete dictionary recovery over the sphere I: overview and the geometric picture. IEEE Trans. Inf. Theory **63**(2), 853–884 (2017)
37. Sun, J., Qu, Q., Wright, J.: Complete dictionary recovery over the sphere II: recovery by Riemannian trust-region method. IEEE Trans. Inf. Theory **63**(2), 885–914 (2017)
38. Sun, J., Qu, Q., Wright, J.: A geometric analysis of phase retrieval. Found. Comput. Math. **18**(5), 1131–1198 (2018)
39. Sun, Y., Flammarion, N., Fazel, M.: Escaping from saddle points on Riemannian manifolds. In: Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc (2019)
40. Toint, P.: Towards an efficient sparsity exploiting Newton method for minimization. In: Duff, I.S. (ed.) Sparse Matrices and Their Uses, pp. 57–88. Academic press, London (1981)

41. Wright, J., Ma, Y.: High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications. Cambridge University Press, Cambridge (2022)
42. Zhang, H., Sra, S.: First-order methods for geodesically convex optimization. In: Conference on learning theory, pp. 1617–1638 (2016)