A New Insight into Newton' s Method for Optimization on Riemannian Manifolds

Authors: WuJun Che, WuJun Che

Date: 2026-02-11T16:55:01+00:00

Abstract

Newton' s method, a cornerstone algorithm for root-finding in Euclidean spaces, has been extended to Riemannian manifolds to tackle an increasingly broad class of optimization problems. While its computational characteristics and convergence properties have been extensively studied in the literature, the underlying algorithmic behavior of Riemannian Newton' s method remains inadequately characterized by existing theoretical frameworks—despite its proven practical effectiveness. This paper presents a novel theoretical perspective grounded in the principle of form invariance. We establish that the second-order Taylor expansion of a real-valued function possesses form invariance in local coordinates on a Riemannian manifold, which naturally gives rise to a fully quadratic model for Newton' s method. The resultant Newton step formula is identical to its Euclidean counterpart, preserving computational simplicity while accommodating the intrinsic curvature of the manifold. This work not only provides a rigorous explanation for the algorithm' s behavior but also paves the way for more efficient computational implementations.

Full Text

Preamble

A New Insight into Newton's Method for Optimization on Riemannian Manifolds
WuJun Che Institute of Automation, CAS, Beijing, 100190, Beijing, China.

Contributing authors: wujun.che@ia.ac.cn;

Abstract

Newton' s method, a cornerstone algorithm for root-finding in Euclidean spaces, has been extended to Riemannian manifolds to tackle an increasingly broad class of optimization problems. While its computational characteristics and conver-

gence properties have been extensively studied in the literature, the underlying algorithmic behavior of Riemannian Newton' s method remains inadequately characterized by existing theoretical frameworks—despite its proven practical effectiveness. This paper presents a novel theoretical perspective grounded in the principle of form invariance. We establish that the second-order Taylor expansion of a real-valued function possesses form invariance in local coordinates on a Riemannian manifold, which naturally gives rise to a fully quadratic model for Newton' s method. The resultant Newton step formula is identical to its Euclidean counterpart, preserving computational simplicity while accommodating the intrinsic curvature of the manifold. This work not only provides a rigorous explanation for the algorithm' s behavior but also paves the way for more efficient computational implementations.

Keywords: Newton' s method, Riemannian manifold, form invariance, optimization, second-Order Taylor Expansion MSC Classification: 49M15 , 65K10 , 53B20 , 41A58

1.1 Background

Optimizing a real-valued function on a Riemannian manifold has emerged as a cen- tral task in science and engineering, spanning a wide range of nonlinear constrained optimization problems [1–4]. In such problems, the feasible set constitutes a nonlinear space, rendering them substantially more challenging to solve than their unconstrained counterparts [5]. While traditional algebraic methods are applicable, differential geom- etry provides a powerful tool and natural language for characterizing these geometric properties, often facilitating more intuitive theoretical analysis and more effective numerical computation [6]. Riemannian optimization addresses this class of problems by treating the constraint set as a manifold. Algorithms within this framework perform a series of descent steps along geodesics, leveraging intrinsic geometry of the manifold.

This paper employs various distinct mathematical notations. To ensure clarity and consistency in notational conventions throughout the work, typical symbol usages are summarized in Appendix A.

Euclidean n-space Rn, is the simplest example of a Riemannian manifold. Extend- ing well-established optimization methods originally developed in Rn to general Riemannian manifolds is a natural and widely adopted practice. The second-order Taylor expansion of an objective function f at a point x   Rn with a perturbation s   Rn takes a particularly concise form: f (x + s) = f (x) + ∂f (x)T s + s T $ $2f (x)s + O(s 3).

The core idea of Newton' s method for minimizing f lies in iteratively applying Eq.1 to refine the current estimate along a descent direction. The steepest descent direction, −∂f (x), arises from a linear approximation of f , whereas the Newton step, −$ $2f(x) − 1$ $f (x), stems from a quadratic approximation and represents the second-order steepest descent direction. Beyond its rapid quadratic convergence rate, Newton' s method frequently serves as a key component in other numerical algorithms, such as trust-region methods [5, 7].

Riemannian optimization fundamentally involves leveraging the local geometric information of the objective function and its constraints. However, due to the substan- tial differences in differential structures between a general manifold M and Euclidean space Rn, it is widely acknowledged that the proper extension of Newton' s method from Rn to M necessitates a robust framework of differential geometry—including concepts like the Riemannian connection and covariant derivatives—as well as com- putationally intensive calculations of geodesics and parallel translation [1, 8, 9].

Furthermore, existing theory still lacks satisfactory explanations for certain algorith- mic behaviors of Riemannian Newton' s method. For instance, while local normal coordinates (induced by the exponential map) are employed to handle the Taylor expansion of f on a " curved manifold" [8], they do not adequately account for why Newton iterations retain robust quadratic convergence, even when the acceleration term is deliberately omitted [9]. The fundamental distinction between the Rieman- nian and Euclidean versions of Newton' s method warrants further study. This paper explores these open questions from a new perspective.

1.2 Preliminaries

This section elaborates on the Taylor expansion of a differentiable function and its application to Riemannian optimization. Foundational concepts in Riemannian geometry can be found in [10–12].

1.2.1 Taylor expansion on Riemannian manifolds

Let M be a smooth manifold equipped with a Riemannian metric g. Consider a vector field X on M and its associated local flow ϕt for small t $(-\epsilon, \epsilon)$. For a point p M , the curve c(t) = ϕt(p) is the integral curve for X passing through p at t = 0. Let D be the set of functions on M that are differentiable at p. The tangent vector to this curve at t = 0 defines a linear operator X p $c'(0)$ on D, given by d(f $\circ$ c) (cid:12) (cid:12) (cid:12) (cid:12)t=0 f D The collection of all such vectors to M at p forms the tangent space, denoted by Tp M .

The vector field v (t) $c'(t)$ Tc(t)M , is referred to as the velocity field of c.

The ordinary derivative of v along c is not intrinsically defined. To resolve this issue, we employ the covariant derivative D dt , which induced by the Riemannian con- nection ∇. For a vector field Y induced by X T M along c, i.e. Y (t) = X $\circ$ c(t), dt Y ∇dc/dtY . This enables us to define the intrinsic its covariant derivative is D acceleration of the curve c as a(t) $c''(t)$ = D dt v (t) Tc(t)M . The curve with zero acceleration is precisely a geodesic of M .

Under the Riemannian connection, the second-order Taylor expansion of f along c is given by [9, 13]: f ∘ c(t) = f (p) + gp (gradf, v) · t + (cid:2)gp (Hessf (v), v) + gp (gradf, a)(cid:3) · + O(t3) (2) where Hessf is the (1, 1)-tensor known as the Riemannian Hessian of f .

In the literature, two common definitions of the Hessian exist. The present work adopts the (0, 2)-tensor definition (See Appendix A), denoted as Hessf . These two Hessian tensors are naturally related by [11]:

Hessf (X , Y) = g(Hessf (X), Y) = g(Hessf (Y), X) By expressing the directional derivatives of f in the directions of v and a, Eq.2 can be reformulated more compactly as f ∘ c(t) = f (p) + ∇v f |p · t + · [Hessf (v , v) + ∇a f]p · + O(t3).

The formulation using Hessf is often preferred in optimization literature, particularly when M is a submanifold of a Euclidean space, since Hessf can be computed by projecting the ambient Euclidean Hessian onto Tp M (See Section 4.3).

Hessf is positive definite if Hessf (X , X) > 0 for all non-zero vectors X Tp M .

Equivalently, Hessf is positive definite if g(Hessf (X), X) > 0 for all non-zero X .

1.2.2 Optimality conditions

The optimality conditions in M are analogous to their classical counterparts in Rn.

Specifically, a point p M is defined as a critical point of f if its gradient vanishes at p—or equivalently, ∂f = 0 (or df = 0) [13, 14]. When the Hessian is nondegenerate, it provides crucial information about the nature of the critical point, as formalized in the following proposition:

Proposition 1 ([13]) Let p M be a nondegenerate critical point of a smooth function f : M (cid:55)→ R. Then p is a strict local minimum of f if and only if the Hessian (either Hessf or Hessf) is positive definite.

Pos.1 is not the most economical, as ∂f = 0 implies Hessf = Hf (See Remark 1). The requirement of positive definiteness for Hessf can be relaxed to that of its Euclidean counterpart Hf . Alternatively, a similar simplification can also be achieved through appropriate parametrization, even for non-critical points. The geodesic- induced exponential map centered at p gives rise to normal coordinates, where the Christoffel symbols vanish at p, significantly simplifying the local representation:

Lemma 2 In normal coordinates around p, Hessfp = Hfp.

While Hessf generally depends on the metric, there is a noteworthy exception:

Lemma 3 (Lemma 68, [11]) If a function f : M (cid:55)→ R has a critical point at p, then Hessfp does not depend on the metric.

Remark 1 Strictly speaking, the expression Hessf = Hf constitutes a mild abuse of nota- tion, as Hf is not a tensor of M . In local coordinates, the components of Hessf are given by Hessf (∂i, ∂j) = ∂i∂j f − Γk ij ∂kf [11], where Einstein summation notation is adopted.

Hessf (∂i, ∂j) reduce to ∂i∂j f (the components of Hf) when either the Christoffel symbols vanish or at a critical point. It is in this sense that we use the notation Hessf = Hf . Con- sequently, the Hessian is often identified with its matrix representation in a given coordinate system, and the two are used interchangeably.

Lemma 2 and 3 establish that in a normal coordinate system, the Taylor expansion in Eq.4 can be simplified to a form analogous to that in Eq.1. We now generalize this result, proving that—with appropriate technical treatments—this property is not unique to normal coordinates but extends to any admissible coordinate system.

1.2.3 Contributions

The preceding observations motivate an investigation of Riemannian Newton' s method through the lens of invariance. The primary contributions of this work are as follows: (cid:136) We demonstrate that the second-order Taylor expansion of a differential function on a manifold exhibits form invariance under the Riemannian connection. (cid:136) Leveraging this invariant form, we formulate a purely second-order Riemannian Newton step. Crucially, it retains the same expression as the standard Euclidean Newton' s method. (cid:136) This invariant formulation clarifies key characteristics of Riemannian Newton' s method, naturally explaining its independence from the Riemannian connection and metric.

These properties summarized in Table 1 show that Riemannian Newton' s method is locally equivalent to its classical Euclidean counterpart and provide insights for guiding the implementation and refinement of the method.

Table 1 Comparison of Descent Directions in Euclidean and Riemannian Newton' s methods Manifolds Quadratic models 1st-order Descent Directions Steepest Standard gradf 2nd-order Descent Directions Steepest Hf −1∂f Hessf −1∂f Hessf −1gradf This work G−1∂f Hf −1∂f The remainder of this paper is organized as follows. Section 2 reviews the develop- ment of descent methods, with a particular focus on Riemannian Newton' s method.

In Section 3, we utilize the Taylor expansion to analyze the invariance properties of Riemannian Newton' s method, demonstrating that on a local chart, it is intrinsi- cally equivalent to the standard Euclidean Newton' s method. Section 4 compares the numerical characteristics and computational approaches of the existing Riemannian Newton' s method with the one proposed in this work, while Section 5 validates our findings with two illustrative examples. Finally, Section 6 concludes the paper.

2 Related work

Gradient descent is a fundamental parameter estimation method, renowned for its computational simplicity. A prominent instance is the steepest descent method; how- ever, its convergence can be notably slow on ill-conditioned problems, as its iterates follow a zigzagging trajectory toward the solution. The conjugate gradient method offers accelerated convergence without requiring matrix storage. Since both methods use only the gradient of the objective function to determine search directions, they are classified as first-order methods. These approaches have since been extended to Riemannian manifolds, where numerous variants have been proposed and are widely applied [3, 4, 8, 13]. In contrast, Newton' s method is a second-order technique that incorporates the Hessian. The resulting Newton direction is widely regarded as one of the most effective search directions available.

Historically, Newton' s method is also referred to as the Newton–Raphson method.

Its original conception traces to Fran¸cois Vi'ete, Isaac Newton, and Joseph Raphson in the 17th century, who laid the foundational idea for computing roots of polynomial equations. In 1740, Thomas Simpson adapted the method to nonlinear equations into a form closely aligned with its modern formulation, and further extended its applicability to minimization problems by setting the gradient to zero [15, 16]. The formal general- ization of Newton' s method to manifolds, by contrast, dates to the 1970s. It was not until Luenberger's seminal 1972 work that Newton' s method was explicitly formulated and analyzed for constrained optimization problems via the Riemannian structure of constraint sets [17]. Gabay (1982) subsequently extended Newton' s method and other descent methods to manifold settings, though this work did not formally employ the notion of Riemannian Hessian [13]. Smith (1994) and Udri¸ste (1994) formulated New- ton' s method for locating zeros of a 1-form on a Riemannian manifold [8, 14]. Edelman et al. (1998) utilized matrix algebraic structures to develop Newton and conjugate gradient algorithms on Stiefel or Grassmann manifolds [18]. In these manifold-based approaches, the exponential map is typically used to retract the Newton step vector from the tangent space back to the manifold. To circumvent the high computational cost of geodesics, Manton (2002) proposed using a simpler projection operator for local parameterization [19]. Adler et al. (2002) formalized a unified framework of retrac- tions—smooth maps from the tangent bundle to the manifold to execute projections [20]. Retractions have become a fundamental component of Riemannian computation [1, 9]. Absil and Malick (2012) showed that the definition of a retraction is highly flexible: any operation that maps a tangent vector back to the manifold along an " admissible" direction suffices [21]. This implies that Newton' s method implementation imposes relatively lenient requirements on retractions. For a systematic treatment of Riemannian Newton' s method, we refer the reader to [1, 9].

Implementing Riemannian Newton's method requires specialized considerations critical to its performance. A primary challenge is the lack of global convergence for the method, especially when initial guesses are poorly chosen. While Kantorovich' s theorem provides verifiable conditions at the initial point that establish convergence of the iterative process and the existence and uniqueness of the solution [22], it offers no criterion for selecting a suitable initial point. To ensure convergence from remote start- ing points, modified variants incorporating regularization [23], damping techniques [24], the Armijo condition [6], or adaptive trust-region strategies [7] can guarantee both global convergence and superlinear local convergence under mild assumptions. A detailed discussion of these implementation aspects lies beyond the scope of this work.

For a comprehensive overview of various approaches to implementing and refining the Newton scheme, we refer the reader to [2, 5, 9, 23] and the references therein.

The core of this generalization lies in establishing the Newton equation on the manifold M , which can be approached in three main ways.

Path-based Optimization This method conceptualizes optimization as a stepwise search for a path c on M that minimizes f [9, 13]. A quadratic model of f is derived from the Taylor expansion of f at p in Tp M to characterize the local variation of f along c. By constraining c to have zero initial acceleration (i.e., assume it is a geodesic or a second-order retraction curve [9]), we obtain the model: mp (v) = f (p) + gp (gradf, v) + gp (Hessf (v), v) Minimizing Eq.5 yields the Newton equation Hessf (X) = −gradf , and the cor- responding step is X = −Hessf −1gradf . Leveraging the assumption gradfp 0 enables a similar simplification.

Vector Field Zero-Finding This approach reframes the problem of identifying a critical point as locating a zero of a vector field Y [1, 20, 22]. Applying Newton' s method for root-finding to Y requires its Jacobian Jp (X) = ∇X Y . Since the Jacobian of Y = gradf is the Riemannian Hessian, the Newton equation Jp (X) = −Y p again results in the step X = −Hessf −1gradf .

One-Form Taylor Expansion This method derives the Newton step by means of a Taylor expansion of the differential 1-form ω [8, 14]. By an abuse of notation, we denote by $(\nabla\omega)$p two distinct yet closely related entities: the bilinear form induced by the covariant derivative of ω evaluated at p, and the corresponding homomorphism from Tp M to its dual space T∗ p M [8]. The Newton equation is then formulated as $(\nabla\omega)$p (• , X) = (∇X ω)p = −ωp . For ω = df , ∇X ω coincides with Hessf (• , X) [14], which yields the step X = −Hessf −1df . In local coordinates, it reduces to X = −Hessf −1∂f .

In this work, we adopt the path-based optimization framework. It provides a more intuitive characterization of the function' s variation, thereby facilitating the in-depth analysis presented in subsequent sections.

3 Invariance of Newton’ s method over a Riemannian

manifold Suppose the pair (U, x) be a chart on M , where the parameterization x : U → x(U) M is a homeomorphism of some domain U Rn on a subset x(U) M . The coor- dinates x1, · · · , xn of a point x U serves as the local coordinates of the point p = x(x) M . In the same sense we term this the local coordinate system of M at p. We adopt the Einstein summation convention in this section.

3.1 Invariance of the second-order Taylor expansion

By expressing the function f and the integral curve c in x, we obtain f ∘ x(x) = f (x1, · · · , xn), x = (x1, · · · , xn)T U x−1 ∘ c(t) = (x1(t), x2(t), · · · , xn(t))T respectively. Here f denotes, by a common abuse of notation, the coordinate repre- sentation of f in x. Therefore, restricting f to c, we compute its directional derivative as follows: c′(0)f = f (x1(t), · · · , xn(t)) (cid:12) (cid:12) (cid:12) (cid:12)t=0 (cid:18) (xi)′ ∂ (cid:19) Let p = c(0). In the basis { ∂i|t=0} of Tp M , we define X i(t) = d dt xi(t).

For notational simplicity, all subsequent evaluations are carried out at t = 0; for instance, X i(0) is denoted as X i, and the tangent vector c′(0) admits the represen- tation X = X i∂i. By invoking the standard identities for the covariant derivative (∇X f = X i∂if), the Hessian (Hessf (∂i, ∂j) = ∂i∂jf − Γk ij∂kf) and the acceleration ijX iX j(cid:17) along c [11], Eq.4 can be written as (cid:16) dX k (cid:16) DX dt + Γk (cid:17) f ∘ c(t) = f (p) + X i∂if · t (cid:20) X iX j(∂i∂jf − Γk ij∂kf) + (cid:18) dX k ijX iX j (cid:19) (cid:21) + O(t3) = f (p) + (cid:18) (cid:19) ∂if + X iX j∂i∂jf · + O(t3) Denote s(t) = x−1 ∘c(t). Expanding s(t) using Taylor series around t = 0 in Tp M , we have ∆s = s(t) − s(0) = s ′(0) · t + s ′′(0) · + O(t3), whence it follows that X · t = s ′(0) · t = ∆s + O(t2) = s ′(0) · t + s ′′(0) · = ∆s + O(t3) Without loss of generality, consider a parameterization of x centered at p = x(0) such that ∆s = s(t). Given that ∆s = O(t) in a neighborhood of p, substituting the last two expressions into Eq.6 and collecting terms up to O(t3) yields: f (s) = f (p) + ∂x f (p)T s + s T Hx fp s + O(s 3) Eq.7 establishes that the second-order Taylor expansion of f ∘c(t) is form-invariant in coordinates upon extension from Rn to M . We thereby get a more rigorous quadratic model function:

Fp (s) = f (p) + ∂f (p)T s + s T Hfp s The first- and second-order steepest descent directions are as follows: (cid:136) Steepest Descent Direction: X sd = −(G−1∂f)T ∂x . This is the well-known local coordinate representation of gradf . It is an invariant expression that also depends on the choice of metric. (cid:136) Newton Step: N ns = −(Hf −1∂f)T ∂x . It is form-invariant that is independent of the choice of metric. This result implies that Riemannian Newton’ s method is locally equivalent to the classical Euclidean Newton’ s method.

The aforementioned results are summarized in 1.

3.2 Local convergence

Given a chart (U, x), a family of local charts can be generated by defining xp (s) = x(x−1(p) + s) = x(x + s) around p x(U). The parameterization xp is now a local diffeomorphism centered at p = x(x) and the differential dxp : Tx U → Tp M is an isomorphism. If {ei} is the basis for Tx U , the differential acts as dxp (xiei) = xi∂i or, for the inverse, dx−1 p (X i∂i) = X iei.

The retraction can be defined as ϕp xp ∘ dx−1 p . Let x 0 = x−1(p 0) for the initial iterate p 0. The iterative scheme is then given by: x k+1 = x−1(ϕp k (−(Hf −1∂f)T ∂x)) = x−1(p k) − dx−1 ((Hf −1∂f)T ∂x) where p k = x(x k).

By the definition that ∂if (p) Di(f ∘ x)(x−1(p)) = ∂ ∂xi f (x1, ⋯ , xn) [12], Eq. 9 is in fact equivalent to the classical Newton-Raphson iteration in Rn : x k+1 = x k − Hf −1 ∂f (x k) Suppose that Hf is Lipschitz continuous in a neighborhood of a critical point p ∗, and that Hfp ∗ is positive definite. Then provided p 0 is sufficiently close to p ∗, the sequence of iterates generated by Eq.9 converges quadratically to p ∗. This is a straightforward consequence of the convergence properties of the classical Newton' s method [5].

4 Comparative Analysis

This section presents a systematic comparison of the properties and computational challenges associated with Newton' s methods derived from the two model functions Eqs.5 and 8.

4.1 The Distinct Roles of gradfp 0 and $c''(0) = 0$

Either of these two conditions reduces Eq.2 to Eq.5, yet their implications differ fundamentally in the derivation of Newton steps on Riemannian manifolds.

A retraction satisfying $c''(0) = 0$ is designated as a second-order retraction [9], with the exponential map being the canonical example. Notably, such a retraction still generates Newton iterates with a quadratic convergence rate, a property invariant to the choice of retraction [1, 9]. This feature can be verified directly via 7: the quadratic model is fully determined by the Euclidean gradient and Hessian, and is entirely independent of the specific form of the curve c. This naturally raises a key question: if the assumption $c''(0) = 0$ does not affect the quadratic convergence of the method, why does the formal derivation appear to rely on it? In particular, what rigorous mathematical procedure justifies the transformation from Eq.5 into Eq.8?

The answer lies in the approximation error. Denote the Riemannian-Euclidean ij∂kf (cid:3) = Hessf − Hf and it is straightforward to verify Hessian difference as Q (cid:2)Γk that Q O(∂f) as p → p ∗. This permits expressing the model mp (v) as: mp (v) = f (p) + ∂f (p)T v + = Fp (v) + O(∂f v 2) v T Hfp v + (p → p ∗) v T Qp v Although the leading terms of Eqs. 7 and 10, namely Fp , are formally identical, Eq.7 is derived from a more rigorously principled formulation. Specifically, the model Fp (s) employs a secant vector s = Δs, whereas mp (v) relies on the tangent vector v = t · s ′(0). This discrepancy in vector selection underpins the divergent mathematical foundations of the two expressions, yet no such distinction is enforced in practical computations.

Remark 2 The core reason that mp give rise to a quadratically convergent Newton' s method lies in gradf (p ∗) = 0. This enables the term associated with a to be treated as a higher- order infinitesimal when p is sufficiently close to p ∗. Since c′′(0) = 0 holds for any geodesic c, the exponential map is often directly utilized in the literature to derive Eq.5. It worth noting, however, that the exponential map serves solely to eliminate the a-related term and acts as a projection operator; it is not a critical prerequisite for quadratic convergence.

To summarize, we formalize the following key result:

Proposition 4 Fp and mp approximate f with the following error bounds: f (s) = Fp(s) + O(s 3) = mp(s) + O(Q s 2 + s 3).

Moreover, Fp = mp hold if either of the following condition is satisfied: (cid:136) p is a critical point of f . (cid:136) Normal coordinates centered at p are adopted.

4.2 Comparison of Newton steps: Hf −1∂f and Hessf −1gradf

The subsequent two propositions elucidate the intrinsic connections between these two steps.

Proposition 5 Hessf −1∂f = Hessf −1gradf . Proof The identity in Eq. 3 holds for all tangent vectors X , Y Tp M . In local coordinates, this relation implies Hessf = G · Hessf . We thus derive the equality below:

Hessf −1∂f = Hessf −1G−1∂f = Hessf −1gradf. For the analysis presented in this section, we impose the norm consistency condi- tion: for example, the vector norm is taken as the p-norm, and the matrix norm is its induced matrix norm. Under the assumptions that Hf is invertible and all relevant functions are Lipschitz continuous in an open neighborhood of p ∗, we establish the following Proposition 6 Hf −1∂f − Hessf −1gradf O(∂f 2), p → p∗.

Proof Let P = Hf and I the identity matrix. By applying the Sherman-Morrison-Woodbury formula, we proceed with the derivation: Hf −1 − Hessf −1 = P−1 − (P − Q)−1 I − (I − QP−1)−1(cid:17) = P−1 (cid:16) = P−1 (cid:16) = P−1Q(I − P−1Q)−1P−1 I − (I + Q(I − P−1Q)−1P−1) (cid:17) As ∂f → 0, we have (I−P−1Q)−1 → I. It follows that P−1 −(P−Q)−1 O(∂f).

Consequently, Hf −1∂f − Hessf −1gradf = (Hf −1 − Hessf −1)∂f O(∂f 2) Remark 3 While the steps Hf −1∂f and Hessf −1gradf yield identical convergence rates, a subtle distinction exists: when the iteration point p is far away from p $*$, the term Q cannot be neglected. In this regime, mp does not constitute a strict quadratic model of f , as further illustrated in Example 2 (Section 5).

4.3 Computing Gradients and Hessians

While a coordinate-free framework is theoretically favored for geometric analysis, practical numerical computations demand a parametric (or algebraic) representation.

The merit of this representation stems from its direct compatibility with numerical implementation and its capacity to streamline algorithmic derivations. A manifold can be parameterized either intrinsically (through internal coordinates) or extrinsically (through embedding in a Euclidean space). For the sake of clarity, this and subse- quent sections dispense with the Einstein summation notation; vector components are explicitly denoted with subscripts instead of superscripts.

Projection Method The computation of Hessf -1gradf poses a challenge as Riemannian connections generally lack closed-form expressions. However, if M is a submanifold of a Euclidean space E, its Riemannian connection coincides with the orthogonal projection onto the tangent space of the standard vector derivative of a smooth extension of the Riemannian gradient vector field in E.

Consider a Riemannian submanifold of M E and a smooth function f : M $\to$ R.

We introduce a local extension ˆf of f to E, along with the corresponding extension ˆg of gradf . Let Projp denote the orthogonal projector onto Tp M at p M . The following two propositions, which underpin the subsequent computations, are invoked herein [1, 9]:

Proposition 7 gradf = Proj(grad ˆf). Proposition 8 Hessf (X) = Proj(Dˆg(X)) A cost function is typically formulated as a function of coordinates in E constrained to a submanifold. If both the function and constraint admit concise algebraic forms in terms of x E (e.g., matrix operations), the projection method tends to yield computational advantages [1, 9, 18]. Otherwise, the parameterization method based on intrinsic coordinates is preferable. We next demonstrate this with a class of implicitly defined manifolds .

Parametrization Method A prevalent form of manifold constraint is given by the implicit equation h(z) = 0, where h : En $\to$ Rm denotes a smooth map with component functions hi for i = 1, $\cdots$, m. We assume that 0 is a regular value of h, meaning the Jacobian of h at each z h -1(0) is of full rank m. Under this regularity condition, the set M = {z En : h(z) = 0} is an embedded submanifold of En with dimension n $-$ m.

By virtue of the implicit function theorem, the regularity of h guarantees the existence of a local parameterization in a neighborhood of a point p. Specifically, we can define it as z : U Rn$-$m $\to$ En wherein ¯x $\to$ z = (¯x , ϕ(¯x)) = (x , y): here, ¯x Rn−m denotes independent parameters, while y = ϕ(¯x) Rm represents dependent (cid:12) coordinates. This parameterization construction requires the Jacobian submatrix ∂h (cid:12) (cid:12)p to be non-singular. Let ¯f (¯x) = f (x , y). Elementary algebraic manipulations yield the following expressions for the first and second derivatives of ¯f with respect to ¯x : $ $2 ¯f ∂ ¯x 2 = (cid:2)In−m, K(cid:3) Hx ,y f (cid:20)(cid:18) $ $2h (cid:20)In−m (cid:21) (cid:19) ∂x 2 + K ∂x ∂y where L = (cid:17)−1 (cid:16) ∂h ∂y , K = − ∂h L In−m + (cid:16) ∂h (cid:17)−1 (cid:18) $ $2h ∂y ∂x (cid:19) (cid:21) L KT The matrix derivative notations employed herein are defined as follows: (cid:21) (cid:20) ∂f ∂xk = (cid:21) (cid:20) ∂hj (cid:21) (cid:20) ∂hj ∂vec(F) ∂vec(X) x11 ⋯ x1l xk1 ⋯ xkl , F = where h = (h1, h2, ⋯ , hm)T , x = (x1, x2, ⋯ , xn)T , X = F11 ⋯ F1q Fp1 ⋯ Fpq . The vec(•) operator stacks the columns of matrix X into a vector: vec(X) = (x11, ⋯ , xk1, x21, ⋯ , xk2, ⋯ , x1l, ⋯ , xkl)T . The symbol denotes the Kronecker product. In denotes the n-by-n identity matrix.

5 Examples

To illustrate the preceding theoretical results, we present two specific examples, focusing on a comparative analysis of the advantages and disadvantages of the two approaches in practical computations. For notional simplicity, we omit the index p in the function.

Example 1: Unit Euclidean Sphere Consider the unit Euclidean sphere Sn = {x Rn+1 : x = 1} and the objective function f (x) = b T x defined on Sn.

Parametrization Method Sn is implicitly defined by h(x) = x T x − 1. We first establish a suitable local param- eterization. Without loss of generality, consider the chart (U, x) where U = {¯x = (¯x1, ⋯ , ¯xn)T Rn : ¯x T ¯x < 1}. The parameterization x : U → Sn is given by Rn, where xi = ¯xi for i = 1 ⋯ n and ¯x → x = (x1, ⋯ , xn, xn+1)T = (cid:21) (cid:20)¯x 1 − ¯x T ¯x > 0. The metric induced on Sn by its embedding in Rn+1 is y = xn+1 = y2 ¯x ¯x T . (cid:21) (cid:20)¯x where b = (cid:21) (cid:20) ¯b , direct computation by 11 Since ¯f (¯x) = f (¯x , y) = (¯b T , by) yields: (cid:20) ∂f ∂ ¯xi (cid:21) = ¯b − $ $2 ¯f ∂ ¯x 2 = − (cid:18) (cid:19) ¯x ¯x T Consequently, the Newton step is N ns = − (cid:17)−1 (cid:16) $ $2 ¯f ∂ ¯x 2 represented in the local coordinate system of Sn at x . ∂ ¯x = y G−1 ∂ ¯f ∂ ¯x = y gradf , Projection Method Alternatively, we can compute the Newton step by treating Sn as an embedded sub- manifold of Rn+1. We apply the analytic expression of gradf to the extension of gradf from Sn to Rn+1, yielding gradf = Proj(grad ˆf) = Proj(b) = b − (b T x)x , Hessf (v) = Proj(D(gradf)(v)) = −(b T x)v .

The Newton step is then N ns = −Hessf −1gradf = 1 coordinates of Rn+1.

Notably, the two Newton directions are directionally consistent with the Riemannian gradient, differing only by a scaling factor.

For manifolds given in an explicit parametric form or admitting a natural cover by a set of coordinate charts, the parameterization method exhibits superior compu- tational efficiency and straightforward implementation. Partitioning the independent variables, however, is non-trivial: while certain problems possess an inherent structure that naturally facilitates variable partitioning, such guiding principles lack univer- sal applicability, and arbitrary partitioning may lead to suboptimal computational performance [13].

Example 2: Paraboloid Consider the paraboloid M defined by the equation z = x2 + y2. Its implicit represen- tation is given by the zero-level set of the function h(x, y, z) = x2 + y2 − z = 0. We define a cost function on M as f (p) = f (x, y, z) = z. This is a strictly convex function with a unique global minimum at the origin.

The x-y plane serves as a natural parameter domain for M . Consequently, computing the Newton step proves straightforward:

N ns = − (cid:20)2 0 (cid:21)−1 (cid:20)2x (cid:21) (cid:21) (cid:20)x This update rule converges to the optimal point in just one iteration, representing the standard full second-order Newton step.

For comparison, we now compute the step using the projected Hessian Hessf .

The unit normal vector to the surface is n = ∂h , where a = (cid:112)4x2 + 4y2 + 1. The Euclidean gradient of ˆf is grad ˆf = (0, 0, 1)T . The Riemannian gradient on the manifold is: ∂h = (2x,2y,−1)T gradf = Proj(grad ˆf) = grad ˆf − grad ˆf , n n = a2 (x, y, 2x2 + 2y2)T .

Calculating Hessf is more involved. Since M can be parameterized as r (x, y) = (x, y, x2 + y2)T , the vectors of {r x, r y} form a basis of Tp M at the point p = r (x, y).

Let u = ur x + vr y be a vector in Tp M . Extending gradf analytically from M 2 to R3, we have D(gradf) = D(grad ˆf) − D(grad ˆf , n)n − grad ˆf , n Dn, and thus Hessf = −Proj(grad ˆf , n Dn) = Applying this to u gives Proj(Dn) Proj(Dn(u)) = a4 Proj(V), Hessf (u) = u(4y2 + 1) − 4vxy v(4x2 + 1) − 4uxy where V = It can be verified that Proj(V) = V − V , n n = with B = (4y2 + 1)2 + 16x2y2 −8xy(2x2 + 2y2 + 1) −8xy(2x2 + 2y2 + 1) (4x2 + 1)2 + 16x2y2 a2 B ¯u, , and ¯u = (u, v)T .

Thus, Hessf (u) = 2 a6 B ¯u. The Newton equation in matrix form is B ¯u = c, 2 gradf . Note that the three scalar equations from B ¯u = c are linearly where c = − a6 dependent. A closed-form solution can be obtained using the pseudo-inverse of B:

N ns = (BT B)−1BT c = − (cid:21) (cid:20)x − 8$ 2(1+2 $2) (cid:21) (cid:20)x (cid:112) x2 + y2.

Comparing the results from Eq.12 and Eq.14 aligns with the expectations of Proposition 6.

Remark 4 Hessf −1gradf is parametrization-independent but requires computing the Rie- mannian connection; whereas Hf −1∂f is parametrization-dependent, does not involve the metric, and eliminates the need to compute the Riemannian connection. The choice between them can be made based on the specific characteristics of the application.

6 Conclusion

All n-dimensional Riemannian manifolds share the same infinitesimal structure [11].

Specifically, a manifold M in a neighborhood of a point p can be approximated by its tangent space Tp M . By analogy, a function f defined on M can be approximated via its Taylor expansion in Tp M . This paper investigates Riemannian Newton' s method from the perspective of Taylor-expansion form invariance of f and demonstrates that the Euclidean and Riemannian Newton' s methods–corresponding to quadratic models Eq.5 and Eq.8, respectively–exhibit strong mathematical consistency. Most crucially, Riemannian Newton' s method is locally equivalent to its Euclidean counterpart.

Specifically, this work reveals several key invariance properties: (cid:136) The second-order Taylor expansion of f on M is form-invariant. (cid:136) The Newton step, Hf −1∂f , derived from this quadratic model, is itself form- invariant in coordinates, generalizing the Euclidean Newton step from Rn to (cid:136) The step Hf −1∂f is independent of the Riemannian metric and connection. (cid:136) The step Hf −1∂f is fully second-order.

These properties yield promising optimization strategies for Riemannian Newton- type algorithms. Since strict projection to constrain each iterate vector within a coordinate chart to the manifold is unnecessary—provided all iterates remain within the same chart until a successive chart update is required—a well-designed algorithm may thus avoid computationally expensive operations such as parallel translation.

Moreover, the first-order steepest descent direction is likewise independent of the Rie- mannian connection. These considerations confer substantial benefits for the design and implementation of manifold-based algorithms, as they reduce computational over- head while preserving theoretical rigor. For example, in trust-region methods, one must approximate Hessf or Hessf with a suitable symmetric matrix [1, 7, 9]. The results of this work demonstrate that Hf can be reliably utilized for algorithm design; this approach is not only simpler but also circumvents potential concerns regarding additional loss of accuracy.

Appendix A Some Notations We use the following notational conventions: scalars are italicized, vectors are in bold italic, and matrices or tensors are in boldface. Vectors are column vectors by default (e.g., x = (x1, x2)T). Common symbols are summarized in Table A1, with local excep- tions noted inline. When no ambiguity arises, the subscripts p or x are typically omitted. To streamline derivations, Section 3 employs the Einstein summation convention with superscripted indices for vector components. In all other sections, this convention is discarded, with either coordinate-free notation or standard subscripted coordinate notation employed.

Table A1 Notation Summary Notation Description Hx f, $ $2f (x) gp (X , Y) gradfp Hessfp Hessfp n-dimensional Euclidean space endowed with an inner product. When n is clear from context, it is abbreviated to E.

Standard En equipped with its canonical inner product. k-th component of a vector x En. When Einstein summation is not used, the traditional notation xk is retained.

Euclidean gradient of a scalar function f with respect to x Rn, i.e., ∂xn f (cid:1)T Rn. In the absence of ambiguity, abbreviated to (cid:0) ∂ ∂x1 f ··· ∂f = (cid:0)$ $1f ··· ∂nf (cid:1)T .

Classical Euclidean Hessian of f . Vector fields on a manifold M .

A parameterization of the bijective map x : U Rn → x(U) M , often termed a chart in the literature. For notational convenience and fol- lowing standard convention, we denote the pair (U, x) as a chart, while reserving " parameterization" for x itself. Notably, these two notions are conceptually equivalent.

Coordinate vector fields on M , abbreviated to ∂i for brevity.

Formal column vector formed by stacking basis vectors (cid:8) ∂ (cid:0) ∂ (cid:9), i.e., ∂x = (cid:1)T . ··· Riemannian metric on M , often denoted by the inner product symbol ·, · too. Its matrix representation is G.

Value of g at point p M , evaluated on X , Y Tp M .

Riemannian gradient of f at p M , defined as the vector field satisfying gp (gradf, X) = dfp (X) for all X Tp M .

Exclusively denotes the Riemannian (Levi-Civita/canonical) connection in this work.

Covariant Hessian of f at p M , a symmetric (0, 2)-tensor defined by the Lie derivative 1 Alternative definition of the Riemannian Hessian of f , a self-adjoint (1, 1)-tensor given by Hessfp (X) = ∇X gradfp for all X Tp M .

Christoffel symbols of the second kind.

References

[1] Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, ??? (2008) [2] Hu, J., Liu, X., Wen, Z.-W., Yuan, Y.-X.: A brief introduction to manifold opti- mization. Journal of the Operations Research Society of China 8(2), 199–248 (2020) [3] Sato, H.: Riemannian Optimization and Its Applications. Springer, Gewerbe- strasse 11, 6330 Cham, Switzerland (2021) [4] Fei, Y., Liu, Y., Jia, C., Li, Z., Wei, X., Chen, M.: A survey of geometric optimization for deep learning: From Euclidean space

to Riemannian manifold (2025) [5] Nocedal, J., Wright, S.J.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY (2000) [6] Yang, Y.: Globally convergent optimization algorithms on Riemannian manifolds:

Uniform framework for unconstrained and constrained optimization. Journal of Optimization Theory and Applications 132(2), 245–265 (2007) [7] Zhao, S., Yan, T., Wang, K., Zhu, Y.: Adaptive trust-region method on riemannian manifold. Journal of Scientific Computing 96(3), 67 (2023) [8] Smith, S.T.: Optimization techniques on Riemannian manifolds. Fields Institute Communications 3, 113–135 (1994) [9] Boumal, N.: An Introduction to Optimization on Smooth Manifolds. Cambridge University Press, Cambridge, United Kingdom (2023) [10] Carmo, M.P.: Riemannian Geometry. Progress in Mathematics, vol. 108.

Birkh¨auser Boston, Boston, MA (1992) [11] Petersen, P.: Riemannian Geometry, 2nd edn. Graduate Texts in Mathematics, vol. 171. Springer, New York (2006) [12] Spivak, M.: A Comprehensive Introduction to Differential Geometry, Vol. 1, 3rd edn. Publish or Perish, Wilmington, DE (1999) [13] Gabay, D.: Minimizing a differentiable function over a differential manifold.

Journal of Optimization Theory and Applications 37(2), 177–219 (1982) [14] Udri¸ste, C.: Convex Functions and Optimization Methods on Riemannian Mani- folds. Kluwer Academic Publishers, Dordrecht (1994) [15] Kollerstrom, N.: Thomas Simpson and "Newton' s method of approximation" : An enduring myth. The British Journal for the History of Science 25(3), 347–354 (1992) [16] Ypma, T.J.: Historical development of the Newton-Raphson method. SIAM Review 37(4), 531–551 (1995) [17] Luenberger, D.O.: The gradient projection method along geodesics. Management Science 18(11), 620–631 (1972) [18] Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications 20(2), 303–353 (1998) [19] Manton, J.H.: Optimization algorithms exploiting unitary constraints. IEEE Transactions on Signal Processing 50(3), 635–650 (2002) [20] Adler, R.L., Dedieu, J.P., Margulies, J.Y., Martens, M., Shub, M.: Newton' s method on Riemannian manifolds and a geometric model for the human spine.

IMA Journal of Numerical Analysis 22(3), 359–390 (2002) [21] Absil, P.-A., Malick, J.: Projection-like retractions on matrix manifolds. SIAM Journal on Optimization 22(1), 155–188 (2012) [22] Ferreira, O.P., Svaiter, B.F.: Kantorovich' s theorem on Newton' s method in Riemannian manifolds. Journal of Complexity 18(1), 304–329 (2002) [23] Hu, J., Milzarek, A., Wen, Z., Yuan, Y.: Adaptive quadratically regularized New- ton method for Riemannian optimization. SIAM Journal on Matrix Analysis and Applications 39(3), 1181–1207 (2018) [24] Bortoloti, M.A.d.A., Fernandes, T.A., Ferreira, O.P., Yuan, J.: Damped Newton' s method on Riemannian manifolds. Journal of Global Optimization 77(3)

Note: Figure translations are in progress. See original paper for figures.